С. В. ПОРШНЕВ

КОМПЬЮТЕРНОЕ МОДЕЛИРОВАНИЕ ФИЗИЧЕСКИХ ПРОЦЕССОВ В ПАКЕТЕ MATLAB

УЧЕБНОЕ ПОСОБИЕ

Издание второе, исправленное

ЛАНЬ®
САНКТ-ПЕТЕРБУРГ • МОСКВА • КРАСНОДАР
2011

ББК 22.311я73
П 60

Поршнев С. В.

П 60 Компьютерное моделирование физических процессов в пакете MATLAB: Учебное пособие. 2-е изд., испр. — СПб.: Издательство «Лань», 2011. — 736 с.: ил. (+CD). — (Учебники для вузов. Специальная литература).

ISBN 978-5-8114-1063-7

Книга посвящена обучению основам компьютерного моделирования физических процессов. Каждая глава содержит теоретический материал, описание математических методов, используемых для построения компьютерных моделей физических процессов, а также задачи для самостоятельного решения. В качестве базового программного продукта используется пакет MATLAB.

К книге прилагается CD-диск, содержащий программные реализации каждой из рассмотренных в ней компьютерных моделей, реализованных автором на внутреннем языке пакета MATLAB в виде *m*-файлов и *m*-функций.

Для студентов вузов, обучающихся по специальностям математика, информатика, физика, может быть полезна для преподавателей соответствующих дисциплин, специалистов.

ББК 22.311я73

Обложка
А. Ю. ЛАПШИН

Охраняется законом РФ об авторском праве.
Воспроизведение всей книги или любой ее части
запрещается без письменного разрешения издателя.
Любые попытки нарушения закона
будут преследоваться в судебном порядке.

© Издательство «Лань», 2011
© С. В. Поршнев, 2011
© Издательство «Лань»,
художественное оформление, 2011

ОГЛАВЛЕНИЕ

Предисловие 8

1 Моделирование относительных движений в классической механике 12
1.1. Введение 12
1.2. Построение орбиты Луны в гелиоцентрической системе отсчета 13
1.3. Построение орбиты Марса в системе отсчета, связанной с Землей . . . 21

2 Физические процессы, описываемые дифференциальными уравнениями первого порядка 29
2.1. Моделирование остывания нагретых тел 29
2.2. Алгоритм Эйлера 30
2.3. Программа для решения дифференциальных уравнений первого порядка 31
2.4. Оценка коэффициента остывания по экспериментальным результатам 36
2.5. Решение дифференциальных уравнений методом Рунге—Кутта четвертого порядка 48
2.6. Моделирование радиоактивного распада 50
2.7. Моделирование цепной реакции ядерного взрыва 51

3 Динамика материальной точки 54
3.1. Общая постановка задачи математического описания динамики материальной точки 54
3.2. Движение тел в гравитационном поле без учета трения 55
3.3. Движение тел в гравитационном поле с учетом силы трения 62

4 Задача Кеплера 68
4.1. Введение 68
4.2. Уравнения движения планет 68
4.3. Численное моделирование орбиты 73
4.4. Проверка второго закона Кеплера 75
4.5. Пространство скоростей 84
4.6. Моделирование Солнечной системы 87

5 Моделирование статических электрических и магнитных полей 91
5.1. Электростатическое поле системы неподвижных электрических зарядов 91
5.2. Магнитное поле витка с постоянным током 102
5.3. Магнитное поле соленоида с постоянным током 109
5.4. Магнитное поле тороидальной обмотки с постоянным током 139
5.5. Численное решение уравнений Лапласа и Пуассона 146

6 Моделирование движения электрических зарядов в постоянных электрических и магнитных полях 164
6.1. Постановка задачи 164
6.2. Рассеивание частиц в центральном поле. Опыт Резерфорда 165
6.3. Моделирование движения электрических зарядов в постоянном магнитном поле 178
6.4. Моделирование движения электрических зарядов в постоянных электрических и магнитных полях 183
6.5. Моделирование движения электрических зарядов в неоднородном магнитном поле 187
6.6. Моделирование движения заряда в сложных электростатических полях 208

7 Фурье-анализ непрерывных и дискретных функций 219
7.1. Введение 219
7.2. Разложение периодических сигналов в ряды Фурье 219
7.3. Эффект Гиббса 223
7.4. Спектральный анализ непрерывных функций 228
7.5. Спектральный анализ дискретных функций 232
7.6. Спектральный анализ дискретных функций конечной длительности . . 235
7.7. Быстрое преобразование Фурье 239
7.8. Влияние длины интервала анализа на точность измерения частоты монохроматического сигнала 243

8 Моделирование колебательных процессов 252
8.1. Линейный гармонический осциллятор 252
8.2. Математический маятник 257
8.3. Затухающие колебания 262
8.4. Вынужденные колебания линейного гармонического осциллятора . . . 263
8.5. Двойной плоский маятник 266
8.6. Маятники с осциллирующей точкой подвеса 275
8.6.1. Компьютерное моделирование математического маятника с точкой подвеса, совершающей горизонтальные гармонические колебания . . 275
8.6.2. Математическая модель маятника с точкой подвеса, совершающей вертикальные гармонические колебания 280
8.6.3. Математическая модель маятника с точкой подвеса, совершающей движение по окружности 285
8.7. Моделирование колебательной химической реакции 289

9 Моделирование волновых явлений 315
9.1. Введение 315
9.2. Моделирование свободных колебаний цепочки связанных осцилляторов 315
9.3. Моделирование вынужденных колебаний цепочки связанных гармонических осцилляторов 330
9.4. Моделирование волновых движений 340
9.5. Фурье-анализ волновых пакетов, движущихся в среде с дисперсией . . 345
9.6. Моделирование явлений интерференции и дифракции 360
9.7. Геометрическая оптика 373
9.8. Поляризация 378

10 Моделирование систем, состоящих из большого числа частиц, методом молекулярной динамики 382
10.1. Введение 382
10.2. Математическая модель статистической системы 382
10.3. Численный алгоритм решения системы уравнений движения 385
10.4. Моделирование методом молекулярной динамики системы, состоящей из большого числа частиц 389
10.5. Оценка макроскопических характеристик статистической системы . . . 403
10.6. Оценка коэффициентов переноса методом молекулярной динамики . . 411
10.7. Моделирование фазовых переходов методом молекулярной динамики . 416
10.8. Основные итоги 419

11 Методы Монте-Карло 420
11.1. Введение 420
11.2. Численные методы интегрирования функций, зависящих от одной переменной 420
11.3. Основы метода Монте-Карло 424
11.4. Алгоритм генерации случайных чисел с равномерным законом распределения 435
11.5. Алгоритм Метрополиса 438

12 Случайные блуждания 444
12.1. Введение 444
12.2. Одномерные случайные блуждания 444
12.3. Метод случайных блужданий на плоскости 454
12.4. Моделирование движения решеточного газа 473
12.5. Непрерывная модель случайных блужданий 478

13 Моделирование статистической системы в процессе релаксации и состоянии равновесия 481
13.1. Введение 481
13.2. Моделирование процесса релаксации статистической системы 482
13.3. Энтропия 492

14 Компьютерное моделирование микроканонического ансамбля методом Монте-Карло 501
14.1. Микроканонический ансамбль 501
14.2. Моделирование микроканонического ансамбля 502
14.3. Модель Изинга 511

15 Моделирование канонического ансамбля методом Монте-Карло 528
15.1. Введение 528
15.2. Канонический ансамбль 528
15.3. Алгоритм Метрополиса для канонического ансамбля 529
15.4. Моделирование двумерной модели Изинга методом канонического ансамбля 542

16 Моделирование квантовых систем 556
16.1. Введение 556
16.2. Методы численного решения стационарного уравнения Шредингера . . 558
16.3. Моделирование колебаний двухатомной молекулы в квазиклассическом приближении 570
16.4. Нестационарное уравнение Шредингера 588
16.5. Оценка энергии основного состояния квантовой системы методом Монте-Карло 601
16.6. Оценка энергии основного состояния квантовой системы вариационным методом Монте-Карло 610

17 Моделирование фрактальных объектов 619
17.1. Введение 619
17.2. Избранные понятия фрактальной геометрии и хаотической динамики . 619
17.3. Рекурсивный алгоритм построения фрактальных объектов 621
17.4. L-системы и терл-графика 630
17.5. Системы итерированных функций 644
17.6. Методы оценки фрактальной размерности 662

A Основные приемы работы с пакетом MATLAB 674
A.1. Введение в пакет MATLAB 674
A.2. Работа в командном окне 687
A.2.1. Вход в систему MATLAB 687
A.2.2. Интерактивный доступ к справочной информации 688
A.3. Редактирование и повторный вызов командной строки 690
A.3.1. Формат ввода 690
A.3.2. Копия протокола сессии 691
A.4. Создание матриц 691
A.4.1. Явное задание матриц 691
A.4.2. Подматрицы и использование двоеточия 692
A.4.3. Функции построения матриц 694
A.5. Операции, выражения и переменные 694
A.5.1. Правила записи операторов 694

A.5.2. Матричные операции 695
A.6. Операции с массивами 696
A.7. Сохранение данных из рабочей области 696
A.8. Операторы FOR, WHILE, IF, CASE и операторы отношения 696
A.8.1. Цикл FOR 696
A.8.2. Цикл WHILE 698
A.8.3. Условный оператор IF 698
A.8.4. Оператор переключения CASE 698
A.9. Условия (операторы отношения) 699
A.10. Функции MATLAB 700
A.10.1. Скалярные функции 700
A.10.2. Векторные функции 700
A.10.3. Матричные функции 700

B M-файлы 702
B.1. Файлы-программы 702
B.1.1. Файлы-функции 702
B.1.2. Текстовые строки, сообщения об ошибках 706
B.2. Работа с m-файлами 706
B.2.1. Список путей доступа к файлам 707
B.2.2. Использование редактора/отладчика 708
B.2.3. Отладка m-файлов 710

C Создание графического интерфейса пользователя 713

D Оценки погрешностей численного интегрирования 724

ПРЕДИСЛОВИЕ

Анализ опыта преподавания физики в высших учебных заведениях показывает, что до настоящего времени преподавание большинства курсов физических дисциплин остается «классическим». Оно базируется на «трех китах»: теоретическом курсе, излагаемом в виде лекций; семинарских занятиях, на которых проводится решение задач без использования ПК; лабораторных занятиях. Как следствие, не удается сформировать у обучаемых целого ряда умений, в том числе: умения дать физически правильную формулировку поставленной задачи; умения обосновать выбор математического аппарата и создать математическую модель изучаемого процесса; умения разработки и отладки компьютерной реализации математической модели; умения проводить анализ и оценивать адекватность получаемых результатов.

Такое положение объясняется во многом тем, что до недавнего времени учебные заведения не были обеспечены достаточным количеством вычислительной техники. Однако в настоящее время большинство высших учебных заведений оснащены ПК, возможности которых позволяют решать в реальном времени задачи, требующие большого объема вычислений, назрела необходимость пересмотра подходов к использованию ПК в преподавании физики.

Автор считает, что выход из создавшегося положения состоит в дополнении «трех китов», на которых основана методика преподавания физики, соответствующими курсами лабораторных работ по компьютерному моделированию. Это, во-первых, позволит внедрить принципы компьютерного мышления в изучение физики. Во-вторых, потребует от студента более глубокого проникновения в суть изучаемой проблемы, будет способствовать закреплению физического материала и развивать физическую интуицию. Первые шаги в развитии именно такого подхода сделаны в [1, 2]. Здесь удалось отойти от традиционных для книг, посвященных численному моделированию, подходов, и на примерах решения широкого класса физических задач показать возможности использования ПК, как экспериментальной установки для проведения физических опытов.

Не ставя под сомнение высокий научно-методической уровень этих книг и их актуальность, необходимо отметить, что они ориентированы на возможности ПК десятилетней давности (в первую очередь это относится к программному обеспечению). Учитывая современное состояние аппаратных и программных средств ПК, по нашему мнению, более разумно ориентироваться на специализированные пакеты для математических вычислений, одним из которых является пакет MATLAB, что позволяет экономить время, необходимое на разработку и отладку программы, в пользу анализа физического содержания решаемой задачи.

Наш выбор из достаточно большого количества специализированных пакетов для математических вычислений в качестве базового программного средства MATLAB обусловлен тем, что:

- в пакет интегрирован мощный математический аппарат, позволяющий решать сложные задачи без вызова внешних процедур, который помогает находить решения:
  - линейных и нелинейных алгебраических уравнений и систем;
  - задачи Коши и краевой задачи для дифференциальных уравнений;
  - дифференциальных уравнений в частных производных;
  - задач статистической обработки данных (вычисление статистических параметров, интерполяция, аппроксимация, сглаживание и т. д.);
  - задач линейной алгебры (операции с матрицами и векторами);
  - задач поиска экстремумов функциональных зависимостей;
- пакет имеет мощные средства графического представления информации (функции, зависящие от одной переменной, полярные графики, графики поверхностей, карты линий уровня, векторные поля и т. д.);
- пакет снабжен средствами анимации, что позволяет рассматривать временну́ю эволюцию математических моделей в динамике и т. д.;
- в пакет интегрирован математический аппарат, реализующий символьные вычисления.

Примеры физических систем и процессов, рассмотренных в этой книге, требуют, с одной стороны, привлечения минимального набора численных методов, с другой, представляют несомненный физический интерес. Приоритет физического содержания задачи определил структуру изложения материала книги: физическая постановка задачи, обсуждение численных методов и алгоритмов, необходимых для ее решения, описание функций и последовательностей команд, позволяющих реализовать данные методы в MATLAB, задачи для самостоятельной работы. Отметим, что книга не является учебником по курсу «Численные методы», поэтому мы не даем в ней строгое с математической точки зрения обоснование применяемым численным методам, но ограничиваемся обсуждением вопросов их практического использования.

При написании этой книги, равно как и любой другой, в которой описывается использование какого-либо математического пакета для решения конкретных задач, перед автором стояла достаточно сложная проблема, связанная с необходимостью одновременного изложения конкретных математических методов и соответствующих приемов работы с математическим пакетом. Здесь был использован подход «от задачи», в котором автор ограничился краткими описаниями основных приемов работы с MATLAB, помещенными в приложения, а в ходе решения конкретной задачи сообщает необходимые дополнительные сведения по работе с пакетом. Данный подход, как показывает личный опыт автора, оказывается более эффективным в отличие от подхода, основанного на изучении различных справочников без привязки к конкретной задаче.

Отметим, что высказанная точка зрения ни в коей мере не ставит под сомнение необходимость издания соответствующей справочной и учебной литературы по MATLAB. Однако при ее выборе читателю следует проявлять определенную осторожность, так как в ряде случаев включенный в них материал оказывается плохо отредактированным, а в некоторых местах неотредактированным, а потому лишенным всякого смысла, вариантом электронного перевода на русский язык англоязычной документации MATLAB. Примером подобной литературы может быть книга [3], по мнению ее авторов являющаяся «одновременно научной монографией по математической системе MATLAB и руководством пользователя» [3, с. 23], которая, однако, никоим образом не ответствует требованиям, предъявляемым к научным монографиям. Кроме того, практически все примеры использования функций и программ, приводимые авторами [3], заимствованы (при этом некоторые из них с ошибками) из фирменной документации MATLAB. В этой связи, представляется уместным посоветовать пользователям математического пакета при выборе соответствующей литературы помнить бессмертный афоризм Козьмы Пруткова: «Бди!»

Первое издание книги «Компьютерное моделирование физических процессов в пакете MATLAB» было выпущено в 2003 г. тиражом 2000 экземпляров и на сегодняшний день стало библиографической редкостью. Во второе издание книги включены разделы, в которых рассматриваются нелинейные физические системы и алгоритмы построения фрактальных объектов. Эти дополнения автор считает наиболее важными, так как нелинейные математические модели, появившись впервые именно в физике, сегодня вышли далеко за ее пределы и используются в различных науках (не только естественных, но и гуманитарных), что привело к возникновению во второй половине XX в. появлению новой науки, названной «Нелинейной динамикой».

Кроме того, ко второму изданию прилагается CD-диск, на котором в папке «`Файлы MATLAB`» размещены компьютерные реализации всех обсуждавшихся в книге m-файлов и m-функций. Для удобства работы m-файлы и m-функции, описанные в одной главе, размещены в соответствующей папке, вложенной в папку «`Файлы MATLAB`». Название папки соответствует номеру главы (например, «`Glava1`»). Для выполнения выбранного m-файла и m-функции с CD-диска необходимо в качестве текущего каталога в поле «`Current Directory`» главного окна MATLAB указать, действуя стандартным образом, каталог: «`Имя CD-диска:\Файлы MATLAB\Имя вложенной папки`» (например, «`D:\Файлы MATLAB\Glava1`»).

Далее, в соответствии с описанным в выбранном разделе синтаксисом выполнить необходимую команду, введя ее имя в командном окне MATLAB и нажав клавишу `<Enter>`.

В заключение необходимо отметить, что организация нетривиального использования компьютеров в процессе обучения физике далеко не простая задача, для решения которой потребуется не один год. Автор надеется, что настоящая книга будет определенным вкладом в решение этой задачи, и с благодарностью примет замечания, предложения и советы, которые можно направить по e-mail: `sergey_porshnev@mail.ru`.

ЛИТЕРАТУРА

1. *Гулд Х., Тобочник Я.* Компьютерное моделирование в физике: В 2 ч. М.: Мир, 1990. Ч. 1: 349 с. Ч. 2: 400 с.

2. *Кунин С.* Вычислительная физика. М.: Мир, 1992.

3. *Дьяконов В. П., Абраменкова И. В.* MATLAB с пакетами расширений. М.: Изд-во «Нолидж», 2001.

Глава 1

МОДЕЛИРОВАНИЕ ОТНОСИТЕЛЬНЫХ ДВИЖЕНИЙ В КЛАССИЧЕСКОЙ МЕХАНИКЕ

1.1. ВВЕДЕНИЕ

В классической механике, для описания механических явлений, в качестве первого шага необходимо выбрать ту или иную систему отсчета. В различных системах отсчета законы движения имеют, вообще говоря, различный вид. Если взять произвольную систему отсчета, то может оказаться, что законы движения даже простых явлений будут выглядеть весьма сложно. Для нахождения траекторий относительных движений в классической механике используется предположение об абсолютности времени во всех системах отсчета (как инерциальных, так и неинерциальных). Используя данное предположение, рассмотрим движение одной и той же точки в двух различных системах отсчета K и K', из которых вторая движется относительно первой с произвольной скоростью $\vec{V}(t) = \frac{d}{dt}\vec{R}(t)$ ($\vec{R}(t)$ — радиус-вектор, описывающий положение точки начала системы координат K' относительно системы отсчета K). Будем описывать движение точки в системе K' радиусом-вектором $\frac{d}{dt}\vec{r}(t)$, направленным из начала координат системы K' в текущее положение точки. Тогда движение рассматриваемой точки относительно системы отсчета K описывается радиусом-вектором $\vec{r}(t)$:

$$\vec{r}(t) = \vec{r}\,'(t) + \vec{R}(t), \tag{1.1}$$

а относительная скорость $\vec{v}(t)$

$$\vec{v}(t) = \frac{d}{dt}\vec{r}(t) = \frac{d}{dt}\vec{r}\,'(t) + \frac{d}{dt}\vec{R}(t), \tag{1.2}$$

где $\frac{d}{dt}\vec{r}\,'(t)$ — скорость точки относительно системы отсчета K', $\frac{d}{dt}\vec{R}(t)$ — скорость движения системы отчета K' относительно системы отсчета K (рис. 1.1).

Таким образом, для нахождения закона движения точки в произвольной системе отсчета K необходимо:

1) задать закон движения точки относительно системы отсчета K' (функцию $\vec{r}\,'(t)$);

2) задать закон движения системы отсчета K' относительно системы отсчета K (функцию $\vec{R}(t)$);

3) определить закон движения точки относительно системы отсчета K в соответствии с (1.1).

Рис. 1.1. К постановке задачи об описании относительных движений

Возможна другая постановка данной задачи, в которой по известному закону движения материальной точки $\vec{r}(t)$ относительно некоторой системы отсчета K и по известному закону движения $\vec{R}(t)$ системы отсчета K' относительно системы отсчета K требуется найти закон движения $\vec{r}'(t)$ материальной точки в системе отсчета K'. Очевидно, что решение задачи в такой постановке дается следующими формулами:

$$\vec{r}'(t) = \vec{r}(t) - \vec{R}(t), \tag{1.3}$$

$$\vec{v}'(t) = \frac{d}{dt}\vec{r}(t) - \frac{d}{dt}\vec{R}(t). \tag{1.4}$$

Несмотря на то что решение задачи о нахождении траектории относительного движения исчерпывается формулами (1.1)–(1.4), построение конкретных траекторий движения без использования ПК может вызвать определенные трудности, а в ряде случаев привести к неправильным результатам. Ниже мы рассмотрим две поучительные, на наш взгляд, задачи, соответствующие обеим постановкам, и продемонстрируем в ходе их решения возможности, заложенные в MATLAB для построения графиков функций, зависящих от одной переменной.

1.2. ПОСТРОЕНИЕ ОРБИТЫ ЛУНЫ В ГЕЛИОЦЕНТРИЧЕСКОЙ СИСТЕМЕ ОТСЧЕТА

С точки зрения кинематического подхода эта задача соответствует первой постановке задачи об относительном движении. В гелиоцентрической системе отсчета (система K) Земля движется по окружности радиуса $R_1 = 1,496 \cdot 10^8$ *км* (период обращения $T_1 = 3,156 \cdot 10^7$ *с*). Луна в свою очередь движется вокруг Земли (система K) по окружности радиуса $R_2 = 3,844 \cdot 10^5$ *км* (период обращения $T_2 = 2,36 \cdot 10^6$ *с*). Как известно [1, 2], при движении материальной точки по окружности радиуса R с постоянной угловой скоростью w координаты радиуса-вектора, проведенного из начала координат к текущему положению точки, меняются по закону

$$\vec{R}(t) = \begin{pmatrix} R\cos(\omega\, t + \varphi_0) \\ R\sin(\omega\, t + \varphi_0) \end{pmatrix} = \begin{pmatrix} R\cos\left(\frac{2\pi}{T}t + \varphi_0\right) \\ R\sin\left(\frac{2\pi}{T}t + \varphi_0\right) \end{pmatrix}, \tag{1.5}$$

где φ_0 — начальная фаза, характеризующая положение частицы в момент времени $t = 0$, которую в дальнейшем мы будем полагать равной нулю. Заменяя в (1.5) R на R_1, R_2 и подставляя в (1.1), получаем зависимость радиуса-вектора Луны в гелиоцентрической системе координат от времени:

$$\vec{r}(t) = \begin{pmatrix} x(t) \\ y(t) \end{pmatrix} = \begin{pmatrix} R_2 \cos\left(\frac{2\pi}{T_2}t\right) + R_1 \cos\left(\frac{2\pi}{T_1}t\right) \\ R_2 \sin\left(\frac{2\pi}{T_2}t\right) + R_1 \sin\left(\frac{2\pi}{T_1}t\right) \end{pmatrix}. \tag{1.6}$$

Выражение (1.6) задает орбиту Луны $y = y(x(t))$ в параметрической форме, где параметром является время. Предваряя дальнейшее чтение параграфа, где орбита Луны построена с использованием MATLAB, рекомендуем качественно оценить ее вид.

Для построения искомой орбиты в MATLAB в режиме непосредственного вычисления необходимо передать пакету следующую последовательность команд, сохраненную в файле **Glava1_1.m**. (В MATLAB часть строки, расположенная после знака %, является комментарием и при самостоятельном вводе команд может быть опущена.)

```
>>  R1=1.496*10^8; % задание численного значения
                   % радиуса орбиты Земли
>>  T1=3.156*10^7; % задание численного значения периода
                   % обращения Земли вокруг Солнца
>>  R2=3.844*10^5; % задание численного значения
                   % радиуса орбиты Луны
>>  T2=2.360*10^6; % задание численного значения
                   % периода обращения Земли  вокруг Земли
>>  t=0:T1/1000:T1; % задание дискретной переменной,
                    % изменяющейся от 0 до T1 с шагом T1/1000
>>  Xz=R1*cos(2*pi*t/T1); % вычисление x-й координаты
                          % радиуса-вектора Земли
>>  Yz=R1*sin(2*pi*t/T1); % вычисление y-й координаты
                          % радиуса-вектора Земли
>>  Xm=R2*cos(2*pi*t/T2); % вычисление x-й координаты
                          % радиуса-вектора Луны
                          % в системе координат, связанной с Землей
>>  Ym=R2*sin(2*pi*t/T2); % вычисление y-й координаты
                          % радиуса-вектора Луны
                          % в системе координат, связанной с Землей
>>  Xotn=Xz+Xm; % вычисление x-й координаты
               % радиуса-вектора Луны
               % в гелиоцентрической системе координат
>>  Yotn=Yz+Ym; % вычисление y-й координаты
               % радиуса-вектора Луны
               % в гелиоцентрической системе координат
>>  plot(Xotn,Yotn); % визуализация орбиты Луны в гелиоцентрической
                     % системе координат (рис. (1.2))
```

```
>>  h = gca; % загрузка дискриптора графического окна
>>  set(h,'DataAspectRatioMode','manual'); % задание режима
                                           % отображения
                                           % пропорциональных
                                           % отрезков по обеим
                                           % координатным осям
```

Рис. 1.2. Орбита Луны в гелиоцентрической системе координат.
(Подробное описание интерфейса окна **Figure No 1** приведено в Приложении С)

Вопрос: *Соответствует ли вид орбиты, представленной на рис. 1.2, орбите, ранее построенной вами?*

Для отображения на одном графике орбиты Земли и орбиты Луны в гелиоцентрической системе координат следует вместо команды

```
>> plot(Xotn,Yont)
```

ввести команду

```
>> plot(Xotn,Yont,Xz,Yz)
```

Фрагмент траекторий Земли и Луны в гелиоцентрической системе координат представлен на рис. 1.3.

Для получения информации о всех активных переменных, находящихся в памяти компьютера (рабочей области) в данный момент времени, используется команда `whos`. На рис. 1.4 показана информация, выведенная данной командой на монитор компьютера, после выполнения описанной выше последовательности команд.

Отметим, что постоянные в MATLAB трактуются как матрицы размерности 1×1. Для сохранения значений переменных, находящихся в рабочей области, используется

Рис. 1.3. Фрагмент траекторий движения Луны (1) и Земли (2) в гелиоцентрической системе координат

команда

```
>> save имя_файла
```

или функция

```
>> save(имя_файла)
```

Команда save имеет несколько различных форм:

`>> save имя_файла` — записываются все переменные рабочей области в файл «имя_файла» с расширением .m;

`save имя_файла X` — записывается только значение переменной X;

`save имя_файла X Y Z` — записываются только значения переменных X, Y и Z.

После записи команды `save` также можно указывать ключи, уточняющие формат записи файлов:

`-mat` — двоичный MAT-формат, используемый по умолчанию;

`-ascii` — ASCII-формат единичной точности (8 цифр);

`-ascii -double` — ASCII-формат двойной точности (16 цифр);

`-ascii -double -tabs` — ASCII-формат двойной точности с разделителем и метками табуляции;

`V4` — запись MAT-файла в стандарте версии MATLAB 4.0;

`-append` — добавление в существующий MAT-файл.

Для загрузки ранее проведенного сеанса работы с MATLAB используется команда

```
>> load имя_файла
```

или функция

```
>> load(имя_файла)
```

```
>> whos
Name    Size     Bytes   Class

R1      1x1      8       double array
R2      1x1      8       double array
T1      1x1      8       double array
T2      1x1      8       double array
Xm      1x1001   8008    double array
Xotn    1x1001   8008    double array
Xz      1x1001   8008    double array
Ym      1x1001   8008    double array
Yont    1x1001   8008    double array
Yz      1x1001   8008    double array
ans     0x0      0       char array
t       1x1001   8008    double array

Grand total is 7011 elements using 56088  bytes
```

Рис. 1.4. Получение информации об активных переменных, находящихся в памяти компьютера после выполнения описанной последовательности команд

Приведенный выше протокол команд можно сохранить в виде файла на диске для последующего анализа решения или использования, как основы программы сценария, используемой при решении подобных задач, командой

```
>> diary имя_файла
```

или функцией

```
>> diary(имя_файла)
```

Например, для записи в файл `Moon.m` протокола приведенных выше команд необходимо ввести следующую команду:

```
>> diary Moon.m % открытие файла для сохранения протокола
```

Далее каждая последовательно выполненная команда будет заноситься в файл `Moon.m`. Для приостановки записи выполняемых команд в файл используется команда

```
diary off
```

Отметим, что данная команда также записывается в файл `Moon.m`. Начиная с версии MATLAB 6.0 и выше, данный набор команд можно выполнить автоматически, набрав в командной строке имя файла `Moon` и нажав клавишу <Enter>.

Созданный нами документ позволяет расширить задачу и посмотреть, какой будет орбита Луны при различных значениях радиуса орбиты Луны и периода обращения. Например, на рис. 1.5 представлена орбита Луны в гелиоцентрической системе координат при $r = 3,844 \cdot 10^7$ *км*. Сравнивая орбиты Луны, представленные на рис. 1.2 и 1.5, обнаруживаем их существенные отличия. Для объяснения причины этих отличий необходимо сравнить линейные скорости движения Луны в первом и во втором случаях и линейную скорость движения Земли.

Рис. 1.5. Орбита Луны в гелиоцентрической системе координат при $r = 3,844 \cdot 10^7$ *км*

Так как направление линейной скорости движения Земли относительно Солнца и направление линейной скорости движения Луны относительно Земли меняются во времени, и при этом по величине эти скорости остаются постоянными, в качестве количественной характеристики соотношения линейных скоростей движения Луны и Земли в гелиоцентрической системе координат следует выбрать разность между модулем линейной скорости движения Земли и проекцией линейной скорости Луны на направление вектора линейной скорости Земли:

$$v(t) = \left|\vec{V}(t)\right| - \frac{\left(\vec{V}(t)\vec{v}(t)\right)}{\left|\vec{V}(t)\right|}, \qquad (1.7)$$

где $\vec{V}(t) = \begin{pmatrix} dX(t)/dt \\ dY(t)/dt \end{pmatrix}$ — вектор скорости движения Земли относительно Солнца, $\vec{v}(t) = \begin{pmatrix} dx(t)/dt \\ dy(t)/dt \end{pmatrix}$ — вектор скорости Луны относительно Земли.

Для визуализации искомой зависимости необходимо выполнить следующую последовательность команд, сохраненную в файле `Glava1_2.m`:

```
>>  R1=1.496*10^8; % задание численного значения
                   % радиуса орбиты Земли
>>  T1=3.156*10^7; % задание численного значения
                   % периода обращения Земли вокруг Солнца
>>  R2=3.844*10^5; % задание численного значения
                   % радиуса орбиты Луны
>>  T2=2.360*10^6; % задание численного значения
                   % периода обращения Луны вокруг Земли
>>  dt=T1/2000; % задание шага по времени
>>  t=0:dt:T1; % задание дискретной переменной,
               % изменяющейся от 0 до T1 с шагом T1/2000
>>  Xz=R1*cos(2*pi*t/T1); % вычисление x-й координаты
                          % радиуса-вектора Земли
>>  Yz=R1*sin(2*pi*t/T1); % вычисление y-й координаты
                          % радиуса-вектора Земли
>>  Xm=R2*cos(2*pi*t/T2); % вычисление x-й координаты
                          % радиуса-вектора Луны
                          % в системе координат, связанной с Землей
>>  Ym=R2*sin(2*pi*t/T2); % вычисление y-й координаты
                          % радиуса-вектора Луны
                          % в системе координат, связанной с Землей
```

```
>>  Xotn=Xz+Xm; % вычисление x-й координаты
               % радиуса-вектора Луны
               % в гелиоцентрической системе координат
>>  Yotn=Yz+Ym; % вычисление y-й координаты
               % радиуса-вектора Луны
               % в гелиоцентрической системе координат
>>  Vx=diff(Xz)/dt; % вычисление значений проекции
                   % скорости движения Земли на ось OX
                   % в гелиоцентрической системе координат
>>  Vy=diff(Yz)/dt; % вычисление значений проекции
                   % скорости движения Земли на ось OY
                   % в гелиоцентрической системе координат
>>  vx=diff(Xm)/dt; % вычисление значений проекций
                   % скорости движения Луны на ось OX
                   % в системе координат, связанной с Землей
>>  vy=diff(Ym)/dt; % вычисление значений проекций
                   % скорости движения Луны на ось OY
                   % в системе координат, связанной с Землей
>>  V=(Vx.^2+Vy.^2).^0.5...
       -(Vx.*vx+Vy.*vy)./(Vx.^2+Vy.^2).^0.5; % вычисление
                                             % значений
                                             % функции,
                                             % задаваемой
                                             % выражением (1.7)
>>  t1=0:dt:T1-dt;
>>  subplot(2,1,1); plot(Xz,Yz,Xotn,Yotn); % построение траектории
                                           % движения Луны и
                                           % траектории движения
                                           % Земли
                                           % в гелиоцентрической
                                           % системе координат
>>  subplot(2,1,2); plot(t1,V); % построение графика функции,
                               % задаваемой выражением (1.7)
```

Результаты выполнения приведенной выше последовательности команд представлены на рис. 1.6 ($R2 = 3,844 \cdot 10^7$ *км*), рис. 1.7 ($R2 = 3,844 \cdot 10^5$ *км*).

Анализ зависимостей $v_{otn}(t)$, представленных на рис. 1.6–1.7, позволяет объяснить причину отличий орбит. Функция $D(t)$ при $R2 = 3,844 \cdot 10^5$ *км* всегда положительна, то есть Луна всегда движется в направлении движения Земли и петли не образуются. При $R2 = 3,844 \cdot 10^7$ *км* величина $D(t)$ принимает отрицательные значения, то есть существуют моменты времени, в которые Луна движется в направлении противоположном направлению движения Земли, а потому орбита имеет петли.

В заключение сделаем ряд замечаний по поводу некоторых операторов и функций, использованных в данном разделе. Обратите внимание на способ задания дискретных переменных (векторов) `t`, `t1`. Здесь мы использовали знак `:`, являющийся одним из важнейших синтаксических знаков MATLAB. Этот знак, поставленный между двумя

Рис. 1.6. Траектория движения Луны в гелиоцентрической системе координат при $R2 = 3,844 \cdot 10^7$ *км* (верхняя зависимость), зависимость мгновенных значений разностей между модулем скорости Земли и проекцией скорости движения Луны на направление скорости движения Земли.

числами, задает вектор, компоненты которого принимают значения от меньшего числа до большего с шагом 1. Например, оператор `x = 0:9` задает целочисленный вектор `x = [0,1,2,3,4,5,6,7,8,9]`. Отметим, что в MATLAB также допускается явное задание вектора с помощью квадратных скобок. Если шаг изменения дискретной переменной отличается от единиц, то при ее определении следует указывать значение шага, аналогично тому как это было сделано выше в программах, приведенных в настоящем разделе.

Так как MATLAB является пакетом изначально ориентированным на матричные вычисления, основные объекты, с которыми оперирует пакет, являются векторами и матрицами. При обращении к функции, например, `y=cos(x)`, где x — вектор, значение которого определены выше, MATLAB проводит вычисления для каждого элемента вектора аргумента и присваивает их соответствующим компонентам вновь создаваемого вектора. Для построения двух графиков на одном чертеже были использованы команда `subplot(2,1,1)`, `subplot(2,1,2)`, позволившие разбить графическое окно на две отдельные части. В общем случае команда обращения к данной команде имеет следующий вид: `subplot(m,n,p)` или `subplot(m n p)`. Здесь значение `m` указывает, на сколько частей разбивается окно по вертикали, `n` указывает, на сколько частей окно разбивается по горизонтали, `p` — порядковый номер подокна, считая слева направо и сверху вниз. Команда `subplot( )` используется как для создания нового подокна, так и для перехода от одного подокна к другому. После вызова данной команды команда `plot( )` график и/или графики в соответствующем подокне. Технология создания интерфейса пользователя, позволяющего упростить процедуру вычислений, описана в Приложении С.

Рис. 1.7. Траектория движения Луны в гелиоцентрической системе координат при $R2 = 3,844 \cdot 10^5$ *км* (верхний рисунок), зависимость мгновенных значений разностей между модулем скорости Земли и проекцией скорости движения Луны на направление скорости движения Земли (нижний рисунок)

***Задача* 1.1**

1. *Определите предельное значение радиуса орбиты Луны, при котором не происходит появление петель. Как выглядит орбита Луны в гелиоцентрической системе координат в этом случае?*
2. *Зафиксируйте радиус орбиты Луны и постройте орбиту Луны при различных значениях периода обращения. Что можно сказать о размере петель в этом случае?*

1.3. ПОСТРОЕНИЕ ОРБИТЫ МАРСА В СИСТЕМЕ ОТСЧЕТА, СВЯЗАННОЙ С ЗЕМЛЕЙ

С точки зрения кинематического подхода эта задача соответствует второй постановке задаче об относительном движении. В гелиоцентрической системе отсчета (система K) Земля движется по окружности радиуса $R_1 = 1,496 \cdot 10^8$ *км*, период обращения $T_1 = 365,2$ *суток*, Марс двигается по эллипсу, большая полуось которого $a_M = 2,2 \cdot 10^8$ *км*, период обращения Марса $T_M = 689,98$ *суток*, эксцентриситет орбиты $= 0,093$ [3]. Движение Земли описывается радиусом-вектором $R(t)$, координаты которого вычисляются в соответствии с (1.5). В связи с тем, что орбита Марса является эллипсом зависимости $x = x(t)$, $y = y(t)$ от времени задаются параметрически [4]:

$$x(\xi) = a_M(\cos\xi - e), \tag{1.8}$$

$$y(\xi) = a_M\sqrt{1-e^2}\sin\xi, \tag{1.9}$$

$$t(\xi) = \frac{T_M}{2\cdot\pi}(\xi - e\cdot\sin\xi). \tag{1.10}$$

Полному обороту по эллипсу соответствует изменение параметра ξ от 0 до 2π.

Для построения орбиты Марса необходимо вычислить в одни и те же моменты времени координаты радиусов-векторов, описывающих положение Земли и Марса в гелиоцентрической системе отсчета, затем в соответствии с (1.3) вычислить координаты Марса в системе отсчета, связанной с Землей. Запишем эту последовательность действий более четко в виде вычислительного алгоритма:

1. Задание интервала изменения параметра ξ.
2. Разбиение интервала изменения параметра ξ на N последовательных интервалов.
3. Вычисление для каждого значения переменной ξ соответствующих значений времени и координат орбиты Марса x, y согласно (1.10), (1.8), (1.9).
4. Вычисление для каждого значения времени координат Земли в соответствии с (1.5).
5. Вычисление для каждого значения времени координат орбиты Марса в системе отсчета, связанной с Землей.
6. Построение орбиты Марса в системе отсчета, связанной с Землей.

Для построения искомой орбиты в MATLAB в режиме непосредственного вычисления необходимо выполнить следующую последовательность команд, сохраненную нами в файле `Glava1_3.m`:

```
>>  R1=1.496*10^8; % задание радиуса орбиты Земли
>>  T1=365.24;  % задание периода обращения
                % Земли вокруг Солнца в сутках
>>  Am=2.28*10^8; % задание радиуса орбиты Марса
>>  Tm=689.98; % задание периода обращения Марса
               % вокруг Солнца в сутках
>>  E=0.093; % эксцентриситет орбиты Марса
>>  Np=1000; % число точек на один оборот Марса
            % вокруг Солнца
>> dksi=(2*pi)/Np;  %  вычисление шага
                    %  изменения переменной ksi
>>  ksi=0:dksi:2*pi; % вычисление значений
                     % координат вектора ksi
>>  T=Tm/(2*pi)*(ksi-E*sin(ksi)); % вычисление значений
                                  % координат вектора
                                  % по формуле (1.10))
>>  Xm=Am*(cos(ksi)-E); % вычисление мгновенных
                        % значений проекции
                        % радиуса-вектора Марса
                        % на ось OX (формула (1.8))
```

```
>>  Ym=Am*((1-E.^2).^0.5)*sin(ksi); % вычисление
                                    % мгновенных значений проекции
                                    % радиуса-вектора Марса
                                    % на ось OY (формула (1.9))
>>  Xz=R1*cos(2*pi*T/T1); % вычисление мгновенных
                          % значений проекции
                          % радиуса-вектора Земли на ось OX
>>  Yz=R1*sin(2*pi*T/T1); % вычисление мгновенных
                          % значений проекции
                          % радиуса-вектора Земли на ось OY
>>  plot(Xz,Yz,Xm,Ym) % визуализация траекторий
                     % Земли и Марса
                     % в гелиоцентрической системе координат
```

Рис. 1.8. Орбиты Марса и Земли в гелиоцентрической системе координат

Результаты выполнения приведенной выше последовательности команд представлены на рис. 1.8. Для построения орбиты Марса в системе координат, связанной с Землей, необходимо выполнить следующую последовательность команд, сохраненную нами в файле `Glava1_4.m`:

```
>>  R1=1.496*10^8;  % задание радиуса орбиты Земли
>>  T1=365.24;  % задание периода обращения Земли
               % вокруг Солнца в сутках
>>  Am=2.28*10^8; % задание радиуса орбиты Марса
>>  Tm=689.98; % задание периода обращения Марса
              % вокруг Солнца в сутках
>>  E=0.093; % эксцентриситет орбиты Марса
>>  Np=1000; % число точек на один оборот Марса
            % вокруг Солнца
>>  K=9;  % число оборотов Марса вокруг Солнца
```

```
>>  dksi=(2*pi)/Np*K;  % вычисление шага dksi
                      % изменения переменной
>>  ksi=0:dksi:2*pi*K; % вычисление значений
                       % координат вектора
>>  T=Tm/(2*pi)*(ksi-E*sin(ksi)); % вычисление значений координат
                                  % вектора T по формуле (1.10)
>>  Xm=Am*(cos(ksi)-E); % вычисление мгновенных значений проекции
                        % радиуса-вектора Марса
                        % на ось oX по формуле (1.8)
>>  Ym=Am*((1-E.^2).^0.5)*sin(ksi); % вычисление мгновенных
                                    % значений проекции
                                    % радиуса-вектора Марса
                                    % на ось OY по формуле (1.9)
>>  Xz=R1*cos(2*pi*T/T1); % вычисление мгновенных
                          % значений проекции
                          % радиуса-вектора Земли на ось OX
>>  Yz=R1*sin(2*pi*T/T1); % вычисление мгновенных
                          % значений проекции
                          % радиуса-вектора Земли на ось OY
>>  Xotn=Xm-Xz; % вычисление мгновенных значений проекции
               % радиуса-вектора Марса на ось OX
               % в системе координат, связанной с Землей
>>  Yotn=Ym-Yz; % вычисление мгновенных значений
               % проекции радиуса-вектора Марса
               % на ось oY в системе координат,
               % связанной с Землей
>>  plot(Xotn,Yotn,...             % визуализация орбиты Марса
             Xotn(1),Yotn(1),... % визуализация начального
                                 % положения Марса
               'ks',...         % режим рисования одних маркеров
               'MarkerEdgeColor','b',...  % задание цвета
                                          % границы маркера
               'MarkerFaceColor','g',... % задание цвета
                                         % закраски маркера
               'MarkerSize',5); % задание размера маркера
```

Результаты выполнения приведенной выше последовательности команд представлены на рис. 1.9. Отметим, что в MATLAB имеется возможность продемонстрировать движение Марса в системе координат, связанной с Землей, в динамике. Для этого достаточно заменить команду `plot( )` командой `comet(Xotn,Yotn)`.

Еще одной важной характеристикой движения Марса (в первую очередь для межпланетных космических полетов) является расстояние между Землей и Марсом $s(t)$, которое определяется модулем радиуса-вектора, описывающего положение Марса в системе отсчета, связанной с Землей. Для построения данной зависимости следует выполнить следующую последовательность команд, сохраненную нами в файле `Glava1_5.m`:

Рис. 1.9. Траектория движения Марса в системе координат, связанной с Землей

```
>>  R1=1.496*10^8;  % задание радиуса орбиты Земли
>>  T1=365.24;  % задание периода обращения Земли
               % вокруг Солнца в сутках
>>  Am=2.28*10^8; % задание радиуса орбиты Марса
>>  Tm=689.98; % задание периода обращения Марса
              % вокруг Солнца в сутках
>>  E=0.093; % эксцентриситет орбиты Марса
>>  Np=1000; % число точек на один оборот
            % Марса вокруг Солнца
>>  K=9;  % число оборотов Марса вокруг Солнца
>>  dksi=(2*pi)/Np*K;  % вычисление шага dksi
                      % изменения переменной ksi
>>  ksi=0:dksi:2*pi*K; % вычисление значений
                       % координат вектора ksi
>>  T=Tm/(2*pi)*(ksi-E*sin(ksi)); % вычисление значений
                                  % координат вектора T
                                  % по формуле (1.10)
>>  Xm=Am*(cos(ksi)-E); % вычисление мгновенных
```

```
                          % значений проекции
                          % радиуса-вектора Марса
                          % на ось oX по формуле (1.8)
>>  Ym=Am*((1-E.^2).^0.5)*sin(ksi); % вычисление
                                    % мгновенных значений
                                    % проекции радиуса-вектора
                                    % Марса на ось oY
                                    % по формуле(1.9)
>>  Xz=R1*cos(2*pi*T/T1); % вычисление мгновенных значений
                          % проекции радиуса-вектора Земли
                          % на ось OX
>>  Yz=R1*sin(2*pi*T/T1); % вычисление мгновенных значений
                          % проекции радиуса-вектора Земли
                          % на ось OY
>>  Xotn=Xm-Xz; % вычисление мгновенных значений проекции
                % радиуса-вектора Марса на ось OX
                % в системе координат, связанной с Землей
>>  Yotn=Ym-Yz; % вычисление мгновенных значений проекции
                % радиуса-вектора Марса на ось OY
                % в системе координат, связанной с Землей
>>  R=(Xotn.^2+Yotn.^2).^0.5; % вычисление мгновенных значений
                              % расстояния между Землей
                              % и Марсом
>>  plot(T,R) % визуализация зависимости расстояния
              % между Землей и Марсом
              % от времени
```

Рис. 1.10. Зависимость расстояния между Землей и Марсом от времени, измеряемого в земных сутках

Результаты выполнения приведенной последовательности команд представлены на рис. 1.10.

Анализ зависимости, представленной на рис. 1.10, показывает, что расстояние между Землей и Марсом является сложной периодической функцией времени. Если воспользоваться терминологией теории сигналов [5], то о зависимости $s(t)$ можно сказать, что она представляет собой амплитудно-модулированный сигнал, который принято представлять в виде произведения двух функций — высокочастотной (несущей) и низкочастотной функции, задающей амплитудную модуляцию (огибающей):

$$u(t) = (\overline{u} + a\sin(\omega_1 t)) \cdot (1 + \Delta a \sin(\omega_2 t)), \qquad (1.11)$$

где $\overline{u}$ — постоянная составляющая функции $u(t)$; a — амплитуда сигнала; ω_1 — частота несущей; Δa — амплитуда функции, задающая глубину амплитудной модуляции, ω_2 — частота модулирующей функции.

Из рис. 1.10 видно, что период несущей составляет $T_1 = 2\pi/\omega_1 \approx 2$ года, период модулирующей функции $T_2 = 2\pi/\omega_2 \approx 17$ лет. Наличие функции в функции $s(t)$ периодической составляющей с периодом ≈ 2 года вполне ожидаемо. Действительно, в первом приближении орбиту Марса можно рассматривать как окружность, поэтому функция $s(t)$ принимает следующий вид:

$$s(t) = \sqrt{R_M^2 + R1^2 - 2R_M R1 \cos\left[2\pi\left(\frac{T_M - T1}{T_M T1}\right)t\right]}. \tag{1.12}$$

Из (1.12) видно, что функция $s(t)$ является периодической, период функции $s(t)$

$$T_S = \frac{T_M T1}{T_M - T1}, \tag{1.13}$$

Поэтому в качестве единицы измерения периода функция $s(t)$ наиболее удобно выбрать земной год. В выбранных единицах измерения выражение (1.13) принимает следующий простой вид:

$$T_S = \frac{T_M}{T_M - T1}, \tag{1.14}$$

позволяющий вычислять длительность периода в земных годах. Для рассматриваемого случая $T_S \approx 2,1247$[1].

***Задача* 1.2**

1. *Получите самостоятельно выражение* (1.12)*, описывающее зависимость расстояния между Землей и Марсом от времени, в предположении, что орбита последнего является окружностью.*
2. *Постройте график функции, задаваемой выражением* (1.12).
3. *Оцените, используя рис. 1.7 или график в созданном вами документе, значения параметров ($\overline{u}$, a, Δa) в выражении* (1.11) *и напишите явный вид функции $s(t)$ в этом случае. Для проверки правильности выражения, постройте график $s(t)$ и сравните его с соответствующей зависимостью, представленной на рис. 1.7.*
4. *Исследуйте зависимость функции $s(t)$ от параметров орбитального движения Земли и Марса. От каких параметров зависят период несущей, глубина и период огибающей функции $s(t)$?*

***Задача* 1.3**

Используя астрономические данные, постройте траектории движения других планет Солнечной системы в системе отсчета, связанной с Землей, предположив, что их орбиты плоскости лежат в одной плоскости.

[1] Отметим, что именно это свойство относительного движения Земли и Марса явилось причиной форсированной подготовки в первой половине 1990-х гг. российских космических аппаратов по программе «Фобос», закончившейся, к сожалению, неудачей.

***Задача* 1.4**

В качестве единиц измерения расстояния можно использовать, например, радиус орбиты Земли, времени — период обращения Земли вокруг Солнца. Запишите в выбранной системе единиц соответствующие формулы и проведите вычисления.

***Задача* 1.5**

Изучите технологию создания графического интерфейса пользователя, описанную в Приложении С, и создайте аналогичный интерфейс для моделирования траекторий движения планет Солнечной системы.

ЛИТЕРАТУРА К ГЛАВЕ 1

1. *Фейман Р., Лейтон Р., Сэндс М.* Феймановские лекции по физике. Т. 1–2. М.: Мир, 1976.
2. *Матвеев А. Н.* Механика и теория относительности. М.: Высшая школа, 1986.
3. Физическая энциклопедия. Т. 3. М.: Большая российская энциклопедия, 1992.
4. *Ландау Л. Д., Лифшиц Е. М.* Курс теоретической физики. Механика. М.: Физматгиз, 2000.
5. *Баскаков С. И.* Радиотехнические цепи и сигналы. М.: Высшая школа, 1988.

Глава 2

ФИЗИЧЕСКИЕ ПРОЦЕССЫ, ОПИСЫВАЕМЫЕ ДИФФЕРЕНЦИАЛЬНЫМИ УРАВНЕНИЯМИ ПЕРВОГО ПОРЯДКА

2.1. МОДЕЛИРОВАНИЕ ОСТЫВАНИЯ НАГРЕТЫХ ТЕЛ

Природа переноса тепла от нагретого тела в окружающий его воздух весьма сложна и включает в себя механизмы конвекции, излучения, испарения и теплопроводности. Однако при небольшой разности температур между нагретым телом и окружающей средой для описания процесса остывания можно использовать простую модель, впервые предложенную И. Ньютоном. В данной модели температура холодильника (окружающей) принимается постоянной, а скорость передачи тепла от нагретого тела к холодильнику пропорциональной разности температур между ними. Это утверждение в математической формулировке записывается на языке дифференциальных уравнений (ДУ) в следующем виде:

$$\frac{dT}{dt} = -r(T - T_s), \tag{2.1}$$

где T — температура нагретого тела, T_s — температура окружающей среды, r — «коэффициент остывания». Значение «коэффициента остывания» зависит от механизма передачи, площади нагретого тела и тепловых свойств самого тела.

Знак минус в (2.1) появляется во избежание физически ненаблюдаемой ситуации, когда $T > T_s$. Соотношение (2.1) называется законом теплопроводности Ньютона.

Так как в уравнение (2.1) входит только первая производная неизвестной функции $T(t)$, его относят к ДУ первого порядка. В связи с тем, что большое количество физических процессов описывается данным классом ДУ, важно научиться находить их решения. По определению решением ДУ первого порядка, разрешенного относительно производной,

$$\frac{dy}{dx} = f(x, y(x)) \tag{2.2}$$

называется дифференцируемая функция $y = \varphi(x)$, удовлетворяющая этому уравнению, то есть такая, что $\varphi'(x) \equiv f(x, \varphi(x))$ тождественно на некотором участке изменения [1]. Задача нахождения решения уравнения (2.2), удовлетворяющего начальному условию $\varphi(x_0) = y_0$, называется задачей Коши.

В общем случае уравнение (2.2) аналитического решения, выражающегося через элементарные функции, не имеет. Кроме того, даже при наличии аналитического решения представление его в графическом виде существенно облегчает его анализ.

Отмеченные причины являются весьма важным аргументом в пользу нахождения приближенных численных решений ДУ.

2.2. АЛГОРИТМ ЭЙЛЕРА

При численном решении вместо исходного ДУ ищется решение конечно-разностного ДУ. Переход к конечно-разностному уравнению от уравнения (2.2) осуществляется следующим образом. Вместо точного значения производной рассматривают ее разностный аналог:

$$\frac{dy}{dx} \to \frac{y(x+\Delta x) - y(x)}{\Delta x}, \tag{2.3}$$

где Δx — достаточно малая величина. Тогда в конечных разностях уравнение (2.2) принимает следующий вид:

$$\frac{y(x+\Delta x) - y(x)}{\Delta x} = f(x, y(x)). \tag{2.4}$$

Из (2.4) сразу получаем формулу для нахождения значения функции $y = y(x)$ в точке $x + \Delta x$:

$$y(x+\Delta x) = y(x) + \Delta x f(x, y(x)). \tag{2.5}$$

Из (2.5) видно, как построить алгоритм для решения задачи Коши:

1. Задать начальные условия (x_0, y_0).
2. Вычислить в точке $x_1 = x_0 + \Delta x$ значение функции $y = y(x)$, определяемое выражением (2.5):
$$y_1 = y_0 + \Delta x f(x_0, y_0). \tag{2.6}$$
3. Повторить процедуру, описанную в п. 2, и найти значение функции $y = y(x)$ в точке $x_2 = x_1 + \Delta x$ и так далее для всех значений переменной $x_n = x_0 + i\Delta x$.

Рис. 2.1. Геометрическая интерпретация метода Эйлера

Наиболее удобно записать данный алгоритм в виде следующей итерационной формулы:

$$y_i = y_{i-1} + \Delta x f(x_{i-1}, y(x_{i-1})), \quad i = 1, 2, \ldots \tag{2.7}$$

Выражение (2.7) реализует метод, называющийся методом касательных, или методом Эйлера. Из (2.4) видно, что метод будет давать хорошее приближение к «истинному» значению функции $y(x)$, если приращение аргумента Δ достаточно мало. Величина шага Δ зависит от конкретной задачи. Геометрическая интерпретация (2.7) приведена на рис. 2.2, из которого понятно, что значение в n-й точке

оценивается по значению функции и ее производной в $n-1$-й точке, при этом принимается, что значение производной (то есть угол наклона касательной) на отрезке $[x_{i-1}; x_i]$ остается неизменным. Невыполнение данного условия приводит к отклонению численного решения от точного решения. Данное отклонение может быть уменьшено уменьшением Δx.

2.3. ПРОГРАММА ДЛЯ РЕШЕНИЯ ДИФФЕРЕНЦИАЛЬНЫХ УРАВНЕНИЙ ПЕРВОГО ПОРЯДКА

Приступая к разработке программы вне зависимости от использованного языка программирования, необходимо разбить всю задачу на последовательность независимых заданий, соответствующих алгоритму, описанному в предыдущем разделе. Программа должна состоять из следующих блоков:

1. Задание начальных условий.
2. Задание функции $f(x, y(x))$.
3. Задание отрезка, на котором ищется решение, и шага интегрирования. (Отметим, что на практике оказывается более удобным задавать не шаг интегрирования, а количество интервалов, на которые разбивается отрезок интегрирования, а затем вычислять значение шага.)
4. Вычисление координат точек, в которых ищется решение дифференциального уравнения.
5. Решение уравнения (2.2) методом Эйлера.
6. Вывод результатов.

Для примера рассмотрим решение задачи Коши для ДУ

$$y' = -y^2 + x \tag{2.8}$$

с начальным условием $y(0) = 1$.

Для решения поставленной задачи в MATLAB потребуется создание двух файлов, называемых m-файлами, потому что имена этих файлов имеют вид `<Имя>.m`. Отметим, что бóльшая часть времени пользователя MATLAB состоит в создании, редактировании и выполнении m-файлов. В MATLAB существуют два вида файлов: файлы-программы и файлы-функции. Файлы программы состоят из последовательности обычных операторов MATLAB. Примером программы является, например, файл сценария, содержащий последовательность команд, выполненных в предыдущем разделе при моделировании относительных движений. Переменные, используемые в программе, являются глобальными — они изменяют значения переменных с аналогичным названием в рабочей области текущей сессии. Файлы-функции содержат описания специфических функций, созданных пользователем для решения конкретной задачи. Они дают возможность пользователю фактически расширить возможности MATLAB, поскольку имеют тот же статус, что и другие функции пакета.

Любой модуль, содержащий функцию, возвращающую один выходной параметр, имеет следующую структуру:

```
function var = name_function(список_параметров)
% Комментрии
Блок команд, реализующих вычисление функции
var = выражение;  % данная строчка вводится,
                  % если функция возвращает
                  % результат вычислений
```

Свойства m-файла, содержащего m-функцию:

1. m-файл начинается с объявления типа `function`, после которого указывается имя переменной, являющейся параметром, возвращаемым функцией, имя самой функции и список формальных параметров, передаваемых в функцию при ее вызове.

2. Если последняя строка функции имеет вид `var = выражение`, то в качестве значения функции возвращается значение выражения, занесенного в переменную `var`.

3. Все переменные, используемые в теле функции, являются локальными переменными, то есть их определение действует только внутри данной функции.

4. Для использования глобальных переменных их список приводится в строке, предваряющей блок проведения вычислений:

   ```
   function var = name_function(список_параметров)
   % Комментрии
   global var1, var2, var3 % объявление глобальных переменных
   Блок команд, реализующих вычисление функции
   var = выражение;  % данная строчка вводится,
                     % если функция возвращает
                     % результат вычислений
   ```

5. Функция является самостоятельным программным модулем, взаимодействующим с другими программными модулями через входные, выходные и глобальные переменные.

6. В файле функции допускаются комментарии, начинающиеся символом %.

7. При вызове файла-функции сначала происходит его компиляция, а затем исполнение. Машинный код m-функции хранится в рабочей области MATLAB.

8. Обращение к функции осуществляется указанием ее имени и значений переменных, перечисленных в списке формальных параметров:

```
имя_переменной = name_function(список_параметров)
name_function(список_параметров)
```

Любой модуль, содержащий функцию, возвращающую несколько выходных параметров, имеет следующую структуру:

```
function [var1,var2,...] = name_function(список_параметров)
% комментарии
Блок команд, реализующих вычисление функции
var1 = выражение1;
var2 = выражение2;
var3 = выражение3;
... ... ... ... ...
```

Обращение к функции, возвращающей несколько выходных параметров:

```
[var1,var2,...] = name_function(список_параметров)
```

После этого переменные `var1,var2,...` становятся определенными и их можно использовать в последующих математических выражениях.

При обращении к функции, возвращающей несколько выходных параметров, в виде: `имя_переменной = name_function(список_параметров)` в переменную `имя_переменной` возвращается только значение переменной `var1`.

Создадим, следуя перечисленным выше правилам, функции, позволяющие найти численные решения задачи Коши для обыкновенного ДУ первого порядка. Первый из создаваемых нами файлов будет содержать функцию, стоящую в правой части уравнения, второй — реализацию вычислительной схемы метода Эйлера. Для создания m-файла необходимо запустить встроенный в MATLAB текстовый редактор, выбрав последовательно следующие пункты меню: **File** → **New** → **M-File**, приемы редактирования текста в котором аналогичны любому редактору текстов, работающему под управлением операционной системы Windows. Затем набрать во вновь создаваемом m-файле следующий текст:

```
function F = f(x,y)
% f(x,y) - функция, стоящая в правой части
% дифференциального уравнения (2.8)
F = x-y^2;
```

и сохранить его на жестком диске под именем `f.m`.

Далее аналогичным образом создаем файл, содержащий следующие команды, реализующие метод Эйлера:

```
function [X,Y]=Euler(y0,x0,x1,N)
% функция, возвращающая численные
% решения дифференциального
% уравнения первого порядка
% методом Эйлера
dx=(x1-x0)/N; % вычисление шага
              % интегрирования
```

```
% задание начальных условий
x(1)=x0;
y(1)=y0;
% реализация вычислительной схемы
% метода Эйлера
for i=1:N
   x(i+1)=x(1)+dx*i;
   y(i+1)=y(i)+dx*f(x(i),y(i));
end;
% возвращение результатов вычислений
X=x;
Y=y;
```

и сохраняем его на диске под именем `Euler.m`. Функция `Euler` возвращает два вектора: `X`, `Y` — векторы, содержащие значения координат узлов сетки, на которой ищется решение ДУ, и значения решения ДУ в данных узлах соответственно.

После создания файлов `f.m` и `Euler.m` для получения решения ДУ и его визуализации следует в командном окне пакета ввести следующие команды, сохраненные нами в файле `Glava2_1.m`:

```
% задание начальных условий
>> x0 = 0;
>> y0= 1;
>> x1 = 10; % задание правой
            % границы отрезка
>> N = 100; % задание числа узлов
            % координатной сетки
>> [x,y]=Euler(y0,x0,x1,N); % вычисление
                            % численного решения
                            % задачи Коши
                            % уравнения (2.8)
>> plot(x,y) % визуализация численного
             % решения задачи Коши
             % уравнения (2.8)
```

Отметим, что приведенную выше последовательность команд можно сначала ввести, для добавления новой строки в командном окне MATLAB одновременно нажимая клавиши **Shift** и **Enter**, и затем для выполнения всей последовательности команд нажать клавишу **Enter**.

Искомое численное решение дифференциального уравнения представлено на рис. 2.2.

Теперь, когда создана программа для численного решения ДУ первого порядка, реализующая метод Эйлера, необходимо обсудить вопрос о точности получаемого решения. Так как мы заменили дифференциальное уравнение его разностным аналогом, то естественно ожидать некоторое отличие численного решения от «истинного» решения ДУ.

Рис. 2.2. Решение дифференциального уравнения $y' = -y^2 + x$ с начальным условием $y(0) = 1$

В качестве оценки погрешности метода Эйлера на i-м шаге интегрирования Δ_i может быть использована разность между точным решением уравнения $y(x_i)$ и соответствующим значением численного решения $y_i = y_{i-1} + \Delta x f(x_{i-1}, y(x_{i-1}))$:

$$\Delta_i = y(x_i) - \big[y_{i-1} + \Delta x f(x_{i-1}, y(x_{i-1}))\big]. \tag{2.9}$$

Вспоминая, что $x_i = x_{i-1} + \Delta x$, и раскладывая $y(x_{i-1}+h)$ в ряд Тейлора, приводим (2.9) к следующему виду:

$$\begin{aligned} \Delta_i = y_{i-1} + y'(x_{i-1})\Delta x + \frac{1}{2}y''(x_{i-1})(\Delta x)^2 - \ldots \\ -\big[y_{i-1} + \Delta x f(x_{i-1}, y(x_{i-1}))\big] + O(\Delta x^3), \end{aligned} \tag{2.10}$$

но в соответствии с уравнением (2.2) $y'(x_{i-1}) = f(x_{i-1}, y(x_{i-1}))$, поэтому

$$\Delta_i = \frac{1}{2}y''(x_i)(\Delta x)^2 + O(\Delta x^3). \tag{2.11}$$

Из (2.11) видно, что главный член погрешности интегрирования ДУ первого порядка по методу Эйлера на одном шаге пропорционален $(\Delta x)^2$. После N шагов погрешность составит равную $N \cdot (\Delta x)^2$. Так как при заданном интервале интегрирования $N \sim 1/\Delta x$, полная погрешность численного решения $\Delta \sim \Delta x$, поэтому говорят, что метод Эйлера является методом первого порядка точности.

В общем случае отклонение численного решения от точного решения обусловлено двумя причинами. Во-первых, компьютеры не оперируют с вещественными числами (например, десятичными числами с дробной частью) бесконечной точности, но представляют вещественные числа в виде конечного числа десятичных цифр, определяемого аппаратными средствами компьютера[1] и его программным обеспечением.

[1] На IBM совместимых компьютерах серий XT, AT точность представления числа определялась наличием дополнительной микросхемы (сопроцессора), в функции которой входило выполнение операций с действительными числами с повышенной точностью. Начиная с поколения Pentium, все компьютеры имеют процессор и сопроцессор, выполненные в виде одной микросхемы.

Это приводит к тому, что арифметические операции, выполняемые с действительными числами, будут выполняться с некоторой погрешностью, называемой погрешностью округления. Например, при использовании компьютера, оперирующего с вещественными числами, содержащими только одну значащую цифру после запятой, для умножения чисел $1,1 \cdot 2,7$ будет получен ответ $3,0$. Погрешности округлений по мере роста объема вычислений имеют свойство накапливаться.

Второй причиной отклонения численного решения от точного является вычислительный алгоритм, применяемый в конкретной задаче. Указанное обстоятельство определяет необходимость проводить специальные исследования применяемых вычислительных алгоритмов и оценивать их точностные характеристики. Подчеркнем, что не существует правил для выбора «наилучшего» метода решения ДУ. У каждого метода, как показывает практика, имеются свои достоинства и недостатки, а конкретный выбор определяется требованиями и квалификацией исследователя, а также характером конкретного решения, который заранее неизвестен. Вот некоторые наиболее типичные вопросы, присущие численным исследованиям. Насколько точным должен быть ответ? На каком интервале следует искать решения задачи? Каковы возможности вашего компьютера?

На практике точность численного решения определяют, уменьшая шаг интегрирования ДУ до тех пор, пока численное решение не перестанет зависеть от шага при заданном уровне точности. Выбирая величину шага, важно помнить, что выбор слишком малого шага приводит к увеличению объема вычислений и, соответственно, погрешности округлений.

Другой не менее важной характеристикой алгоритма является его устойчивость. В ряде задач возникают ситуации, когда численные результаты находятся в хорошем соответствии с «истинным решением» на коротких интервалах, а на больших интервалах отклоняются от него. Это обусловлено тем, что малые погрешности алгоритма, многократно перемножаясь, приводят к геометрическому росту погрешности. О таком алгоритме применительно к данной задаче говорят как о неустойчивом алгоритме. Очевидно, что использование алгоритма, оказывающегося неустойчивым в данной задаче, приведет к неверным численным результатам. Поэтому при решении конкретной задачи проводят специальные исследования, позволяющие оценить точность и устойчивость выбранного вычислительного алгоритма.

2.4. ОЦЕНКА КОЭФФИЦИЕНТА ОСТЫВАНИЯ ПО ЭКСПЕРИМЕНТАЛЬНЫМ РЕЗУЛЬТАТАМ

Прежде чем приступать к исследованию точности численного решения уравнения (2.1), получаемого методом Эйлера, найдем оценку «коэффициента остывания» r. Для этого воспользуемся экспериментальными результатами, представленными в таблице 2.1.

В таблице 2.1 сведены результаты измерений температуры чашки кофе в последовательные моменты времени (температура окружающего воздуха $22,0^{\circ}$C). Температура регистрировалась с точностью $0,1^{\circ}$C.

Таблица 2.1. Экспериментальные значения температуры кофе

Время, мин	T,°C	Время, мин	T,°C
0,0	83,0	8,0	64,7
1,0	77,7	9,0	63,4
2,0	75,1	10,0	62,1
3,0	73,0	11,0	61,0
4,0	71,1	12,0	59,9
5,0	69,4	13,0	58,7
6,0	67,8	14,0	57,8
7,0	66,4	15,0	56,6

В связи с тем, что задача обработки экспериментальных данных присутствует во всех лабораторных физических практикумах, мы считаем целесообразным на рассматриваемом примере продемонстрировать методику реализации метода Эйлера наименьших квадратов (МНК) в MATLAB.

Напомним постановку задачи МНК и общий подход к ее решению. Для некоторого набора данных, заданных таблично, требуется найти формулу, выражающую данную зависимость аналитически. Поставим задачу так, чтобы с самого начала обязательно учитывался характер исходной функции: найти функцию заданного вида

$$y = F(x), \tag{2.12}$$

которая в точках $x_1, x_2, \ldots, x_n$ принимает значения как можно более близкие к табличным значениям $y_1, y_2, \ldots, y_n$. Предположим, что приближающая функция $F(x)$ в точках $x_1, x_2, \ldots, x_n$ имеет значения

$$\overline{y}_1, \overline{y}_2, \ldots, \overline{y}_n. \tag{2.13}$$

Требование близости табличных значений $y_1, y_2, \ldots, y_n$ и значений (2.13) можно истолковать следующим образом. Будем рассматривать совокупность значений функции $f(x)$ и совокупность значений (2.13) как координаты двух точек n-мерного пространства. С учетом этого задача приближения функции может быть переформулирована следующим образом: найти такую функцию $F(x)$ заданного вида, чтобы расстояние между точками $M(y_1, y_2, \ldots, y_n)$ и $\overline{M}(\overline{y}_1, \overline{y}_2, \ldots, \overline{y}_n)$ было наименьшим. Воспользовавшись метрикой евклидова пространства, приходим к требованию, чтобы величина

$$\sqrt{(y_1 - \overline{y}_1)^2 + (y_2 - \overline{y}_2)^2 + \ldots + (y_n - \overline{y}_n)^2} \tag{2.14}$$

была наименьшей, что равносильно следующему: сумма квадратов

$$(y_1 - \overline{y}_1)^2 + (y_2 - \overline{y}_2)^2 + \ldots + (y_n - \overline{y}_n)^2. \tag{2.15}$$

Окончательно задача аппроксимации функции $f(x)$ теперь формулируется следующим образом: для функции $f(x)$, заданной таблицей 2.1, найти функцию $F(x)$

определенного вида так, чтобы сумма квадратов (2.14), (2.15) была наименьшей. Эта задача называется приближением функции методом наименьших квадратов.

Рассмотрим метод нахождения параметров приближающей функции в общем виде на примере приближающей функции, зависящей от трех параметров:

$$y = F(x, a, b, c). \tag{2.16}$$

Имеем

$$F(x_i, a, b, c) = \overline{y}_i, \quad i = 1, 2, \ldots, n. \tag{2.17}$$

Сумма квадратов разностей соответствующих значений функций $f(x)$ и $F(x)$ имеет вид

$$\sum_{i=1}^{n} (y_i - F(x_i, a, b, c))^2 = \Phi(a, b, c). \tag{2.18}$$

Из (2.18) видно, что сумма является функцией $\Phi(a, b, c)$ трех переменных. Используя необходимое условие экстремума

$$\frac{\partial \Phi}{\partial a} = 0, \quad \frac{\partial \Phi}{\partial b} = 0, \quad \frac{\partial \Phi}{\partial c} = 0,$$

получаем систему уравнений

$$\begin{aligned} &\sum_{i=1}^{n} \big[y_i - F(x_i, a, b, c)\big] \cdot F'_a(x_i, a, b, c) = 0, \\ &\sum_{i=1}^{n} \big[y_i - F(x_i, a, b, c)\big] \cdot F'_b(x_i, a, b, c) = 0, \\ &\sum_{i=1}^{n} \big[y_i - F(x_i, a, b, c)\big] \cdot F'_c(x_i, a, b, c) = 0. \end{aligned} \tag{2.19}$$

Решив систему (2.19) относительно параметров a, b, c, получаем конкретный вид функции $F(x, a, b, c)$. Изменение количества параметров не приведет к искажению сущности самого подхода, но выразится в изменении количества уравнений в системе (2.19).

Значения разностей

$$y_i - F(x_i, a, b, c) = \varepsilon_i \tag{2.20}$$

называют остатками или отклонениями измеренных значений от вычисленных по формуле (2.16). Сумма квадратов отклонений

$$\sigma = \sum_{i}^{n} \varepsilon_i^2 \tag{2.21}$$

в соответствии с принципом наименьших квадратов для заданного вида приближающей функции должна быть наименьшей.

Из двух разных приближений одной и той же табличной функции лучшим считается то, для которого (2.20) имеет наименьшее значение.

Для примера получим систему уравнений для определения неизвестных коэффициентов при использовании линейной аппроксимации

$$F(x, a, b) = ax + b. \tag{2.22}$$

Сумма квадратов разностей соответствующих значений функций $f(x)$ и $F(x,a,b)$ в рассматриваемом случае записывается в виде

$$\sum_{i=1}^{n}(y_i - F(x_i,a,b))^2 = \Phi(a,b). \tag{2.23}$$

Найдем частные производные (2.23)

$$\frac{\partial F}{\partial a} = x, \quad \frac{\partial F}{\partial b} = 1. \tag{2.24}$$

Составим систему вида (2.19)

$$\sum_{i=1}^{n}(y_i - ax_i - b)x_i = 0,$$
$$\sum_{i=1}^{n}(y_i - ax_i - b) = 0.$$

(Здесь и далее сумма ведется по переменной $i = 1, 2, \ldots, n$.)

Далее имеем

$$\begin{aligned} &\sum_{i=1}^{n} y_i x_i - a\sum_{i=1}^{n} {x_i}^2 - b\sum_{i=1}^{n} x_i = 0, \\ &\sum y_i - a\sum x_i - bn = 0. \end{aligned} \tag{2.25}$$

Разделив каждое уравнение (2.26) на n, получим

$$\begin{aligned} &\left(\frac{1}{n}\sum_{i=1}^{n} {x_i}^2\right)a + \left(\frac{1}{n}\sum_{i=1}^{n} x_i\right)b = \frac{1}{n}\sum_{i=1}^{n} y_i x_i, \\ &\left(\frac{1}{n}\sum_{i=1}^{n} x_i a\right) + b = \frac{1}{n}\sum_{i=1}^{n} y_i. \end{aligned} \tag{2.26}$$

Введем обозначения

$$\frac{1}{n}\sum x_i = M_x, \quad \frac{1}{n}\sum y_i = M_y,$$
$$\frac{1}{n}\sum x_i y_i = M_x y, \quad \frac{1}{n}\sum {x_i}^2 = M_{x^2},$$

используя которые можно записать систему (2.26) в следующем виде:

$$\begin{aligned} &M_{x^2}\cdot a + M_x \cdot b = M_{xy}, \\ &M_x \cdot a + b = M_y, \end{aligned} \tag{2.27}$$

или в матричной форме

$$\begin{pmatrix} M_{x^2} & M_x \\ M_x & 1 \end{pmatrix}\begin{pmatrix} a \\ b \end{pmatrix} = \begin{pmatrix} M_x y \\ M_y \end{pmatrix}.$$

Откуда

$$\begin{pmatrix} a \\ b \end{pmatrix} = \begin{pmatrix} M_{x^2} & M_x \\ M_x & 1 \end{pmatrix}^{-1} \begin{pmatrix} M_x y \\ M_y \end{pmatrix}. \tag{2.28}$$

Для аппроксимации экспериментальных данных, представленных в табл. 2.1, будем использовать экспоненциальную функцию вида

$$F(x, a, b) = \exp(b + at), \tag{2.29}$$

где a и b — неизвестные коэффициенты, значения которых нужно оценить по методу наименьших квадратов. Решение данной задачи может быть сведено к рассмотренной выше задаче отыскания коэффициентов линейной функции. Действительно, введем в рассмотрение функцию

$$\Phi(x, a, b) = \ln(F(x, a, b)) = a + b \cdot t, \tag{2.30}$$

которая оказывается линейной функцией относительно неизвестных коэффициентов a, b. Таким образом, для решения задачи о нахождении аппроксимирующей функции (2.29) можно использовать следующий алгоритм:

1. По известной таблице 2.1 построить таблицу, содержащую зависимость
$$\ln(T) = f(T).$$

2. Используя данные новой таблицы, найти значения неизвестных коэффициентов a, b линейной функции, решив систему уравнений (2.28).

3. Определить аппроксимирующую функцию, подставив найденные значения коэффициентов a, b в (2.29).

Описанный алгоритм реализуется в MATLAB следующей последовательностью команд, которую мы сохранили в файл `Glava2_2.m`:

```
>> x = 0:15; % задание моментов времени
             % из табл. 2.1
>> y = [83 77.7 75.1 73.0 71.1 69.4...
        67.8 66.4 64.7 63.4 62.1...
        61.0 59.9 58.7 57.8 56.6]; % задание значений
                                   % температуры из табл. 2.1
>> N = length(x);
>> y1 = log(y);
% вычисление элементов матрицы системы (2.27)
>>  Mx = sum(x)/N;
>>  My = sum(y1)/N;
>>  Mx2 = sum(x.^2)/N;
>>  Mxy = x*y1'/N;;
>>  M = [Mx2 Mx; Mx 1]; % задание матрицы системы (2.28)
>>  d = [Mxy My]'; % задание вектора-столбца
                   % свободных членов системы (2.28)
```

```
>>  s = M\d; % решение системы линейных уравнений (2.28)
>>  t = 0:0.01:x(N); % задание
                     % дискретной переменной
                     % для вычисления
                     % значений
                     % аппроксимирующей функции
>> T = exp(s(1)*t+s(2)); % вычисление значений
                         % аппроксимирующей функции
>> plot(x,y,'o',t,T,'MarkerSize',2) % визуализация
                                    % исходных данных и
                                    % аппроксимирующей
                                    % функции
```

Рис. 2.3. Исходные данные и аппроксимирующая их функция

Представляется целесообразным сделать небольшой комментарий к некоторым командам, использованным при решении задачи об аппроксимации исходных данных экспоненциальной функцией. При вычислении значения выражения $M_{xy} = \frac{1}{n}\sum x_i y_i$ требуется вычислить скалярное произведение двух векторов, которые в MATLAB по умолчанию записываются в виде строк, поэтому выполнение команды `x*y` приводит к появлению следующего сообщения об ошибке:

```
??? Error using ==> *
Inner matrix dimensions must agree.
(ошибка в использовании ==> *
размерности матриц должны быть согласованными)
```

Причина данной ошибки является следствием того, что в MATLAB, ориентированном на выполнение матричных операций, любой вектор также рассматривается как матрица. В рассматриваемой задаче `x`, `y` — векторы размерности 1×16. По правилу умножения матриц число столбцов первой матрицы должно равняться числу строк второй матрицы. Как очевидно, данное условие для векторов `x`, `y` не выполняется, поэтому операцию умножения следует предварить операцией транспонирования вектора `y` — `y1'`.

Ниже показаны значения переменных `y` и `y1'`

```
>> y
y =
  Columns 1 through 5
   83.0000   77.7000   75.1000   73.0000   71.1000
  Columns 6 through 10
   69.4000   67.8000   66.4000   64.7000   63.4000
  Columns 11 through 15
   62.1000   61.0000   59.9000   58.7000   57.8000
  Column 16
           56.6000
 >> y'
ans =
   83.0000
   77.7000
   75.1000
   73.0000
   71.1000
   69.4000
   67.8000
   66.4000
   64.7000
   63.4000
   62.1000
   61.0000
   59.9000
   58.7000
   57.8000
   56.6000
```

Для вывода данных, представленных в таблице 2.1 в виде кругов в команде `plot` после ввода имен векторов `x`, `y`, указан тип маркера точек `'o'`. Размер маркера точек задается в команде `plot` строкой `'MarkerSize'` и следующим за ней числом 2. Размер маркера точек задается после перечисления всех векторов, используемых для построения графиков.

Отметим, что можно использовать два других способа решения рассматриваемой задачи. Первый способ состоит в использовании встроенной в MATLAB функции `polyfit(x,y,n)`, возвращающей значения коэффициентов полинома n-й степени $p(x) = \sum\limits_{i=0}^{n} a_i x^{N-i}$ в порядке уменьшения степеней, который с наименьшей среднеквадратичной погрешностью аппроксимирует табличные данные, занесенные в векторы `x`, `y`. Данный метод реализуется следующей последовательностью команд, которую мы сохранили в файле `Glava2_3.m`:

```
>>  x = 0:15; % задание моментов времени
              % из табл. 2.1
```

```
>>  y = [83 77.7 75.1 73.0 71.1 69.4...
        67.8 66.4 64.7 63.4 62.1...
        61.0 59.9 58.7 57.8 56.6]; % задание значений
                                   % температуры
>>  N = length(x);
>>  x1 = x';
>>  y1 = log(y)';
>>  s = polyfit(x1,y1,1); % вычисление коэффициентов полинома
>>  t = 0:0.01:x(N); % задание дискретной переменной
                     % для вычисления значений
                     % аппроксимирующей функции
>>  T = exp(polyval(s,t)); % вычисление значений
                           % аппроксимирующей функции
>>  plot(x,y,'o',t,T,'MarkerSize',2) % визуализация
                                     % исходных данных и
                                     % аппроксимирующей функции
```

Второй способ состоит в использовании функции `lsqcurvefit`, решающей задачу нелинейного метода наименьших квадратов, записанную в форме (2.23)

$$\min\left(\sum_{i=1}^{n}\left[y_i - e^{a\cdot x_i+b}\right]^2\right). \tag{2.31}$$

Данная функция входит в пакет `Optimization`, который расширяет возможности MATLAB в области решения задач оптимизации, систем нелинейных уравнений и задач аппроксимации. Для решения задачи сначала создадим m-файл, содержащий определение минимизируемой функции:

```
function Z = MNK(s,xdata)
   % функция, возвращающая значения выражения (2.31)
   N=length(xdata);
   k=1:N;
   Z=exp(s(1)*xdata(k)+s(2));
```

который сохраним на жестком диске в каталоге, предлагаемом MATLAB, по умолчанию под именем `MNK.m`. Далее необходимо выполнить в командном окне следующую последовательность команд, сохраненную нами в файле `Glava2_4.m`:

```
>> xdata = 0:15; % задание моментов времени из табл. 2.1
>> ydata = [83 77.7 75.1 73.0 71.1...
            69.4 67.8 66.4 64.7 63.4...
            62.1 61.0 59.9 58.7 57.8 56.6]; % задание
                                            % значений
                                            % температуры
                                            % из табл. 2.1
>> s0=[-0.1 3]; % задание начального приближения
>> [s,resnorm] = lsqcurvefit('MNK',s0,xdata,ydata)
```

После выполнения приведенной выше последовательности команд в рабочем окне будет выведена следующая информация:

```
Optimization terminated successfully:
Relative function value changing by less than OPTIONS.TolFun
s =
   -0.0241    4.3749
resnorm =
   21.8311
(Оптимизация успешно завершена
Относительное отклонение функции меньше, чем OPTIONS.TolFun)
```

Здесь переменная `s` содержит значения искомых параметров функции (`a` и `b` соответственно), `resnorm` — сумму квадратов отклонений, вычисляемых в соответствии с (2.21). Отметим, что существует возможность настроить пакет `Optimization`, установив значения соответствующих опций пакета. Общий список опций пакета можно получить, выполнив команду

```
>> optimset
    DerivativeCheck: [ on | {off} ]
    Diagnostics: [ on | {off} ]
    DiffMaxChange: [ positive scalar {1e-1} ]
    DiffMinChange: [ positive scalar {1e-8} ]
    Display: [ off | iter | notify | final ]
    GoalsExactAchieve: [ positive scalar | {0} ]
    GradConstr: [ on | {off} ]
    GradObj: [ on | {off} ]
    Hessian: [ on | {off} ]
    HessMult: [ function | {[]} ]
    HessPattern: [ sparse matrix |
            {sparse(ones(NumberOfVariables))} ]
    HessUpdate: [ dfp | gillmurray |
                    steepdesc | {bfgs} ]
    Jacobian: [ on | {off} ]
    JacobMult: [ function | ([]) ]
    JacobPattern: [ sparse matrix |
                  {sparse(ones(Jrows,Jcols))} ]
    LargeScale: [ {on} | off ]
    LevenbergMarquardt: [ on | off ]
    LineSearchType: [ cubicpoly | {quadcubic} ]
    MaxFunEvals: [ positive scalar ]
    MaxIter: [ positive scalar ]
    MaxPCGIter:[positive scalar |
              {max(1,floor(numberOfVariables/2))}]
    MeritFunction: [ singleobj | multiobj ]
    MinAbsMax: [ positive scalar | {0} ]
```

```
    PrecondBandWidth: [ positive scalar | {0} | Inf ]
    TolCon: [ positive scalar ]
    TolFun: [ positive scalar ]
    TolPCG: [ positive scalar | {0.1} ]
    TolX: [ positive scalar ]
    TypicalX: [ vector | {ones(NumberOfVariables,1)} ]
```

Подробное описание каждой опции приведено в документации, поставляемой вместе с пакетом MATLAB. Например, для того чтобы включить вывод в командное окно значения функции, минимизирующей (2.31), на каждом шаге итерации следует выполнить следующую команду:

```
>> [s,resnorm] = lsqcurvefit('MNK',s0,...
                            xdata,ydata,'','',...
                            optimset('display','iter'))
                 Norm of      First-order
 Iteration  Func-count     f(x)          step
     1          4           50659             1
     2          7           50659       2.74811
     3         10           50659      0.687029
     4         13         20192.1      0.171757
     5         16         12355.5      0.343514
     6         19         1885.55      0.687029
     7         22         50.9973       0.34649
     8         25         21.8404     0.0314554
     9         28         21.8311   0.000625592
    10         31         21.8311   5.9392e-006
         optimality   CG-iterations
          5.09e+004            0
          5.09e+004            1
          5.09e+004            0
          8.05e+004            0
          2.14e+004            1
          5.35e+003            1
          4.55e+003            1
                 14            1
                1.3            1
                0.0179         1



Optimization terminated successfully:
Relative function value changing by less than OPTIONS.TolFun
s =
   -0.0241    4.3749
resnorm =
   21.8311
```

Здесь использована функция `optimset('display','iter')`, позволяющая установить значение опции `display` равным `iter`. Для отмены вывода в командное окно какой-либо информации о процессе вычислений следует набрать команду

```
>> [s,resnorm] = lsqcurvefit('MNK',s0,...
                             xdata,ydata,'','',...
                             optimset('display','off'))
s =
   -0.0241    4.3749
resnorm =
   21.8311
```

***Задача* 2.1**

Постройте на одном чертеже аппроксимирующие функции, коэффициенты которых вычислены каждым из трех выше перечисленных методов. Сравните сумму квадратов отклонений между табличными значениями и значениями аппроксимирующих функций, полученные при использовании функций `polyfit` *и* `lsqcurvefit`.

***Задача* 2.2**[2]

Для изучения точности метода Эйлера можно воспользоваться аналитическим решением ДУ (2.1)*, которое записывается в виде*

$$T(t) = T_S - (T_S - T_0) \cdot e^{-r\,t}, \tag{2.32}$$

где T_S — температура окружающей среды, T_0 — начальная температура тела[3].

1. *Выберите значение коэффициента r, соответствующее реальному процессу, и для того чтобы находить численное решение уравнения* (2.32)*, внесите необходимые изменения в файл* `F.m`. *Задайте временной интервал, на котором будет находиться численное решение, шаг интегрирования Δt, температуру окружающей среды и начальную температуру жидкости в чашке. Постройте на одном чертеже точное и численное решения.*
2. *Вычислите разность между численным и точным решениями для следующих значений шага интегрирования $\Delta t = 0,1, 0,05, 0,025, 0,01$ и $0,005$. Зависит ли разность между численным и точным решениями от времени? Нарисуйте график разности как функцию Δt. Будет ли эта разность убывающей функцией Δt? Если разность между точным и аналитическим решениями при заданном значении t пропорциональна $(\Delta t)^n$, численный метод называется методом n-го порядка точности. Какой порядок точности у метода Эйлера?*
3. *Какой шаг необходимо выбрать, чтобы точность $0,1\%$ достигалась в момент времени $t = 1$? Какой необходимо выбрать величину шага Δt, чтобы точность $0,1\%$ достигалась в момент времени $t = 5$?*

[2] Задачи 2.1, 2.2, 2.3 взяты автором из [2].

[3] Заметим, что $T(t = 0) = T_S - (T_S - T_0) = T_0$, а $T(t \to \infty) = T_S$.

***Задача* 2.3**

Процесс зарядки изначально незаряженного электрического конденсатора в RC-цепи (рис. 2.4), к которому в момент времени $t = 0$ приложено напряжение U, описывается ДУ первого порядка

$$R\frac{dQ}{dt} = V - \frac{Q}{C} \qquad (2.33)$$

с начальным условием $Q(0) = 0$.

Рис. 2.4. RC-цепочка

1. *Найдите численное решение уравнения* (2.33), *задав следующие характеристики цепи $R = 2000$ Ом, $C = 10^{-6}$ Ф, $V = 10$ В. Как зависит заряд на конденсаторе от времени? Увеличивается ли заряд до бесконечного значения или происходит его насыщение?*
2. *Один из методов определения точности численного решения заключается в повторении вычислений с меньшим шагом и сравнении полученных результатов. Если в обоих результатах совпадают n десятичных цифр, то можно предположить, что совпадают и следующие десятичные знаки. Какой необходимо выбрать шаг, чтобы получить решение с тремя правильными десятичными знаками в момент времени $t = 0,005$ с?*
3. *Каковы особенности численного решения уравнения* (2.9) *и значений шагов $\Delta t = 0,005, 0,004, 0,003$? Приводит ли малое изменение шага к большому изменению вычисляемой величины Q? Устойчив ли метод Эйлера для любой величины шага?*

***Задача* 2.4**

Предположим, что начальная температура кофе $90°C$, однако начинать пить кофе можно, когда температура опустится ниже $60°C$. Предположим, что вы можете добавить в кофе молоко, которое уменьшает температуру кофе на $5°C$. Для того чтобы охладить кофе до нужной температуры вы можете поступить двумя способами: 1) добавить молоко при температуре кофе $90°C$; 2) добавить молоко при температуре кофе $75°C$. В каком случае охлаждение кофе до $60°C$ произойдет быстрее? (Ответ необходимо получить, выполнив численные расчеты.)

В заключение необходимо отметить, что приведенное в данном разделе решение задачи об аппроксимации экспериментальной зависимости температуры от времени, вообще говоря, не является полным, поскольку за его рамками осталось обсуждение ряда сопутствующих проблем (нахождение доверительных интервалов коэффициентов регрессии, оценка значимости коэффициентов регрессии, вычисление доверительных границ построенной модели, анализ остатков регрессионной модели и др.). Это сделано сознательно, поскольку они требуют детального обсуждения, что выходит за рамки нашей книги. Для читателей, желающих более подробно изучить теорию методов аппроксимации экспериментальных зависимостей и познакомиться с примерами решений практических задач, мы рекомендуем обратиться к соответствующим научным монографиям [7, 8], справочникам [9, 10] и учебно-методическим пособиям [11].

2.5. РЕШЕНИЕ ДИФФЕРЕНЦИАЛЬНЫХ УРАВНЕНИЙ МЕТОДОМ РУНГЕ–КУТТА ЧЕТВЕРТОГО ПОРЯДКА

Изучив разделы 2.1–2.3 и решив задачи 2.1–2.2, вы могли убедиться в известных ограничениях метода Эйлера по точности, а также в возникновении в ряде случаев неустойчивых решений. Эти недостатки определили необходимость разработки новых вычислительных алгоритмов. Их подробное описание заинтересованный читатель может найти в списке литературы [2–6]. Несмотря на то что не следует отдавать предпочтение одному какому-нибудь алгоритму, для наших дальнейших целей оказывается достаточным ограничиться одним вычислительным алгоритмом, который позволяет получать устойчивые решения, отвечающие требованиям по точности, во всех рассмотренных нами задачах. Указанными свойствами обладает метод Рунге–Кутта четвертого порядка. Данный метод реализуется следующей итерационной формулой [3]:

$$y_{n+1} = y_n + \frac{\Delta x}{6}(k_1(y_n) + 2k_2(y_n) + 2k_3(y_n) + k_4(y_n)), \tag{2.34}$$

где k_1, k_2, k_3, k_4 — поправки, вычисляемые по формулам

$$\begin{aligned} k_1 &= f(x_n, y_n), \\ k_2 &= f(x_n + \Delta x/2, y_n + \Delta x k_1/2), \\ k_3 &= f(x_n + \Delta x/2, y_n + \Delta x k_2/2), \\ k_4 &= f(x_n + \Delta x, y_n + \Delta y). \end{aligned} \tag{2.35}$$

Условие сходимости данного метода записывается в виде [3]

$$\Delta x \lambda \leq 2.78, \tag{2.36}$$

где λ — коэффициент, входящий в модельное уравнение

$$\frac{dy}{dx} + \lambda y = 0, \quad \lambda = \text{const} > 0, \quad x > 0, \quad y(0) = y_0. \tag{2.37}$$

Метод Рунге–Кутта 4-го порядка решения задачи Коши для ДУ в MATLAB реализован в виде функции `ode45`. Данный метод рекомендуется использовать при первой попытке нахождения численного решения задачи. Помимо данной функции в MATLAB реализованы и другие методы решения дифференциальных уравнений и систем ДУ:

- `ode23` — функция реализует одношаговые явные методы Рунге–Кутта (2-го и 3-го) порядков. Функция используется при решении нежестких систем ДУ, обеспечивает удовлетворительную точность при меньших, нежели функция `ode45`, временных затратах.

- `ode113` — функция реализует многошаговый метод Адамса–Башворта–Мултона переменного порядка. Функция используется при необходимости обеспечить высокую точность численного решения.

- `ode15s` — функция реализует многошаговый метод переменного порядка (от 1-го до 5-го по умолчанию), основанный на формулах численного дифференцирования. Данный метод следует использовать в том случае, если не удается найти численное решение с помощью функции `ode45`.

- `ode23s` — функция реализует одношаговый метод, использующий модифицированную формулу Розенброка 2-го порядка. Данный метод по сравнению с другими методами обеспечивает более высокую скорость вычислений при относительно более низкой точности вычислений.

- `ode23t` — функция реализует метод трапеций с интерполяцией. Данный метод используют при решении уравнений, описывающих колебательные системы с почти гармоническим выходным сигналом.

- `ode23tb` — функция реализует неявный метод Рунге–Кутта в начале интервала интегрирования и далее метод, использующий формулы обратного дифференцирования 2-го порядка. Данный метод обладает большей скоростью, нежели метод `ode15s`, соответственно, при меньшей точности.

Все перечисленные выше функции, описанные в документации пакета Solver (решатель), могут решать системы ДУ явного вида $\vec{y}' = \vec{F}(t, \vec{y})$. Кроме того, решатели `ode15s`, `ode23s`, `ode23t` и `ode23tb` системы дифференциальных уравнений неявного вида $M(t, \vec{y})\vec{y}' = \vec{F}(t, \vec{y})$, а также все решатели, кроме `ode23s`, находят решения уравнений вида $M(\vec{y})\vec{y}' = \vec{F}(t, \vec{y})$.

Для решения дифференциального уравнения (2.1) сначала создаем m-файл, содержащий определение функции, стоящей в правой части уравнения (2.1):

```
function Z=Tempr(t,T)
  % определение функции, стоящей в правой части уравнения (2.1)
  global Ts r
  Z(1)=-r*(T-Ts);
```

который мы сохранили на диске под именем `Tempr.m`.

Далее необходимо выполнить в командном окне следующую последовательность операторов, сохраненную нами в файл `Glava2_5.m`:

```
>> global Ts r % объявление глобальных переменных
>> Ts = 22; % задание значения температуры
            % окружающей среды
>> r = 0.024; % задание значения
              % коэффициента остывания
>> T0 = 80; % задание начальной температуры тела
>> [t,T]=ode45('Tempr',[0:0.01:15],T0);
               % Tempr - имя файла,
               % содержащего определение
               % функции, стоящей в правой части
               % уравнения (2.1);
               % [0:0.01:15] - вектор,
               % определяющий интервал
               % интегрирования,
               % T0 - вектор начальных условий
>> plot(t,T) % визуализация решения ДУ (2.1)
```

После выполнения приведенной выше последовательности команд будет создано окно, содержащее график зависимости температуры тела от времени, представленный на рис. 2.5.

Рис. 2.5. Численное решение уравнения теплопроводности, возвращенное функцией `ode45`

По умолчанию решатели систем дифференциальных уравнений пакета MATLAB используют параметры, относительная погрешность которых не превосходит переменной `RelTol` $= 10^{-3}$, граница абсолютной погрешности численного решения — переменная `AbsTol` равна 10^{-6}. Для изменения значений этих переменных используется команда

```
>> options = odeset('RelTol',1e-4,'AbsTol',1e-4);
```

предваряющая команду вызова функции решателя системы ДУ.

***Задача* 2.5**

Решите задачи 2.1, 2.2, используя вместо функции, реализующей метод Эйлера, функции `ode45`, `ode23`, `ode15s`, `ode23s`, `ode23t`, `ode23tb`. *Проведите сравнительную оценку точности данных методов, сравнив численные и точные решения.*

2.6. МОДЕЛИРОВАНИЕ РАДИОАКТИВНОГО РАСПАДА

Ядерный распад зависит только от внутренних свойств данного типа атомного ядра. Особенность атомных ядер состоит в том, что они могут распадаться. Вероятность распада описывается выражением

$$W = \frac{1}{N}\frac{dN}{dt}, \tag{2.38}$$

где $N(t)$ — количество ядер в образце в момент времени t, dN — количество ядер в образце в момент времени dt,

$$dN = N(t) - N(t + dt). \tag{2.39}$$

Тогда скорость распада определяется как

$$\frac{dN}{dt} = -W \cdot N. \tag{2.40}$$

Уравнение (2.38) называется уравнением распада. Оно является ДУ первого порядка с разделяющимися переменными, которое может быть легко проинтегрировано. Его решение имеет следующий вид

$$N(t) = N_0 e^{-Wt}, \tag{2.41}$$

где N_0 — число ядер в момент времени $t = 0$.

***Задача* 2.6**

Найдите численное решение уравнения (2.16)*, используя функции* `ode45`*,* `ode23`*,* `ode15s`*,* `ode23s`*,* `ode23t`*,* `ode23tb`*. Сравните точные численные решения уравнения* (2.17)*, полученные разными методами. Как меняется точность численного решения при изменении шага интегрирования?*

2.7. МОДЕЛИРОВАНИЕ ЦЕПНОЙ РЕАКЦИИ ЯДЕРНОГО ВЗРЫВА

В 1938 г. было открыто деление атомного ядра при попадании в него нейтрона. При этом оказалось, что ядро распадается на две части с одновременным вылетом 2–3 нейтронов. Осколки ядра под действием кулоновских сил приобретают кинетическую энергию, которая затем переходит в тепловую энергию кристаллической решетки. В свою очередь образовавшиеся нейтроны могут попасть в другие ядра, что опять приведет к делению ядра с появлением дополнительных нейтронов, и т. д. Данное явление получило название цепной реакции. В том случае, если число нейтронов на каждом последующем шаге больше числа нейтронов на предыдущем шаге, происходит рост числа осколков в геометрической прогрессии. В результате происходит ядерный взрыв.

Для описания цепной реакции используется следующая математическая модель. Будем рассматривать ядро U^{235} как шарик радиуса R. Нейтрон, двигаясь со скоростью v, за время dt захватит цилиндрический объем V:

$$V = \pi \cdot R^2 \cdot v \cdot dt. \tag{2.42}$$

Нейтрон вызовет деление ядер, находящихся в данном объеме. Количество делений, вызванных одним нейтроном, за время dt:

$$N \cdot V \cdot \alpha = N \cdot \pi \cdot R^2 \cdot v \cdot \alpha \cdot dt, \tag{2.43}$$

где α — вероятность распада ядер, N — концентрация ядер. Удобно преобразовать (2.43), введя величину $\sigma_r = \alpha \cdot \pi \cdot R^2$ (называемую сечением деления), характеризующую способность ядер данного вещества к делению под действием нейтронов:

$$N \cdot V \cdot \alpha = N \cdot \sigma_r \cdot v \cdot dt. \tag{2.44}$$

Будем предполагать, что при единичном акте деления образуется m нейтронов и один нейтрон поглощается. Тогда изменение числа нейтронов за время dt определяется выражением

$$dn = n \cdot N \cdot \sigma_r \cdot v \cdot dt \cdot (\mu - 1). \tag{2.45}$$

Уравнение (2.45) является ДУ первого порядка. Решением уравнения (2.45) является функция

$$n(t) = n_0 \cdot \exp(N \cdot \sigma_r \cdot v \cdot t \cdot (\mu - 1)). \tag{2.46}$$

Оценим порядок величины $n(t)$ для образца со следующими параметрами: плотность образца $\rho = 18\,г/см^3$, $\mu = 2,5$, $\sigma_r = 1,6 \cdot 10^{-24}\,см^2$, $n = 2 \cdot 10^9\ см/с$. Так как для выбранной плотности вещества концентрация ядер

$$N = \frac{\rho}{\mu_{U^{235}}} N_A \approx 4,6 \cdot 10^{22}\,см^{-3}, \tag{2.47}$$

то за время 10^{-6} $с$ начальное количество нейтронов увеличивается в 10^{26} раз, то есть происходит ядерный взрыв.

При построении модели мы предполагали, что все образовавшиеся нейтроны участвуют в делении, поэтому даже один свободный нейтрон будет вызывать ядерный взрыв. Однако этот вывод не соответствует опытным данным, которые показывают, что ядерный взрыв может происходить только при определенных условиях. Данное обстоятельство свидетельствует о необходимости уточнения модели ядерного взрыва. Это уточнение состоит в учете того обстоятельства, что часть образовавшихся нейтронов может покидать образец, не участвуя в дальнейшем делении ядер.

Окончательно модель, учитывающая рождение и вылет нейтронов, описывается следующим уравнением:

$$\frac{dn}{dt} = (a - b) \cdot n, \tag{2.48}$$

здесь

$$a = N \cdot \sigma_r \cdot v \cdot (\mu - 1), \tag{2.49}$$

$$b = \frac{k \cdot v}{R_{U^{235}}}, \tag{2.50}$$

где k — дополнительный множитель, учитывающий реальное уменьшение плотности нейтронов, $R_{U^{235}}$ — характерный размер образца урана[4].

Из (2.50) видно, что существует линейный размер образца, называемый критическим радиусом R, для которого $a - b = 0$:

$$R = \frac{k}{N \cdot \sigma_r \cdot (\mu - 1)}. \tag{2.51}$$

[4] Вид зависимости коэффициента b от линейных размеров образца очевиден из следующих соображений. Количество нейтронов, выходящих из образца, прямо пропорционально скорости их движения и площади поверхности образца и обратно пропорционально количеству ядер, содержащихся в данном образце. Так как площадь поверхности пропорциональна квадрату линейного размера образца, а количество ядер — его объему, то есть кубу линейного размера, то коэффициент b будет обратно пропорционален линейному размеру.

Масса образца, соответствующая критическому радиусу (критическая масса), определяется как

$$M = \frac{4}{3}\pi R^3 \cdot \rho. \tag{2.52}$$

Для реального U^{235} $M = 50$ *кг* при критическом радиусе $8,5$ *см*.

***Задача* 2.7**

Считая образец, изготовленный из U^{235}, сферическим, исследуйте поведение решений уравнения (2.48) *при различных значениях массы образца. (Указание: изобразите на одном чертеже зависимости $n(t)$, полученные при различных значениях массы образца.)*

ЛИТЕРАТУРА К ГЛАВЕ 2

1. *Самойленко А. М., Кривошея С. А., Перстюк Н. А.* Дифференциальные уравнения: примеры и задачи. М.: Высшая школа, 1989.
2. *Гулд Х., Тобочник Я.* Компьютерное моделирование в физике. М.: Мир, 1990.
3. *Самарский А. А.* Введение в численные методы. М.: Наука, 1987.
4. *Бахвалов Н. С.* Численные методы. М.: Наука, 1975.
5. *Березин И. С., Жидков Н. П.* Методы вычислений. Ч. 1. М.: Наука, 1966.
6. *Марчук Г. И.* Методы вычислительной математики. М.: Наука, 1980.
7. *Дрейпер Н., Смит Г.* Прикладной регрессионный анализ. Множественная регрессия. М.: Диалектика; Вильямс, 2006.
8. *Поршнев С. В., Овечкина Е. В., Каплан В. Е.* Теория и алгоритмы аппроксимации эмпирических зависимостей и распределений. Екатеринбург: УрО РАН, 2006.
9. Справочник по прикладной статистике. В 2 т. Т. 1. / Под ред. Э. Ллойда, У. Ледермана, Ю. Н. Тюрина. М.: Финансы и статистика, 1989.
10. Справочник по прикладной статистике. В 2 т. Т. 2. / Под ред. Э. Ллойда, У. Ледермана, Ю. Н. Тюрина. М.: Финансы и статистика, 1989.
11. Компьютерный анализ и интерпретация эмпирических зависимостей / С. В. Поршнев и др. // М.: Бином, 2009.

Глава 3

ДИНАМИКА МАТЕРИАЛЬНОЙ ТОЧКИ

3.1. ОБЩАЯ ПОСТАНОВКА ЗАДАЧИ МАТЕМАТИЧЕСКОГО ОПИСАНИЯ ДИНАМИКИ МАТЕРИАЛЬНОЙ ТОЧКИ

Большинство задач, связанных с определением закона движения тела вблизи земной поверхности, имеют известные аналитические решения. Несмотря на это, мы считаем полезным вновь вернуться к этим задачам и получить их решения с помощью ПК, поскольку здесь можно продемонстрировать наиболее общие приемы и подходы к решению физических задач на ПК.

Отметим, что при решении данных задач используется простейшая модель, в которой не учитываются возможные вращения и внутренние движения тела, то есть реальное физическое тело заменяется идеализированным объектом, не обладающим внутренней структурой, — материальной точкой. Задача описания закона движения материальной точки традиционно рассматривается в кинематике, где движение материальной точки рассматривается безотносительно причины ее вызывающей[1], и динамике, где рассматриваются силы, действующие на тело. В соответствии со вторым законом Ньютона ускорение $\vec{a}$, с которым движется тело, прямо пропорционально действующей на него силе $\vec{F}$ (а если сил, приложенных к материальной точке, несколько, то их равнодействующей, то есть векторной сумме сил) и обратно пропорционально его массе m:

$$\vec{a} = \frac{\vec{F}}{m}. \tag{3.1}$$

В общем случае сила $\vec{F}(t)$ зависит от положения точки, характеризуемого радиусом-вектором $\vec{F}(t)$, скорости $\vec{v}(t)$ и времени

$$\vec{F}(t) = \vec{F}(t, \vec{r}, \vec{v}).$$

Тот факт, что силы, действующие на тело, не зависят от производных скорости, является свойством природы, но не следствием математического описания. Это свойство позволяет найти простые объяснения для многих наблюдаемых на практике движений.

Так как по определению

$$\vec{a} = \frac{d\vec{v}}{dt} = \frac{d^2\vec{r}}{dt^2},$$

[1] Данный подход уже был использован нами в главе 1.

то зависимость скорости от времени описывается уравнением

$$\frac{d\vec{v}}{dt} = \frac{\vec{F}}{m}, \tag{3.2}$$

а зависимость координаты от времени —

$$\frac{d^2\vec{r}}{dt^2} = \frac{\vec{F}}{m}, \tag{3.3}$$

поэтому для описания движения материальной точки необходимо решить ДУ второго порядка относительно координаты (3.3), которое для тела с постоянной массой эквивалентно системе ДУ первого порядка

$$\vec{v}(t) = \frac{d}{dt}\vec{r}, \quad \vec{a}(t) = \frac{d}{dt}\vec{v}. \tag{3.4}$$

Для описания движения тела с переменной массой $m(t)$ необходимо решать уравнение движения, в которых масса является функцией времени:

$$\frac{d\vec{p}}{dt} = \frac{d}{dt}[m(t)\vec{v}(t)] = \frac{dm(t)}{dt}\vec{v}(t) + m(t)\frac{d\vec{v}(t)}{dt} = \vec{F}(t),$$

называемое уравнением Мещерского.

3.2. ДВИЖЕНИЕ ТЕЛ В ГРАВИТАЦИОННОМ ПОЛЕ БЕЗ УЧЕТА ТРЕНИЯ

Рассмотрим задачу об одномерном движении материальной точки с постоянной массой в гравитационном поле Земли. В соответствии с законом всемирного тяготения сила, действующая на тело массы m, равна

$$F = \frac{\gamma M m}{(R+y)^2} = \frac{gm}{(1+y/R)^2}, \tag{3.5}$$

где y — расстояние от поверхности Земли, R — радиус Земли, g — гравитационная постоянная, M — масса Земли, $g = \gamma M/R^2$. При этом на первом шаге будем считать силу тяжести постоянной, что справедливо, как видно из (3.5), для случая, когда $y/R \ll 1$.

Выберем систему отсчета с положительным направлением координатной оси вверх (см. рис. 3.1). Подставляя (3.5) в (3.3), получаем уравнение движения в выбранной системе отсчета

$$\frac{d^2y}{dt^2} = -g. \tag{3.6}$$

Данное уравнение является ДУ второго порядка, для которого задача Коши ставится заданием начальных условий: значений координаты $y(0) = y_0$ и скорости $v(0) = y'(0) = v_0$. Аналитический ответ может быть получен последовательным интегрированием (3.6):

$$v(t) = v_0 - g \cdot t, \tag{3.7}$$

Рис. 3.1. К решению задачи о движении тела в гравитационном поле Земли

$$y(t) = y_0 + v_0 \cdot t + \frac{1}{2} g \cdot t^2. \tag{3.8}$$

Рассмотрим последовательность действий, реализующих решение этой задачи в MATLAB с начальными условиями $y(0) = 10$, $y'(0) = 1$.

Поскольку все описанные выше решатели работают с системами ОДУ, приведенными к форме Коши, необходимо привести системе ОДУ второго порядка к системе обыкновенных дифференциальных уравнений первого порядка, введя новую переменную

$$\frac{dy}{dt} = z. \tag{3.9}$$

Тогда уравнение (3.6) принимает следующий вид:

$$\frac{dz}{dt} = -g. \tag{3.10}$$

Уравнения (3.9), (3.10), образующие систему ОДУ первого порядка, как очевидно, можно записать в матричном виде

$$\frac{d}{dt}\begin{pmatrix} y \\ z \end{pmatrix} = \begin{pmatrix} z \\ -g \end{pmatrix}. \tag{3.11}$$

Запись системы в виде (3.11) позволяет создать m-файл, содержащий описание вектора-функции, стоящей в правой части (3.11):

```
function dy=Angle(t,y)
  global g  % объявление глобальной переменной
  dy=zeros(2,1); % задание вектора-столбца размерности 2x1
  % задание координат вектора-функции,
  % стоящей в правой части (3.11)
  dy(1)=y(2);
  dy(2)=-g;
```

и сохранить его на диске под именем `Angle.m`.

Далее необходимо выполнить в командном окне следующую последовательность команд, сохраненную нами в файле `Glava3_1.m`:

```
% листинг файла Glava3_1.m
global g % объявление глобальной переменной
g=9.8; % задание значения
       % ускорения свободного падения
% задание начальных условий
y0=10;
v0=1;
% решение системы ОДУ (3.11)
[t,y]=ode45('Angle',[0:0.01:1.5],[y0 v0]);
figure(1); plot(t,y(:,1)); % построение зависимости x=x(t)
                           % в графическом окне No 1
figure(2); plot(t,y(:,2)); % построение зависимости y=y'(t)
                           % в графическом окне No 2
```

После выполнения приведенной выше последовательности команд на экране монитора появляются два независимых стандартных графических окна. В окне с названием **Figure № 1** отображается зависимость координаты от времени (рис. 3.2), в окне с названием **Figure № 2** отображается зависимость скорости от времени (рис. 3.3).

Рис. 3.2. Зависимость координаты материальной точки от времени

Рис. 3.3. Зависимость скорости движения материальной точки от времени

***Задача* 3.1**

Проведите анализ полученных численных решений системы ДУ, описывающей движение материальной точки в поле тяжести Земли:

1. *Сравните зависимости скорости и координаты от времени, задаваемые выражениями* (3.7), (3.8), *с аналогичными зависимостями, полученными численным решением уравнения* (3.6). *Какова точность численного решения?*

2. *Постройте зависимость скорости от координаты.*

3. *Постройте зависимость пройденного пути материальной точки от времени.*

4. *Постройте зависимости кинетической и потенциальной энергий от времени.*

5. *Одним из фундаментальных законов физики является закон сохранения энергии, в соответствии с которым полная энергия материальной точки, движущейся в гравитационном поле Земли, должна оставаться постоянной. Проверьте полеченное численное решение ДУ на соответствие этому закону. С какой точностью выполняется закон сохранения энергии и чем можно объяснить обнаруженные расхождения?*

***Задача* 3.2**

Точное уравнение движения, полученное подстановкой выражения для силы тяжести, действующей на тело в гравитационном поле Земли, (3.5) *в выражение* (3.3)*, имеет вид:*

$$\frac{d^2y}{dt^2} = -\frac{g}{\left(1+\dfrac{y}{R}\right)^2} \ . \tag{3.12}$$

1. *Постройте зависимость ускорения свободного падения от высоты подъема материальной точки над поверхностью Земли.*

2. *Получите численное решение уравнения* (3.12) *и сравните его с соответствующим решением уравнения* (3.6) *для выбранных ранее начальных условий. Сколь велико отличие этих решений?*

3. *Сравните решения уравнений* (3.6) *и* (3.7) *при различных значениях начальной высоты. Начиная с каких высот гравитационное поле Земли нельзя считать однородным?*

4. *Постройте зависимости кинетической и потенциальной энергий от времени.*

5. *Проверьте полеченное численное решение ДУ на соответствие закону сохранения энергии. С какой точностью выполняется закон сохранения энергии?*

***Задача* 3.3**

Шарик, имеющий небольшой радиус (по сравнению с чем?), падает с высоты h на упруго отражающую поверхность.

1. *Разработайте программу, позволяющую исследовать движение шарика, падающего с заданной высоты на отражающую поверхность, предполагая, что удар является абсолютно упругим.*

2. *Постройте зависимости координаты, скорости, перемещения от времени и скорости от координаты.*

3. *Исследуйте движение шарика при неабсолютно упругом ударе.*

Рассмотрим двумерное движение материальной точки в гравитационном поле Земли (задача о движении тела, брошенного под углом к горизонту), описываемое уравнением (3.3). В системе отсчета, показанной на рис. 3.4, уравнение (3.3) эквивалентно следующей системе ОДУ второго порядка:

$$\frac{d^2x}{dt^2} = 0, \quad \frac{d^2y}{dt^2} = -g. \tag{3.13}$$

Система уравнений (3.13) заменой переменных

$$\frac{dx}{dt} = z_1, \qquad \frac{dy}{dt} = z_2$$

сводится к эквивалентной системе линейных уравнений первого порядка:

$$\begin{aligned} \frac{dx}{dt} &= z_1, \\ \frac{dz_1}{dt} &= 0, \\ \frac{dy}{dt} &= z_2, \\ \frac{dz_2}{dt} &= -g. \end{aligned} \tag{3.14}$$

Рис. 3.4. К решению задачи о движении тела, брошенного под углом к горизонту

Из (3.14) видно, что для решения задачи Коши данной системы следует задать четыре начальных условия $x(0)$, $x'(0)$, $y(0)$, $y'(0)$, а также создать m-файл, содержащий определение вектора-функции, стоящей в правой части систему ДУ (3.14), состоящий из следующих строк:

```
function dy=Angle1(t,y)
  global g % объявление глобальной переменной
  % задание вектора-столбца размерности 4x1
  dy=zeros(4,1);
  % задание координат вектора-функции,
  % стоящей в правой части (3.14)
  dy(1)=y(2);
  dy(2)=0;
  dy(3)=y(4);
  dy(4)=-g;
```

который следует сохранить на диске в файле `Angle1.m`.

Далее необходимо выполнить в командном окне следующую последовательность команд, сохраненную нами в файле `Glava3_2.m`:

```
>>  global g % объявление глобальной переменной
% задание начальных условий
>>  x0 = 0;
>>  vx0 = 2;
>>  y0 = 0;
>>  vy0 = 5;
% решение системы ДУ (3.14)
>>  [t,X]=ode45('Angle1',[0:0.01:1.019],[x0 vx0 y0 vy0]);
% построение зависимости в графическом окне No 1
>>  figure(1); plot(t,X(:,1)); title('x=x(t)');
% построение зависимости в графическом окне No 2
>>  figure(2); plot(t,X(:,3)); title('y=y(t)');
% построение зависимости в графическом окне No 3
>>  figure(3); plot(t,X(:,2)); title('vx=vx(t)');
% построение зависимости в графическом окне No 4
>>  figure(4); plot(t,X(:,4)); title('vy=vy(t)');
% построение зависимости в графическом окне No 5
>> figure(5); plot(X(:,1),X(:,3)); title('y=y(x)');
```

Вычисленные зависимости координат и скорости материальной точки от времени, а также ее траектория движения представлены на рис. 3.5–3.9.

Рис. 3.5. Зависимость $x = x(t)$

Рис. 3.6. Зависимость $y = y(t)$

Рис. 3.7. Зависимость $v_x = v_x(t)$

Рис. 3.8. Зависимость $v_y = v_y(t)$

Рис. 3.9. Траектория движения материальной точки

***Задача* 3.4**

Проведите сравнительный анализ аналитических и численных решений:

1. *Выберите начальные условия и решите систему уравнений* (3.14).

2. *Постройте зависимости* $x = x(t)$, $y = y(t)$, $v_x = x'(t)$, $v_y = y'(t)$, $v_x = v_x(x)$, $v_y = v_y(x)$, $y = y(x)$.

3. *Найдите аналитическое решение системы ДУ* (3.14). *Сравните зависимости* $x = x(t)$, $y = y(t)$, $v_x = x'(t)$, $v_y = y'(t)$, $v_x = v_x(x)$, $v_y = v_y(x)$, $y = y(x)$, *задаваемые аналитическими выражениями, с аналогичными зависимостями, полученными численным интегрированием.*

4. *Постройте зависимости перемещения материальной точки и угла между вектором скорости и осью OX от времени.*

5. *Постройте зависимости кинетической и потенциальной энергий от времени.*

6. *Проверьте полученное численное решение ДУ на соответствие закону сохранения энергии. С какой точностью выполняется закон сохранения энергии?*

3.3. ДВИЖЕНИЕ ТЕЛ В ГРАВИТАЦИОННОМ ПОЛЕ С УЧЕТОМ СИЛЫ ТРЕНИЯ

В предыдущем параграфе мы рассмотрели движение тел в гравитационном поле Земли без учета сил сопротивления, действующих на тело со стороны среды, через которую оно движется. При движении реальных тел в газовой или жидкой средах, как показывают результаты экспериментов, сила трения является главным фактором, который определяет характер движения тела.

О силе сопротивления, действующей на движущееся в среде тело, известно, что она (за исключением некоторых отдельных случаев) увеличивается с ростом скорости. Наиболее общими зависимостями силы сопротивления от скорости являются

$$F_d(v) = k_1 v, \tag{3.15}$$

$$F_d(v) = k_2 v^2. \tag{3.16}$$

где параметры k_1, k_2 зависят от свойств среды и геометрии тела.

Особо отметим, что зависимости (3.15), (3.16) не являются точными законами физики, но представляют собой феноменологические выражения, приближенно описывающие функцию $F_d(v)$ в некоторых диапазонах скоростей.

При относительно малых скоростях сила сопротивления описывается выражением (3.15). Для тел, имеющих относительно простую форму, известны аналитические выражения коэффициентов k_1. Например, для шарика радиуса r коэффициент сопротивления задается формулой Стокса

$$k_1 = 6\pi\mu r, \tag{3.17}$$

где μ — динамическая вязкость среды.

Числовые значения плотности и динамической вязкости различных сред при $t = 20°C$ и давлении 1 *атм* представлены в табл. 3.1.

Таблица 3.1. Числовые значения плотности и динамической вязкости выбранных сред при $t = 20°C$ и давлении 1 *атм*

	Вода	Воздух	Глицерин
ρ, кг/м^3	1,29	$1 \cdot 10^3$	$1,29 \cdot 10^3$
μ, (Н · с)/м^2	0,0182	1,002	1480

Представляет практический интерес оценить величину предельной скорости v_1 движения шара, падающего вертикально, то есть скорости движения, при которой сила тяжести будет равна силе трения:

$$6\pi\mu r v_1 = mg, \tag{3.18}$$

откуда

$$v_1 = \frac{mg}{6\pi\mu r} = \frac{4/3\pi r^3 \rho g}{6\pi\mu r} = \frac{2r^2\rho g}{9\mu}, \tag{3.19}$$

где ρ — плотность тела.

Значения предельной скорости v_1 для шара радиусом $r = 0,1$ м, $\rho = 0,8 \cdot 10^3$ кг/м^3 (дерево) представлены в табл. 3.2.

Таблица 3.2. Значения предельной скорости v_1 для шара радиусом $r = 0,1$ м, $\rho = 0,8 \cdot 10^3$ кг/м^3 (дерево), движущегося в выбранных средах

	Вода	Воздух	Глицерин
v, м/с	960	17	0,012

При больших скоростях сила сопротивления описывается выражением (3.16). Из экспериментов о величине k_2 известно, что она пропорциональна площади поперечного к потоку сечения тела S, плотности среды ρ и зависит от формы тела. Данную зависимость принято выражать следующей формулой:

$$k_2 = \frac{1}{2} c S \rho_0, \tag{3.20}$$

где c — безразмерный коэффициент (коэффициент формы), характеризующий лобовое сопротивление, ρ — плотность среды. Известные значения коэффициентов формы тел, представленных на рис. 3.10, приведены в табл. 3.3.

Таблица 3.3. Значения коэффициентов формы тел, представленных на рис. 3.10

Обозначение тела на рис. 3.10	а)	б)	в)	г)	д)	е)
Коэффициент формы	1,11	0,4	1,33	0,55	0,045	0,1

Проведем аналогичные оценки предельной скорости v_2 для квадратичного закона сопротивления. Приравнивая силу сопротивления и силу тяжести, имеем

$$cS\frac{\rho_0}{2} v_2^2 = mg, \tag{3.21}$$

откуда

$$v_2 = \sqrt{\frac{2mg}{\rho}} \tag{3.22}$$

или

$$v_2 = \sqrt{\frac{2\rho \cdot 4/3\pi r^3 g}{c\pi r^2 \rho_0}}. \tag{3.23}$$

Рис. 3.10. Примеры тел, для которых известны коэффициенты формы (стрелка указывает направление встречного потока)

Значения предельной скорости v_2 для шара радиусом $r = 0,1$ м, $\rho = 0,8 \cdot 10^3$ кг/м^3 (дерево) представлены в табл. 3.4.

Таблица 3.4. Значения предельной скорости v_2 для шара радиусом $r = 0,1$ м, $\rho = 0,8 \cdot 10^3$ кг/м^3 (дерево), движущегося в выбранных средах

	Воздух	Вода	Глицерин
v, м/с	18,0	0,65	0,58

Сравнение результатов, представленных в табл. 3.2, 3.3, показывает, что при движении тела в воздухе и воде $v_1 \gg v_2$. Это означает, что при движении тела в этих средах преобладающей будет сила сопротивления, квадратичная по скорости, которая сделает движение тела равномерным при значительно меньших скоростях, нежели линейная часть силы сопротивления.

Для дальнейшего анализа особенностей движения тел в гравитационном поле Земли при наличии силы сопротивления удобно измерять скорость движения в единицах установившейся скорости v_1, v_2. В выбранных единицах измерения выражения (3.15), (3.16) принимают следующий вид:

$$F_d = k_1 v_1 \left(\frac{v}{v_1}\right) = mg\left(\frac{v}{v_1}\right), \tag{3.24}$$

$$F_d = k_2 v_2 \left(\frac{v}{v_2}\right)^2 = mg\left(\frac{v}{v_2}\right)^2. \tag{3.25}$$

Рассмотрим падение тела вертикально вниз с нулевой начальной скоростью при наличии силы сопротивления. Так как сила сопротивления зависит от скорости, при аналитическом решении данной задачи удобно использовать уравнение движения в форме (3.2):

$$\frac{dv}{dt} = -\frac{mg - F_d}{m} = -\frac{mg - mg\left(\frac{v}{v_1}\right) - mg\left(\frac{v}{v_2}\right)^2}{m} = -g\left(1 - \frac{v}{v_1} - \left(\frac{v}{v_2}\right)^2\right), \tag{3.26}$$

которое, как очевидно, является ОДУ с разделяющимися переменными и может быть легко проинтегрировано аналитически. Повторным интегрированием зависимости $v = v(t)$ находят зависимость координаты материальной точки от скорости. Дополнив уравнение (3.26) уравнением

$$\frac{dx}{dt} = v, \tag{3.27}$$

получаем систему дифференциальных уравнений, которую можно решить в MATLAB численно.

В связи с тем, что выше мы подробно обсудили методы решения ДУ второго порядка в MATLAB, далее мы приводим только текст m-файла, содержащего описание вектора-функции, стоящей в правой части системы уравнений (3.26), (3.27):

```
function Z=Friction(t,y)
  global g,v1,v2 % определение
                 % глобальных переменных
  dy=zeros(2,1);
  % задание координат вектора-функции,
  % стоящей в правой части
  % системы (3.26), (3.27)
  dy(1)=y(2);
  dy(2)=-g*(1-y(2)/v1-(y(2)/v2)^2);
```

***Задача* 3.5**

1. *Для значений предельной скорости, представленных в табл. 3.2, 3.4, решите уравнение движения* (3.26) *и постройте зависимости* $x = x(t)$, $y = y(t)$, $v = v(y)$.

2. *Решите уравнение движения* (3.26)*, предполагая наличие в правой части только линейного и квадратичного членов. Сравните полученные решения с предыдущим решением. Какие можно сделать выводы?*

3. *Сравните зависимости* $x = x(t)$, $y = y(t)$, $v = v(y)$*, полученные без учета и при учете силы сопротивления. Можно ли считать реалистичной*

модель движения тела в гравитационном поле Земли, не учитывающую наличие силы трения?

4. *Решите уравнения движения, не предполагая постоянства силы тяжести.*

***Задача* 3.6**

Заметим, что уравнение движения (3.26) *описывает только частный случай — падение с нулевой начальной скоростью. Получим более общее уравнение движения, справедливое при любых начальных условиях. Так как сила сопротивления всегда направлена против скорости, то при движении тела в положительном направлении оси OY сила сопротивления отрицательна, и наоборот, при движении тела в отрицательном направлении оси OY сила сопротивления положительна, в то время как сила тяжести всегда направлена в отрицательном направлении оси OY. Эти условия наиболее просто выразить, используя знаковую функцию* $\mathrm{sgn}(x)$*:*

$$\mathrm{sgn}(x) = \begin{cases} 1, & \text{если } x \geq 0, \\ -1, & \text{если } x < 0. \end{cases} \tag{3.28}$$

При произвольном направлении движения вдоль оси OY выражения (3.24), (3.25) *для сил сопротивления принимают следующий вид:*

$$F_d = -\mathrm{sgn}(v) mg \left(\frac{v}{v_1}\right), \tag{3.29}$$

$$F_d = -\mathrm{sgn}(v) mg \left(\frac{v}{v_2}\right)^2. \tag{3.30}$$

Подставив (3.29), (3.30) в (3.26), получаем уравнение движения тела в виде

$$\frac{dv}{dt} = -\frac{mg - F_d}{m} = -g\left(1 + \mathrm{sgn}(v)\frac{v}{v_1} + \mathrm{sgn}(v)\left(\frac{v}{v_2}\right)^2\right). \tag{3.31}$$

1. *Для значений предельной скорости, представленных в табл. 3.2, 3.4, решите численно уравнение движения* (3.28) *и постройте зависимости* $x = x(t)$, $y = y(t)$, $v = v(y)$ *(в MATLAB функция, возвращающая знак переменной, имеет вид* `sign(имя_переменной)`*).*

2. *Решите уравнения движения* (3.28)*, предполагая наличие в правой части только линейного и квадратичного членов. Сравните полученные решения с предыдущим решением. Какие можно сделать выводы?*

3. *Сравните зависимости* $x = x(t)$, $y = y(t)$, $v = v(y)$*, полученные без учета и при учете силы сопротивления. Насколько отличаются максимальные высоты подъема тела при учете и без учета силы трения?*

***Задача* 3.7**

При движении материальной точки массой m с начальной скоростью $\vec{v}_0$, направленной под углом θ_0 к горизонту (рис. 3.4), на материальную точку действует сила тяжести и сила сопротивления $\vec{F}_d$, направленная противоположно скорости (рис. 3.10). Уравнения движения тела в данном случае имеют следующий вид:

$$\frac{d^2x}{dt^2} = -g\left(\frac{|v|}{v_1} + \left(\frac{|v|}{v_2}\right)^2\right)\cos\theta, \tag{3.32}$$

$$\frac{d^2y}{dt^2} = -g - g\left(\frac{|v|}{v_1} + \left(\frac{|v|}{v_2}\right)^2\right)\operatorname{sgn}(v_y)\sin\theta, \tag{3.33}$$

где $|v| = \sqrt{v_x^2 + v_y^2}$.

Так как $\cos\theta = v_x/|v|$, $\sin\theta = v_y/|v|$, *то уравнения системы ДУ* (3.32), (3.33) *преобразуются к виду*

$$\frac{d^2x}{dt^2} = -g\left(|v|/v_1 + (|v|/v_2)^2\right)\frac{1}{|v|}\frac{dx}{dt}, \tag{3.34}$$

$$\frac{d^2y}{dt^2} = -g - g\left(|v|/v_1 + (|v|/v_2)^2\right)\operatorname{sgn}\left(\frac{dy}{dt}\right)\frac{1}{|v|}\frac{dy}{dt}. \tag{3.35}$$

1. *Выберите начальные условия и решите систему уравнений* (3.34), (3.35).
2. *Постройте зависимости* $x = x(t)$, $y = y(t)$, $v_x = x'(t)$, $v_y = y'(t)$, $v_x = v_x(x)$, $v_y = v_y(x)$, $y = y(x)$.
3. *Постройте зависимости перемещения материальной точки и угла между вектором скорости и осью OX от времени.*
4. *Сравните полученные результаты с результатами задачи 3.1.*

ЛИТЕРАТУРА К ГЛАВЕ 3

1. *Гулд Х., Тобочник Я.* Компьютерное моделирование в физике. Ч. 1. М.: Мир, 1990.
2. *Хеннер Е. К., Шестаков А. П.* Математическое моделирование. Пермь: Пермский государственный педагогический университет, 1995.
3. *Матвеев А. Н.* Механика и теория относительности. М.: Высшая школа, 1986.
4. *Аппель П.* Теоретическая механика. М.: ГИФМЛ, 1960.

Глава 4

ЗАДАЧА КЕПЛЕРА

4.1. ВВЕДЕНИЕ

Задача о движении планет в поле тяжести небесных светил, являющаяся частным случаем задачи о движении в поле центральных сил, известна на протяжении нескольких тысячелетий истории человечества и в настоящее время рассматривается как в школьных курсах физики, астрономии, так и в вузовских курсах классической механики и астрономии.

Бóльшую часть наших знаний о движении планет объединили в себе законы Кеплера, полученные на основе анализа данных астрономических наблюдений, которые формулируются следующим образом:

1. Всякая планета движется по эллиптической орбите, в одном из фокусов которой находится Солнце.

2. Скорость планеты возрастает по мере удаления от Солнца таким образом, что прямая, соединяющая Солнце и планету, в равные промежутки времени заметает одинаковую площадь.

3. Для всех планет, вращающихся вокруг Солнца, отношение T^2/R^3 одинаково (T — период обращения планеты вокруг Солнца, R — большая полуось эллипса).

Отметим, что получить аналитическое решение задачи Кеплера удается только в случае рассмотрения движения двух тел, взаимодействующих по закону обратных квадратов. Это решение рассматривается во всех учебниках по классической механике, только малая часть из которых приведена в списке литературы [1–3]. Задача Кеплера для трех и более тел аналитического решения не имеет, может быть решена только численно, поэтому в этой главе основное внимание мы уделяем численному решению уравнений движения тела в центральном поле.

4.2. УРАВНЕНИЯ ДВИЖЕНИЯ ПЛАНЕТ

В качестве отправной точки решения задачи Кеплера рассмотрим движение двух тел, взаимодействующих друг с другом, считая их при этом материальными точками (рис. 4.1).

Функция Лагранжа такой системы имеет вид

$$L = \frac{m_1\dot{\vec{r}}_1^2}{2} + \frac{m_2\dot{\vec{r}}_2^2}{2} - U\left(|\vec{r}_1 - \vec{r}_2|\right) = \frac{m_1\dot{\vec{r}}_1^2}{2} + \frac{m_2\dot{\vec{r}}_2^2}{2} + \frac{\gamma\, m_1 m_2}{|\vec{r}_1 - \vec{r}_2|}, \qquad (4.1)$$

где $\vec{r}_1$, $\vec{r}_2$ — радиусы-векторы первого и второго тела соответственно, $U\left(|\vec{r}_1 - \vec{r}_2|\right)$ — потенциальная энергия взаимодействия тел, γ — гравитационная постоянная. Введем вектор взаимного расстояния обоих тел

$$\vec{r}_{12} = \vec{r}_1 - \vec{r}_2. \qquad (4.2)$$

Тогда в системе отсчета с началом координат в центре масс, рассматриваемой системы тел

$$m_1\vec{r}_1 + m_2\vec{r}_2 = 0. \qquad (4.3)$$

Из (4.2), (4.3) находим

$$\vec{r}_1 = \frac{m_2}{m_1 + m_2}\vec{r}_{12}, \qquad (4.4)$$

$$\vec{r}_2 = -\frac{m_1}{m_1 + m_2}\vec{r}_{12}. \qquad (4.5)$$

Рис. 4.1. К решению задачи Кеплера

Подставляя (4.4), (4.5) в (4.1), получаем

$$L = \frac{m\dot{\vec{r}}_{12}^2}{2} + U\left(|\vec{r}_{12}|\right) = \frac{m\dot{\vec{r}}_{12}^2}{2} + \frac{\gamma\, m\left(m_1 + m_2\right)}{|\vec{r}_{12}|}, \qquad (4.6)$$

где введено обозначение

$$m = \frac{m_1 m_2}{m_1 + m_2}. \qquad (4.7)$$

Величину, определяемую в соответствии с (4.7), принято называть приведенной массой. Функция (4.6) формально совпадает с функцией Лагранжа одной материальной точки с массой m, движущейся в потенциале $U\left(|\vec{r}_{12}|\right)$, симметричным относительно начала выбранной системы отсчета. Таким образом, задача о движении двух взаимодействующих тел сводится к решению задачи о движении одного тела с массой m в заданном внешнем поле $U\left(|\vec{r}_{12}|\right)$, создаваемом неподвижным центром с массой $m_1 + m_2$. Отметим, что если масса одного из взаимодействующих тел значительно меньше массы другого тела, последнее можно рассматривать как неподвижный притягивающий центр и найденная зависимость $\vec{r}_{12}\,(t)$ будет описывать траекторию движения более легкого тела. В противном случае, решив задачу о движении тела с массой m в потенциале $U\left(|\vec{r}|\right)$, по зависимости $\vec{r}\,(t)$ в соответствии с (4.4), (4.5) находят траектории каждой частицы $\vec{r}_1(t), \vec{r}_2(t)$.

Воспользовавшись уравнениями Лагранжа (здесь обобщенными координатами являются координаты радиуса-вектора $\vec{r}_{12}(t)$, обобщенными скоростями — координаты вектора $\dot{\vec{r}}_{12}(t)$)

$$\frac{d}{dt}\frac{\partial L}{\partial \dot{\vec{r}}_{12}} = \frac{\partial L}{\partial \vec{r}_{12}}, \qquad (4.8)$$

получим уравнения движения тела

$$m\frac{d^2\vec{r}_{12}}{dt^2} = -\frac{\gamma\, m(m_1+m_2)}{|\vec{r}_{12}|^3}\vec{r}_{12}, \tag{4.9}$$

которое при $m_1 \gg m_2$ в полном соответствии с законом всемирного тяготения Ньютона принимает вид:

$$m_2\frac{d^2\vec{r}_{12}}{dt^2} = -\frac{\gamma\, m_1 m_2}{|\vec{r}_{12}|^3}\vec{r}_{12}. \tag{4.10}$$

Отметим два важных свойства силы тяготения, вытекающих из (4.10):

1. сила зависит только от расстояния между телами;
2. сила направлена по прямой, проходящей через центры взаимодействующих тел (такие силы называются центральными).

Можно показать [1], что следствием указанных свойств является сохранение момента импульса тела

$$m\frac{d^2\vec{r}_{12}}{dt^2} = -\frac{\gamma\, m(m_1+m_2)}{|\vec{r}_{12}|^3}\vec{r}_{12}, \tag{4.11}$$

где $\vec{p} = m\dot{\vec{r}}_{12}$.

Сохранение момента импульса, в свою очередь, означает, что траектория движения тела в центральном поле лежит в плоскости, которой перпендикулярен вектор $\vec{L}$. Кроме того, движение тела ограничивается условиями сохранения полной энергии

$$E = \frac{1}{2}m\dot{\vec{r}}_{12}^2 - \frac{\gamma\, m(m_1+m_2)}{|\vec{r}_{12}|}. \tag{4.12}$$

и величины

$$\left[\dot{\vec{r}}_{12}\times\vec{L}\right] - \frac{\gamma\, m(m_1+m_2)\vec{r}_{12}}{|\vec{r}_{12}|} = \text{const}. \tag{4.13}$$

Рис. 4.2. Система координат, используемая для описания движения тел, под действием гравитационного взаимодействия

Для решения уравнений движения выберем прямоугольную систему координат, начало которой находится в центре масс (рис. 4.2). (Отметим, что в отличие от аналитического решения, наиболее просто получаемого в полярной системе координат, численное решение задачи Кеплера более удобно находить, используя декартовую систему координат.)

Уравнения движения (4.9) в выбранной системе координат имеют следующий вид

$$m\frac{d^2x_{12}}{dt^2} = -\frac{\gamma\, m(m_1+m_2)}{|\vec{r}_{12}|^3}x_{12}, \tag{4.14}$$

$$m\frac{d^2y_{12}}{dt^2} = -\frac{\gamma\, m(m_1+m_2)}{|\vec{r}_{12}|^3}y_{12}. \tag{4.15}$$

Введя обозначение $M = m_1 + m_2$ и сократив общие множители, запишем выражения (4.14), (4.15), составляющие систему ДУ второго порядка, в виде

$$\frac{d^2x_{12}}{dt^2} = -\frac{\gamma M}{(x_{12}^2 + y_{12}^2)^{3/2}} x_{12}, \tag{4.16}$$

$$\frac{d^2y_{12}}{dt^2} = -\frac{\gamma M}{(x_{12}^2 + y_{12}^2)^{3/2}} y_{12}. \tag{4.17}$$

Предваряя численное решение системы уравнений (4.16), (4.17), проведем обезразмеривание этих уравнений, выбрав в качестве единиц измерения расстояния радиус орбиты R и времени — период обращения T, соответствующие движению тела по окружности. Тогда можно ввести безразмерные переменные $X = x_{12}/R$, $Y = y_{12}/R$, $\tau = t/T$. Выполнив в (4.16), (4.17) замену переменных $x \to X$, $y \to Y$, $t \to \tau$, получаем

$$\frac{d^2X}{d\tau^2} = -\frac{\gamma M T^2}{R^3(X^2 + Y^2)^{3/2}} X, \tag{4.18}$$

$$\frac{d^2Y}{d\tau^2} = -\frac{\gamma M T^2}{R^3(X^2 + Y^2)^{3/2}} Y. \tag{4.19}$$

Как известно, при движении тела по окружности величина центростремительного ускорения a связана с радиусом круговой орбиты $|\vec{R}|$ и скоростью тела $|\vec{v}|$ соотношением

$$a = \frac{|\vec{v}|^2}{|\vec{R}|}. \tag{4.20}$$

При движении в гравитационном поле по окружности центростремительное ускорение обусловлено гравитационной силой. Следовательно,

$$\frac{m|\vec{v}|^2}{|\vec{R}|} = \frac{\gamma m M}{|\vec{R}|^2}, \tag{4.21}$$

откуда находим

$$|\vec{v}| = \left(\frac{\gamma M}{|\vec{R}|}\right)^{1/2}. \tag{4.22}$$

Выражение (4.21), являясь общим условием для любой круговой орбиты, позволяет найти зависимость периода движения от радиуса орбиты:

$$T = \frac{2\pi|\vec{R}|}{|\vec{v}|}, \tag{4.23}$$

поэтому, подставив в (4.22) выражение (4.21), получим

$$T = \sqrt{\frac{4\pi^2|\vec{R}|^3}{\gamma M}}. \tag{4.24}$$

Подставляя выражение (4.23) в (4.18), (4.19), окончательно получаем обезразмеренную систему уравнений:

$$\frac{d^2X}{d\tau^2} = -\frac{4\pi^2}{(X^2+Y^2)^{3/2}}X, \tag{4.25}$$

$$\frac{d^2Y}{d\tau^2} = -\frac{4\pi^2}{(X^2+Y^2)^{3/2}}Y. \tag{4.26}$$

Из уравнений (4.25), (4.26) видно, что они оказываются универсальными, поскольку в (4.25), (4.26) не входят ни период обращения тела вокруг центра поля, ни радиус орбиты. Следовательно, величина T^2/R^3, входящая в (4.18), (4.19), одинакова для всех тел, совершающих движение в гравитационном поле по замкнутым траекториям. Данный результат является доказательством справедливости третьего закона Кеплера.

Рис. 4.3. К выбору начальных условий численного интегрирования СДУ (4.25), (4.26)

При решении системы дифференциальных уравнений будем считать, что в начальный момент времени тело находилось в точке с радиусом-вектором $\vec{r} = (R, 0)$, скорость тела была направлена вертикально вверх $\vec{v} = (0, v)$ (рис. 4.3).

Так как система уравнений (4.25), (4.26) является безразмерной, необходимо также привести к безразмерному виду начальные условия. Выполнив, как и выше, замену переменных $\vec{r} = \vec{\rho} \cdot R$, $t = \tau \cdot T$, приводим начальные условия к следующему виду:

$$\vec{\rho} = (1, 0), \tag{4.27}$$

$$\vec{V} = \left(0, v\frac{T}{R}\right), \tag{4.28}$$

где T определяется выражением (4.24).

Однако использовать конкретные числовые значения R, T, M для проверки законов Кеплера не требуется, так как безразмерные начальные условия также обладают известным универсализмом. Для того чтобы это показать, найдем безразмерную скорость тела, движущегося в гравитационном поле по окружности. Подставив (4.23) в (4.28), получаем

$$\vec{V} = (0, 2\pi). \tag{4.29}$$

Из (4.29) видно, что для получения орбит, отличных от круговых достаточно задавать значения начальной скорости, отличные от 2π.

4.3. ЧИСЛЕННОЕ МОДЕЛИРОВАНИЕ ОРБИТЫ

Предваряя нахождение численного решения системы уравнений (4.25), (4.26), приведем ее к эквивалентной системе уравнений первого порядка, выполнив замену переменных $X \to z_1$, $X' \to z_2$, $Y \to z_3$, $Z' \to z_4$:

$$\frac{dz_1}{d\tau} = z_2, \tag{4.30}$$

$$\frac{dz_2}{d\tau} = -\frac{4\pi^2 \cdot z_1}{[z_1^2 + z_3^2]^{3/2}}, \tag{4.31}$$

$$\frac{dz_3}{d\tau} = z_4, \tag{4.32}$$

$$\frac{dz_4}{d\tau} = -\frac{4\pi^2 \cdot z_3}{[z_1^2 + z_3^2]^{3/2}}. \tag{4.33}$$

Следуя общим правилам MATLAB решения систем ОДУ первого порядка, создадим m-функцию `Orbit.m`, возвращающую значения координат вектора-функции, стоящей в левой части системы (4.30)–(4.33):

```
function dy=Orbit(t,z)
  dy=zeros(4,1); % задание вектора-столбца
                 % размерности 4x1
  % задание координат вектора-функции,
  % стоящей в правой части (4.30)-(4.33)
  dy(1)=z(2);
  dy(2)=-4*pi^2*z(1)/(z(1)^2+z(3)^2)^(3/2);
  dy(3)=z(4);
  dy(4)=-4*pi^2*z(3)/(z(1)^2+z(3)^2)^(3/2);
```

Далее необходимо выполнить в командном окне следующую последовательность команд, сохраненную нами в файле `Glava4_1.m`:

```
% задание начальных условий  x(0), x'(0), y(0), y'(0)
>>  x0=1;
>>  y0=0;
>>  vx0=0;
>>  vy0=2*pi*1.2;
% вычисление численного
% решения системы (4.30)-(4.33)
>>  [t,Y]=ode45('Orbit',[0:10^-3:2.5],[x0 vx0 y0 vy0]);
>>  figure(1); plot(t,Y(:,1)); % построение зависимости x(t)
>>  figure(2); plot(t,Y(:,2)); % построение зависимости x'(t)
>>  figure(3); plot(t,Y(:,3)); % построение зависимости y(t)
>>  figure(4); plot(t,Y(:,4)); % построение зависимости y'(t)
>>  figure(5); plot(Y(:,1),Y(:,3)); % построение траектории
                                   % движения тела
```

Зависимости кинематических характеристик от времени, а также и траектория движения представлены на рис. 4.4—4.8.

Рис. 4.4. Зависимость $x = x(t)$

Рис. 4.5. Зависимость $y = y(t)$

Рис. 4.6. Зависимость $x' = x'(t)$

Рис. 4.7. Зависимость $y' = y'(t)$

Для отображения на одном чертеже траектории материальной точки и притягивающего центра необходимо выполнить следующую последовательность команд:

```
>>  X0=0;Y0=0; % задание координат притягивающего центра
>>  plot(Y(:,1),Y(:,3),X0,Y0,'o','MarkerSize',5).
```

Для вычисления эксцентриситета орбиты необходимо выполнить следующую последовательность команд, сохраненную нами в файле `Glava4_2.m`:

```
>>  MinX=min(Y(:,1)); % вычисление минимального элемента
                      % в первом столбце матрицы Y
>>  MaxX=max(Y(:,1)); % вычисление максимального элемента
                      % в первом столбце матрицы Y
>>  a=(MaxX-MinX)/2; % вычисление длины первой полуоси
```

Рис. 4.8. Траектория движения тела

```
>> MinY=min(Y(:,3)); % вычисление минимального элемента
                     % во втором столбце матрицы Y
>>  MaxY=max(Y(:,1)); % вычисление максимального элемента
                     % во втором столбце матрицы Y
>>  b=(MaxY-MinY)/2; % вычисление длины второй полуоси
% вычисление эксцентриситета
>>  if b<=a
         e=(1-(b/a)^2)^0.5
      else
           e=(1-(a/b)^2)^0.5
      end
e =
    0.4386
```

4.4. ПРОВЕРКА ВТОРОГО ЗАКОНА КЕПЛЕРА

Для проверки второго закона Кеплера необходимо провести сравнение площади, заметаемой радиусом-вектором за равные промежутки времени, используя значения кинематических характеристик движения тела в гравитационном поле, полученные численным решением системы ДУ (4.30)–(4.33), возвращенные в матрицу `Y`. Напомним, что размерность матрицы `Y`, содержащей решения системы ДУ, равна $N \times 4$; нумерация столбцов и строк матрицы начинается с первого значения; значения зависимостей $x = x(t)$, $v = x'(t)$, $y = y(t)$, $y = y'(t)$ содержатся соответственно в первом, втором, третьем и четвертом столбцах матрицы.

При достаточно малом шаге интегрирования Δt уравнений движения (определяемом количеством точек N, в которых ищется решение системы дифференциальных уравнений, $\Delta t = T_{\text{finish}}/N$) площадь, заметаемая радиусом-вектором за время Δt, примерно равна площади треугольника с вершинами в точках $(0, 0)$, $(Y(n,\, 1), Y(n,\, 3))$, $(Y(n+1,\, 1), Y(n+1,\, 3))$ (см. рис. 4.9).

Рис. 4.9. К объяснению вычислительного алгоритма проверки второго закона Кеплера

Для вычисления площади треугольника, у которого заданы координаты вершин, можно воспользоваться формулой Герона:

$$S = \sqrt{p(p-a)(p-b)(p-c)}, \tag{4.34}$$

где p — полупериметр треугольника, изображенного на рис. 4.9, a, b, c — длины его сторон:

$$a = \sqrt{Y(n,\,1)^2 + Y(n,\,3)^2}, \tag{4.35}$$

$$b = \sqrt{Y(n+1,\,1)^2 + Y(n,\,3)^2}, \tag{4.36}$$

$$c = \sqrt{\left[Y(n+1,\,1) - Y(n,\,1)\right]^2 + \left[Y(n+1,\,3) - Y(n,\,3)\right]^2}. \tag{4.37}$$

Однако программа вычисления площади сектора, заметаемого радиусом-вектором, получается более компактной, если использовать встроенную в MATLAB функцию `cross`, вычисляющую векторное произведение трехмерных векторов. Как известно из аналитической геометрии, площадь треугольника, изображенного на рис. 4.9, равна

$$S = \frac{1}{2}\left|\left[\vec{a} \times \vec{b}\right]\right|, \tag{4.38}$$

поэтому алгоритм вычисления площади сектора, заметаемого радиусом-вектором, реализуется следующей последовательностью действий:

1. Задание начального момента времени и длины временного интервала, на котором вычисляется площадь, заметаемая радиусом-вектором.
2. Присвоение в момент времени $t(n)$ координатам вектора $\vec{a}$, соответствующих значений координат орбиты $\vec{a} = \begin{pmatrix} Y(n,\,1) \\ Y(n,\,3) \\ 0 \end{pmatrix}$.
3. Присвоение в момент времени $t(n)$ координатам вектора $\vec{b}$ соответствующих значений координат орбиты $\vec{b} = \begin{pmatrix} Y(n+1,\,1) \\ Y(n+1,\,3) \\ 0 \end{pmatrix}$.
4. Вычисление площади треугольника ΔS, заметаемого радиусом-вектором на временном интервале $[t(n), t(n+1)]$.

5. Нахождение площади, заметаемой радиусом-вектором на временном интервале $[t(1), t(n+1)]$, суммированием известной площади, заметаемой радиусом-вектором на интервале $[t(1), t(n)]$, и ΔS.

При составлении ниже описанной программы, реализующей описанный вычислительный алгоритм, мы исходили из следующих предположений:

1. Материальная точка в момент времени $t = 0$ находится в точке с радиусом-вектором $\begin{pmatrix} 1 \\ 0 \\ 0 \end{pmatrix}$.

2. Длительность временного интервала, на котором получено численное решение системы ДУ (4.30)–(4.33), такова, что материальная точка совершает не более 1,25 оборотов.

Так как нам заранее неизвестно точное значение периода одного оборота материальной точки вокруг притягивающего центра, в программе мы находим его численно, определяя при этом номер Nm соответствующего элемента вектора. Кроме того, следует учесть, что периодичность движения приводит к следующим правилам вычисления номера текущей координаты радиуса-вектора:

$$\begin{array}{rl} n, & \text{если} \quad n \le Nm, \\ n - Nm, & \text{если} \quad n > Nm. \end{array} \tag{4.39}$$

Далее мы приводим листинг m-файла `KeplerLow2.m`, содержащий функцию `KeplerLow2`, реализующую описанный выше алгоритм.

```
function z=KeplerLow2(t,Y,Tstart,dT)
%  функция, возвращающая площадь,
%  заметаемую радиусом-вектором
%  t,Y - переменные, возвращенные
%        одним из решателей систем ДУ
%  Tstart - начальный момент времени
%  dT - длительность временного интервала
dt=t(2)-t(1); % шаг интегрирования ДУ
Nstart=ceil(Tstart/dt)+1; % вычисление номера
                          % начальной координаты
dN=ceil(dT/dt); % число шагов
Nstop=Nstart+dN; % номер последней координаты
% вычисление значения Nm
i=length(Y);
while Y(1,1)>Y(i,1)
   Nm=i-1;
   i=i-1;
end;
Nlength=Nm;
% вычисление площади, заметаемой радиусом-вектором
```

```
s=0;
for i=Nstart:Nstop
  % проверка условий (4.39)
  if i<Nlength-1
    x1=[Y(i,1) Y(i,3) 0];
    x2=[Y(i+1,1) Y(i+1,3) 0];
  end;
  if i==Nlength
    x1=[Y(i,1) Y(i,3) 0];
    x2=[Y(1,1) Y(1,3) 0];
  end;
  if i>Nlength
    x1=[Y(i-Nlength,1) Y(i-Nlength,3) 0];
    x2=[Y(i+1-Nlength,1) Y(i+1-Nlength,3) 0];
  end;
  s=s+norm(cross(x1,x2))/2;
end;
z=s;
```

После нахождения численных решений системы ДУ (4.30)–(4.33) для обращения к функции `KeplerLow2` следует ввести команду

```
>> KeplerLow2(t,Y,1,0.1)
  ans =
    0.3811
>> KeplerLow2(t,Y,2,0.1)
  ans =
    0.3809
```

Представляет определенный интерес не только вычисление площади сектора, заметаемого радиусом-вектором, но и изображение данного сектора на рисунке вместе с траекторией тела. Для этого необходимо модифицировать функцию `KeplerLow2` таким образом, чтобы она возвращала помимо значения площади, заметаемой радиусом-вектором, координаты притягивающего центра и радиуса-вектора в соответствующие последовательные моменты времени, и затем использовать функцию `patch`. Ниже приведен листинг m-файла `KeplerLow.m`, содержащего модифицированную функцию `KeplerLow`:

```
function [z,Xr,Yr]=KeplerLow(t,Y,Tstart,dT)
% функция, возвращающая площадь,
% заметаемую радиусом-вектором,
% и координаты соответствующих точек орбиты
% t,Y - переменные, возвращенные одним
% из решателей систем ДУ
% Tstart - начальный момент времени
% dT - длительность временного интервала
dt=t(2)-t(1); % шаг интегрирования ДУ;
```

```
Nstart=ceil(Tstart/dt)+1; % вычисление номера
                          % начальной координаты
dN=ceil(dT/dt); % число шагов
Nstop=Nstart+dN; % вычисление номера
                 % начальной координаты
x(1)=0; y(1)=0; % координаты притягивающего центра
% вычисление значения Nm
i=length(Y);
while Y(i,1)>Y(i-1,1)
   Nm=i-1;
   i=i-1;
end;
Nlength=Nm;
% вычисление площади, заметаемой радиусом-вектором
s=0;
m=1;
for i=Nstart:Nstop
   m=m+1;
   % проверка условий (4.39)
   if i<Nlength-1
      x1=[Y(i,1) Y(i,3) 0];
      x2=[Y(i+1,1) Y(i+1,3) 0];
   end;
   if i==Nlength
      x1=[Y(i,1) Y(i,3) 0];
      x2=[Y(1,1) Y(1,3) 0];
   end;
   if i>Nlength
      x1=[Y(i-Nlength,1) Y(i-Nlength,3) 0];
      x2=[Y(i+1-Nlength,1) Y(i+1-Nlength,3) 0];
   end;
   s=s+norm(cross(x1,x2))/2;
   x(m)=x1(1);
   y(m)=x1(2);
end;
x(m+1)=x2(1);
y(m+1)=x2(2);
z=s;
Xr=x;
Yr=y;
```

Для вычисления площади, заметаемой радиусом-вектором, и визуализации сектора необходимо выполнить следующую последовательность команд:

```
>> [z,Xr,Yr]=KeplerLow(t,Y,0.5,0.5); % обращение к
                                     % функции KeplerLow
>> plot(Y(:,1),Y(:,3));  %  построение орбиты
```

```
>> patch(Xr,Yr,'g'); %  визуализация сектора
>> h = gca; % загрузка дискриптора графического окна
% задание режима отображения
% пропорциональных отрезков по
% обеим координатным осям
>> set(h,'DataAspectRatioMode','manual');
```

Результат выполнения описанной последовательности команд представлен на рис. 4.10.

Рис. 4.10. Траектория движения и сектор, заметаемый радиусом-вектором на временном интервале $[0,5;1]$

Задача 4.1

1. *Проведите расчеты при различных значениях начальной скорости и убедитесь, что для безразмерных скоростей незначительно меньших или превосходящих 2π орбита является замкнутым эллипсом, в одном из фокусов которого находится Солнце. Это является иллюстрацией правильности первого закона Кеплера.*

2. *Убедитесь в том, что при задании начальных условий $\vec{R}(0) = (1,0), \vec{v}(0) = (0,2\pi)$ орбита является круговой.*

3. *Вычислите значения нормировочных переменных для системы Солнце—Земля.*

4. *Реальные значения параметров движения орбиты Земли составляют: период обращения $T = 3,156 \cdot 10^7$ с, линейная скорость движения по орбите $v = 29,75 \cdot 10^3$ м/с, радиус орбиты Земли $R = 1,496 \cdot 10^{11}$ км. Используя эти данные, определите эксцентриситет орбиты Земли e и сравните это значение со значением, известным из эксперимента ($e = 0,0167$). Объясните обнаруженные отличия.*

5. *Определите значение начальной скорости, при которой происходит падение тела на центр поля. Чему равно значение полной энергии системы в этом случае? Каковы значения этой скорости для системы Солнце—Земля?*

6. *Определите значение начальной скорости, при которой тело может удалиться от притягивающего центра на бесконечно большое расстояние. Чему равно значение полной энергии системы в этом случае? Каковы значения этой скорости для системы Солнце—Земля?*

7. *Проведите численные эксперименты и найдите зависимость эксцентриситета орбиты и периода обращения от величины начальной скорости.*

8. *Изменяя номер начальной точки, убедитесь в выполнении третьего закона Кеплера.*

9. *Оцените точность, с которой выполняется второй закон Кеплера. Какой параметр определяет точность сохранения указанной величины?*

10. *Убедитесь в существовании интегралов движения, описываемых выражениями* (4.12), (4.13).

***Задача* 4.2**

Рассмотрите динамические эффекты, обусловленные малым отклонением силы притяжения от закона обратных квадратов (соответственно потенциала от закона $1/|\vec{r}|$*), предположив, что*

$$U(|\vec{r}|) \sim \frac{1}{|\vec{r}|^{1\pm\delta}}, \quad \text{где } \delta \ll 1.$$

1. *В чем отличие орбиты тела, двигающегося в новом потенциале, от орбиты тела, двигающегося в потенциале* $1/|\vec{r}|$ *? Убедитесь в том, что полученный результат не зависит от шага интегрирования.*

2. *При каких начальных условиях планета будет двигаться вокруг центра притяжения по раскручивающейся или по закручивающейся спирали?*

3. *Один из способов описания траектории движения тела — рассмотрение ее как эллиптической орбиты, которая медленно поворачивается (прецессирует) в направлении движения тела. Количественно прецессия характеризуется углом между прямыми, проходящими через последовательные перигелии и фокус эллипса (рис. 4.11). Оцените величину этого угла для выбранных параметров орбиты и значения* δ.

4. *Предложите алгоритм и составьте программу, позволяющую автоматизировать процедуру вычисления угла прецессии.*

5. *Используя составленную программу, зафиксируйте значение* φ *и исследуйте зависимость угла прецессии от длины большой полуоси* a *невозмущенного эллипса. Постройте график соответствующей зависимости.*

Рис. 4.11. К постановке задачи 4.2

Оцените функциональную зависимость угла прецессии от длины большой полуоси эллипса. (Указание: постройте график зависимости в логарифмическом масштабе $\varphi = \varphi(a)$[1]*.)*

6. *Используя составленную программу, зафиксируйте длину большой полуоси эллипса a и исследуйте зависимость угла прецессии φ от величины δ. Постройте график соответствующей зависимости. Оцените функциональную зависимость угла прецессии φ от величины δ. (Указание: постройте график зависимости $\varphi = \varphi(\delta)$ в полулогарифмическом масштабе.)*

***Задача* 4.3**

Для тела, совершающего движение в потенциале

$$U(|\vec{r}|) = \frac{1}{|\vec{r}|^2}.$$

1. *Запишите систему уравнений движения тела.*

2. *Для безразмерных начальных условий $x(0) = 1,\ y(0) = 0,\ x'(0) = 1$ аналитически получите значение безразмерной скорости, обеспечивающей движение тела по окружности.*

3. *Получите численное решение уравнений движения, постройте зависимость кинематических характеристик от времени и орбиту тела. Подберите шаг, обеспечивающий получение круговой орбиты в течение нескольких периодов.*

4. *Внесите незначительные изменения в значения начальной скорости, обеспечивающей существование круговой орбиты. Какой будет новая орбита? Чему равна величина полной энергии?*

[1] Для построения графика функции в полулогарифмическом масштабе используется функция `semilogy`. Например, для построения графика функции e^{-x} на интервале $[0; 1,5]$ нужно выполнить следующую последовательность команд:

```
» x = 0:0.01:1.5;
» semilogy(x,exp(-x))
```

***Задача* 4.4**

Используя описанную программу, проведите исследование влияния возмущений на орбитальную станцию, движущуюся по круговой орбите вокруг Земли:

1. *Как меняется орбита станции, если за счет включения на короткое время двигателя пристыкованной к ней ракеты-носителя станция получила приращение скорости в радиальном направлении?*

2. *Как зависит это изменение от величины приращения скорости в радиальном направлении и его длительности?*

3. *Установите, меняется ли угловой момент и полная энергия под действием радиального возмущения?*

4. *Будет ли орбита устойчивой, то есть будет ли малый переданный радиальный импульс вызывать малые возмущения орбиты, а также будет ли орбита оставаться периодической при отсутствии возмущений при дальнейшем движении?*

5. *Проведите исследования устойчивости в случае движения в тела в потенциале*
$$U(|\vec{r}|) = \frac{1}{|\vec{r}|^2}$$
под действием радиальных сил.

6. *Как меняется орбита станции, если за счет включения на короткое время двигателя пристыкованной к ней ракеты-носителя станция получила приращение скорости в касательном (тангенциальном) направлении?*

7. *Как зависит это изменение от величины приращения скорости в тангенциальном направлении и его длительности?*

8. *Установите, меняется ли угловой момент и полная энергия под действием тангенциального возмущения?*

9. *Будет ли орбита устойчивой, то есть будет ли малый переданный тангенциальный импульс вызывать малые возмущения орбиты, а также будет ли орбита оставаться периодической при отсутствии возмущений при дальнейшем движении?*

10. *Исследуйте устойчивость орбиты в случае движения в тела в потенциале*
$$U(|\vec{r}|) = \frac{1}{|\vec{r}|^2}$$
под действием тангенциальных сил.

***Задача* 4.5**

Известно, что от Солнца в направлении Земли существует поток частиц, называемый солнечным ветром, который приводит к возмущению орбиты движущегося тела. Воздействие солнечного ветра на движущееся тело можно рассматривать как действие слабой постоянной силы W, действующей на тело в горизонтальном направлении.

1. *Запишите уравнения движения, описывающие движение тела в гравитационном поле при учете солнечного ветра.*
2. *Промоделируйте орбиту при различных значениях величины W.*
3. *Как меняется орбита тела?*
4. *Что происходит с интегралами движения при учете солнечного ветра?*

4.5. ПРОСТРАНСТВО СКОРОСТЕЙ

Ранее в разделе 4.2 мы описали программу, позволяющую строить орбиту тела (график зависимости $y = y(x)$), двигающегося в потенциале

$$U(|\vec{r}|) \sim \frac{1}{|\vec{r}|}.$$

О данной зависимости говорят, что она построена в пространстве координат x, y. Можно поступить совершенно аналогично и построить график зависимости $v_y = v_y(v_x)$ (ряд последовательных значений вектора скорости тела), который представляет собой траекторию движения тела в пространстве скоростей v_x, v_y. Далее в этом разделе мы более подробно обсудим вопрос о том, какую форму имеет орбита в пространстве скоростей, если в пространстве координат орбита является эллипсом.

Для описания орбиты в пространстве скоростей воспользуемся интегралом движения (4.13), который в предположении $m_1 \gg m_2$ имеет вид

$$\left[\dot{\vec{r}}_{12} \times \vec{L}\right] - \frac{\gamma\, mM\, \vec{r}_{12}}{|\vec{r}_{12}|} = \left[\dot{\vec{r}}_0 \times \vec{L}\right] - \frac{\gamma\, mM\, \vec{r}_0}{|\vec{r}_0|}, \tag{4.40}$$

где, как и ранее, $M = m_1 + m_2$, $m_2 \equiv m_1$, $\vec{r}_0 = (x_0, y_0)$ — радиус-вектор тела в начальный момент времени. Запишем выражение (4.40) в координатной форме

$$\begin{pmatrix} mv_y|\vec{L}| - \gamma\, mMx/|\vec{r}| \\ -mv_x|\vec{L}| - \gamma\, mMy/|\vec{r}| \end{pmatrix} = \begin{pmatrix} mv_{y0}|\vec{L}| - \gamma\, mMx_0/|\vec{r}| \\ -mv_{x0}|\vec{L}| - \gamma\, mMy_0/|\vec{r}_0| \end{pmatrix}. \tag{4.41}$$

Из (4.41) найдем

$$\begin{aligned} x &= -r/\gamma\, mM \left(mv_y|\vec{L}| - A_{y0}\right), \\ y &= r/\gamma\, mM \left(mv_x|\vec{L}| - A_{x0}\right), \end{aligned} \tag{4.42}$$

где

$$A_{x0} = mv_{x0}|\vec{L}| - \gamma\, mMy_0/|\vec{r}_0|, \tag{4.43}$$

$$A_{y0} = mv_{y0}|\vec{L}| + \gamma\, mMx_0/|\vec{r}_0|. \tag{4.44}$$

Возведя в квадрат выражения для x, y в (4.42), сложив их, и учитывая, что

$$|\vec{r}|^2 = x^2 + y^2,$$

получаем

$$\left(v_x - A_{x0}/|\vec{L}|\right)^2 + \left(v_y - A_{y0}/|\vec{L}|\right)^2 = \left(\gamma\, mM/|\vec{L}|\right)^2. \tag{4.45}$$

Сравнив (4.45) с уравнением окружности радиуса R, центр которой смещен относительно начала координат в точку с координатами X_0, Y_0 (рис. 4.12)

$$(x - X_0)^2 + (y - Y_0)^2 = R^2, \tag{4.46}$$

приходим к выводу о том, что в пространстве скоростей орбита тела представляет собой окружность радиусом

$$R_v = \frac{\gamma\, mM}{|\vec{L}|}, \tag{4.47}$$

центр которой находится в точке

$$\left(\frac{A_{y0}}{|\vec{L}|}, \frac{A_{x0}}{|\vec{L}|}\right),$$

поэтому скорость частицы можно записать в виде

$$\vec{v} = \vec{u} + \vec{w}, \tag{4.48}$$

где

$$\vec{u} = \left(\frac{A_{y0}}{|\vec{L}|}, \frac{A_{x0}}{|\vec{L}|}\right), \tag{4.49}$$

$\vec{u}$ — вектор, проведенный из начала координат в центр окружности, $\vec{w}$ — вращающийся вектор, модуль которого равен радиусу окружности R_v.

Рис. 4.12. К объяснению формы траектории в пространстве скоростей

Отметим, что в каждый момент времени радиус-вектор планеты $\vec{r}$ перпендикулярен вектору $\vec{w}$. Для доказательства данного факта достаточно вычислить скалярное произведение векторов $\vec{r}$ и $\vec{w}$:

$$(\vec{r} \cdot \vec{w}) = (\vec{r} \cdot \vec{v}) - (\vec{r} \cdot \vec{u}). \tag{4.50}$$

Используя (4.42) и (4.49), получаем

$$(\vec{r} \cdot \vec{v}) = x \cdot v_x + y \cdot v_y = \frac{|\vec{r}|\,|\vec{L}|}{\gamma\, mM}\left(\frac{A_{y0}}{|\vec{L}|}v_x - \frac{A_{x0}}{|\vec{L}|}v_y\right), \tag{4.51}$$

$$(\vec{r}\cdot\vec{u}) = x\cdot u_x + y\cdot u_y = \frac{|\vec{r}|\,|\vec{L}|}{\gamma\, mM}\left(-\frac{A_{x0}}{|\vec{L}|}v_y + \frac{A_{y0}}{|\vec{L}|}v_x\right). \tag{4.52}$$

Подставив (4.51), (4.52) в (4.50), находим, что скалярное произведение векторов $\vec{r}$ и $\vec{w}$ действительно равно нулю, следовательно, векторы являются взаимно перпендикулярными.

Так как в разделе 4.2. было получено численное решение в безразмерных переменных, для проверки сделанных выше выводов необходимо найти выражения для безразмерных значений момента импульса $\vec{\tilde{L}}$, радиуса окружности $\tilde{R}_v$, а также выражений (4.43), (4.44) и, подставив далее в (4.45), получить уравнение окружности в безразмерных переменных.

Выбрав в качестве единиц измерения длины радиус круговой орбиты R, в качестве единиц измерения времени — период движения по круговой орбите T, найдем

$$\vec{L} = m[\vec{r}\times\vec{v}] = m\left[\left(\vec{\tilde{r}}\cdot R\right)\times\left(\vec{\tilde{v}}\cdot\frac{R}{T}\right)\right] = m\frac{R^2}{T}\left[\vec{\tilde{r}}\times\vec{\tilde{v}}\right] = m\frac{R^2}{T}\vec{\tilde{L}}, \tag{4.53}$$

где $\vec{\tilde{r}} = \frac{\vec{r}}{R}$, $\vec{\tilde{v}} = \vec{v}\frac{T}{R}$.

Подставляя (4.53) в (4.47) и учитывая (4.22), (4.24), находим

$$\tilde{R}_v = \frac{4\pi^2}{\left|\vec{\tilde{L}}\right|}, \tag{4.54}$$

$$\tilde{A}_{x0} = \tilde{v}_{x0} - \frac{4\pi^2\tilde{y}_0}{\left|\vec{\tilde{r}}\right|\,\left|\vec{\tilde{L}}\right|}, \tag{4.55}$$

$$A_{y0} = \tilde{v}_{y0} + \frac{4\pi^2\tilde{x}_0}{\left|\vec{\tilde{r}_0}\right|\,\left|\vec{\tilde{L}}\right|}. \tag{4.56}$$

Более подробное рассмотрение параметров орбит в пространстве скоростей можно найти в [6]. Геометрическое рассмотрение происхождения замкнутых орбит при движении тел в полях, обратно пропорциональных расстоянию до центра, проведено в [7].

***Задача* 4.6**

1. *Получите самостоятельно выражения* (4.54)–(4.56).

2. *Убедитесь в том, что орбита в пространстве скоростей является окружностью, хотя в пространстве координат она является эллипсом.*

3. *Дополните документ, в котором проводится решение уравнений движения тела в центральном поле, выражениями, позволяющими вычислять координаты центра окружности и ее радиус.*

4. *Используя численное решение уравнений движения, проверьте справедливость выражений* (4.54)–(4.56).

5. *Убедитесь в том, что в каждый момент времени радиус-вектор $\vec{r}$ планеты перпендикулярен вектору скорости $\vec{w}$. Как можно объяснить данный факт?*

6. *Исследуйте, как изменяется орбита в пространстве скоростей под действием импульсного удара в тангенциальном направлении.*

7. *Исследуйте, как изменяется орбита в пространстве скоростей под действием импульсного удара в радиальном направлении.*

8. *Используя представление о том, что в каждый момент времени радиус-вектор $\vec{r}$ планеты перпендикулярен вектору скорости $\vec{w}$, объясните наблюдаемое изменение орбиты в системе пространственных координат под действием импульсных ударов в тангенциальном и радиальном направлениях.*

9. *Исследуйте изменение орбиты в пространстве скоростей под действием солнечного ветра.*

4.6. МОДЕЛИРОВАНИЕ СОЛНЕЧНОЙ СИСТЕМЫ

Ранее мы занимались моделированием орбит, рассматривая задачу двух сил. Однако реальные планетные системы не являются системами двух тел, так как между всеми телами действуют силы гравитационного взаимодействия. Несмотря на то что силы гравитационного взаимодействия малы по сравнению с силой притяжения к центральному светилу, в ряде случаев они приводят к реально наблюдаемым эффектам. Например, на основании несовпадения экспериментальных данных об орбите планеты Уран и теоретических расчетов было предсказано существование планеты Нептун.

Учет взаимного влияния других тел, находящихся в гравитационном поле, создаваемом телом большой массы (M), означает, что полная сила, действующая на планету, уже не является центральной. Кроме того, в действительности орбиты планет могут не лежать в одной плоскости (как, например, в Солнечной системе), поэтому для проведения точных расчетов необходимо решать задачу в трехмерном пространстве. Однако для понимания последствий влияния сил гравитационного взаимодействия между движущимися телами оказывается достаточно рассмотреть двумерную модель, состоящую из двух тел массой m_1, m_2.

Для описания движения тел этой системы выберем прямоугольную систему координат, представленную на рис. 4.13. В выбранной системе координат в предположении, что $M \gg m_1 + m_2$, функция Лагранжа имеет вид:

$$L = \frac{m_1 \dot{\vec{r}}_1^{\,2}}{2} + \frac{m_2 \dot{\vec{r}}_2^{\,2}}{2} + \frac{\gamma\, m_1 M}{|\vec{r}_1|} + \frac{\gamma\, m_2 M}{|\vec{r}_2|} + \frac{\gamma\, m_1 m_2}{|\vec{r}_{12}|}. \qquad (4.57)$$

Рис. 4.13. К постановке задачи моделирования Солнечной системы

Воспользовавшись уравнениями Лагранжа (4.8), получим уравнения движения тел:

$$m_1 \frac{d^2\vec{r}_1}{dt^2} = -\frac{\gamma\, m_1 M}{|\vec{r}_1|^3}\vec{r}_1 - \frac{\gamma\, m_1 m_2}{|\vec{r}_{12}|^3}\vec{r}_{12}, \tag{4.58}$$

$$m_2 \frac{d^2\vec{r}_1}{dt^2} = -\frac{\gamma\, m_2 M}{|\vec{r}_1|^3}\vec{r}_1 + \frac{\gamma\, m_1 m_2}{|\vec{r}_{12}|^3}\vec{r}_{12}. \tag{4.59}$$

Для нахождения численного решения системы уравнений (4.58), (4.59), как и было сделано выше, следует провести обезразмеривание уравнений движения. Выбрав в качестве единиц измерения длины и времени величины, соответствующих движению по круговой орбите тела массой в гравитационном потенциале, создаваемом телом массой M, получаем

$$\frac{d^2X1}{d\tau^2} = -\frac{4\pi^2 X1}{((X2-X1)^2+(Y2-Y1)^2)^{3/2}} - \frac{4\pi^2\delta_1(X2-X1)}{((X2-X1)^2+(Y2-Y1)^2)^{3/2}}, \tag{4.60}$$

$$\frac{d^2Y1}{d\tau^2} = -\frac{4\pi^2}{((X2-X1)^2+(Y2-Y1)^2)^{3/2}}Y1 - \frac{4\pi^2\delta_1(Y2-Y1)}{((X2-X1)^2+(Y2-Y1)^2)^{3/2}}, \tag{4.61}$$

$$\frac{d^2X2}{d\tau^2} = -\frac{4\pi^2 X1}{((X2-X1)^2+(Y2-Y1)^2)^{3/2}} + \frac{4\pi^2\delta_2(X2-X1)}{((X2-X1)^2+(Y2-Y1)^2)^{3/2}}, \tag{4.62}$$

$$\frac{d^2Y2}{d\tau^2} = -\frac{4\pi^2 Y1}{((X2-X1)^2+(Y2-Y1)^2)^{3/2}} + \frac{4\pi^2\delta_2(Y2-Y1)}{((X2-X1)^2+(Y2-Y1)^2)^{3/2}}. \tag{4.63}$$

При решении системы дифференциальных уравнений будем считать, что в начальный момент времени тело массой m_1 находилось в точке с радиусом-вектором $\vec{r}_1 = (R_1, 0)$, скорость тела была направлена вертикально вверх $\vec{v}_1 = (0, v_1)$, тело массой m_2 находилось в точке с радиусом-вектором $\vec{r}_2 = (R_2, 0)$ ($R_2 = R_1 + \Delta R$), скорость тела была направлена вертикально вверх $\vec{v}_2 = (0, v_2)$ (рис. 4.14).

Безразмерные начальные условия имеют вид

$$\begin{aligned} &\vec{\rho}_1 = (1, 0),\\ &\vec{\rho}_2 = (1 \pm \tfrac{\Delta R}{R_1}, 0),\\ &\vec{v}_1 = (0, v_1 \tfrac{R_1}{T_1}),\\ &\vec{v}_2 = (0, v_2 \tfrac{R_1}{T_1}). \end{aligned} \tag{4.64}$$

Рис. 4.14. К выбору начальных условий в задаче моделирования Солнечной системы

***Задача* 4.7**

Найдите решение системы уравнений движения (4.58), (4.59), *считая, что параметры орбит и массы движущихся тел совпадают с соответствующими параметрами Земли и Луны.*

1. *Постройте зависимости координат и скоростей движения обоих тел от времени.*

2. *Используя вычисленные значения координат, постройте зависимости координат от времени и орбиту Луны в системе отсчета, связанной Землей. Сравните полученные результаты с данными о движении Луны, приведенными в главе 1.*

***Задача* 4.8**

Исследуйте особенности движения системы трех тел, считая, что $\Delta R = 0,6$, $\delta_1 = 10^{-2}$, $\delta_2 = 10^{-3}$, $\vec{v}_1 = (0, 2\pi)$, $\vec{v}_1 = (0, 2\pi/1,6)$.

1. *Постройте орбиты планет, считая, что взаимодействием между ними можно пренебречь. Чему равны их периоды обращения вокруг центра поля?*

2. *Постройте орбиты планет, решив систему* (4.55), (4.56). *В чем качественно проявляется их взаимодействие? Опишите форму орбит. Почему влияние взаимодействия на одну из планет меньше, чем на другую?*

3. *Проверьте, сохраняются ли интегралы движения отдельно для первой и второй планет. Сохраняются ли полная энергия и полный момент обеих планет?*

4. *Используя реальные астрономические данные, оцените степень влияния Юпитера на орбиту Земли.*

ЛИТЕРАТУРА К ГЛАВЕ 4

1. *Ландау Л. Д., Лифшиц Е. М.* Курс теоретической физики. Механика. М.: Физматгиз, 1958.

2. *Гольдстейн Г.* Классическая механика. М.: Мир, 1974.

3. *Аппель П.* Теоретическая механика. М.: ГИФМЛ, 1960.

4. *Гулд Х., Тобочник Я.* Компьютерное моделирование в физике. Ч. 1. М.: Мир, 1990.

5. *Хеннер Е. К., Шестаков А. П.* Математическое моделирование. Пермь: Пермский государственный педагогический университет, 1995.

6. *H. Abelson, A. di Sessa, L. Rudolph*, Velocity Space the Geometry of Planetary Orbits, Am. J. Phys., 43, 579 (1975).

7. *A. di Sessa*, Orbits: A mini-environment for exploring orbital mechanics, in: Computers in Edication, eds. O. Lecarrme, R. Lewis, 359. North-Holland, 1975.

Глава 5

МОДЕЛИРОВАНИЕ СТАТИЧЕСКИХ ЭЛЕКТРИЧЕСКИХ И МАГНИТНЫХ ПОЛЕЙ

5.1. ЭЛЕКТРОСТАТИЧЕСКОЕ ПОЛЕ СИСТЕМЫ НЕПОДВИЖНЫХ ЭЛЕКТРИЧЕСКИХ ЗАРЯДОВ

Как известно, электрическое поле, создаваемое в вакууме неподвижным точечным электрическим зарядом q, в данной точке пространства характеризуется скалярным потенциалом

$$\varphi(\vec{R}) = \frac{1}{4\pi\varepsilon_0}\frac{q}{|\vec{R}-\vec{r}|}, \tag{5.1}$$

где $\vec{R}$ — радиус-вектор точки наблюдения, $\vec{r}$ — радиус-вектор точки, в которой находится электрический заряд, $\varepsilon_0 = 8,85 \cdot 10^{-12}$ *ф/м*. Векторной характеристикой данного поля является напряженность $\vec{E}$

$$\vec{E} = -\vec{\nabla}\varphi(\vec{R}) = \frac{1}{4\pi\varepsilon_0}\frac{q}{|\vec{R}-\vec{r}|^3}(\vec{R}-\vec{r}). \tag{5.2}$$

Сила взаимодействия $\vec{F}$ двух точечных электрических зарядов q_1 и q_2 описывается законом Кулона:

$$\vec{F} = \frac{1}{4\pi\varepsilon_0}\frac{q_1 q_2}{|\vec{r}_1-\vec{r}_2|^3}(\vec{r}_1-\vec{r}_2), \tag{5.3}$$

где $\vec{r}_1$, $\vec{r}_2$ — радиусы-векторы, зарядов соответственно.

Скалярный потенциал электрической системы, состоящей из N электрических зарядов $q_1, q_2, \ldots, q_N$, и напряженность электрического поля $\vec{E}$ удовлетворяют принципу суперпозиции:

$$\varphi(\vec{R}) = \frac{1}{4\pi\varepsilon_0}\sum_{i=1}^{N}\frac{q_i}{|\vec{R}-\vec{r}_i|}, \tag{5.4}$$

$$\vec{E}(\vec{R}) = \frac{1}{4\pi\varepsilon_0}\sum_{i=1}^{N}\frac{q_i}{|\vec{R}-\vec{r}_i|^3}(\vec{R}-\vec{r}_i), \tag{5.5}$$

где $\vec{r}_i$ — координата i-го заряда.

При анализе электростатических полей системы произвольно расположенных зарядов, характеризующихся скалярной функцией — потенциалом $\varphi(\vec{R})$ и векторной

функцией — напряженностью $\vec{E}$, возникает задача наглядного представления этих величин. Один из возможных способов представления потенциала электростатического поля реализуется следующей последовательностью действий:

1) задание функции, возвращающей значения потенциала, вычисляемые в соответствии (5.4) в узлах заданной координатной сетки;

2) задание дискретной координатной сетки;

3) вычисление в соответствии с (5.3) значений $\varphi(\vec{R})$ в каждом узле координатной сетки;

4) построение графика поверхности и карты эквипотенциалей.

Следуя последовательности действий, описанной выше, создадим m-файл, содержащий описание функции $\varphi(\vec{R})$:

```
function Phi=Potential(q,xq,yq,X,Y);
% задание функции, возвращающей
% значения потенциала в узлах
% координатной сетки,
% вычисляемых в соответствии с (5.4)
% q - вектор, содержащий значения
% электрических зарядов
% xq, yq - векторы, содержащие x-е и y-е координаты
% электрических зарядов
% X, Y - векторы, содержащие
% x-е и y-е координаты узлов сетки

e0=8.85*10^-12; % диэлектрическая проницаемость вакуума
Nq=length(q); % число зарядов в системе
Nx=length(X); % число узлов по оси oX
Ny=length(Y); % число узлов по оси oY
% проход по каждому узлу сетки
for i=1:Ny
   for j=1:Nx
      s=0;
      % суммирование по зарядам
      for k=1:Nq
         s=s+q(k)/((X(j)-xq(k))^2+(Y(i)-yq(k))^2)^0.5;
      end;
      M(i,j)=s/(4*pi*e0);
   end;
end;
Phi=M;
```

Функция `Potential.m` возвращает матрицу размерности $Ny \times Nx$, содержащую значения потенциала в соответствующих узлах координатной сетки.

Рис. 5.1. К вычислению потенциала, создаваемого линейной системой точечных зарядов

Вычислим потенциал, создаваемый системой, состоящей из $N = 50$ электрических зарядов, расположенных в точках с координатами (рис. 5.1)

$$(-5R0 + \frac{10R0}{N} \cdot i), \quad y_i = 0, \quad i = 0, \ldots, N.$$

Для проведения вычислений в качестве единиц измерения заряда будем использовать заряд электрона $e = 1,6 \cdot 10^{-16}$ *Кл*, единиц измерения длины — $R0 = 10^{-6}$ *м*.

Решение данной задачи в MATLAB выполняется следующей последовательностью команд, сохраненной нами в файле `Glava5_1.m`:

```
>>  e=1.6*10^-16; % заряд электрона
>>  R0=10^-6; % единица измерения расстояния
>>  N=50; % число зарядов
>>  i=1:N;
>> q(i)=e; % заполнение вектора,
           % содержащего значения зарядов
>> x1=-5*R0; % x-я координата левого
             % конца системы зарядов
>> x2=5*R0; % x-я координата правого конца
            % системы зарядов
>> xq=x1+(x2-x1)/N*i; % вычисление x-х координат
                      % системы зарядов
|>> yq=zeros(1,N); % задание y-х координат
                  % системы зарядов
>> N1=79; % число узлов прямоугольной
          % координатной сетки
>> Xmin=-10*R0; Ymin=-10*R0; % задание координат
                             % нижнего левого
                             % угла координатной сетки
>> Xmax=10*R0; Ymax=10*R0; % задание координат
                           % верхнего правого
                           % угла координатной сетки
>> i=1:N1;
>> X=Xmin+(Xmax-Xmin)/N1*i; % вычисление x-х координат
                            % узлов сетки
```

```
>> j=1:N1;
>> Y=Ymin+(Ymax-Ymin)/N1*j; % вычисление y-x координат
                            % узлов сетки
>> M=Potential(q,xq,yq,X,Y); % вычисление значений потенциала
                             % в узлах координатной сетки
>> [X1,Y1]=meshgrid(X,Y); % вычисление матриц X1,Y1,
                          % используемых функциями визуализации
                          % двумерных и трехмерных зависимостей
>> surf(X1,Y1,M); % визуализация поверхности
```

Рис. 5.2. Визуализация потенциала, создаваемого системой линейных зарядов с помощью функции `surf`

Результат выполнения приведенной выше последовательности команд представлен на рис. 5.2.

Для построения отображения поверхности без изображения сетки и заполнения поверхности с использованием интерполяции цвета дополнительно следует ввести команды:

```
>> figure(2);
>> surf(X1,Y1,M);
>> shading interp
```

Результаты их выполнения представлены на рис. 5.3.

Для построения двумерной карты линий уровня (эквипотенциалей) функции $\varphi = \varphi(x, y)$ следует ввести команды:

```
>> figure(3);
>> contour(X1,Y1,M,33)
% здесь 33 - число
% линий уровня
```

Рис. 5.3. Визуализация потенциала, создаваемого системой линейных зарядов (результат выполнения команды `shading interp`)

Результат выполнения приведенных выше команд представлен на рис. 5.4.

Для вывода значений соответствующей каждой эквипотенциали необходимо выполнить следующую последовательность команд:

```
>> figure(4);
>> [C,h] = contour(X1,Y1,M);
```

```
>> clabel(C,h)
>> colormap cool
```

Результат выполнения приведенных выше команд представлен на рис. 5.5.

Здесь команда `contour` возвращает матрицу `C`, содержащую координаты вершин ломаных, аппроксимирующих соответствующие эквипотенциальные линии, вектор-столбец **h** дескрипторов для каждой линии уровня, используемый далее командой `clabel`, и строит карту линий уровня. Далее команда `clabel`, используя данные, возвращенные функцией `contour`, подписывает значения функции на каждой линии уровня. Команда `colormap` определяет цвет заливки карты линий уровня.

Рис. 5.4. Карта эквипотенциалей функции $\varphi = \varphi(x, y)$

Рис. 5.5. Карта эквипотенциалей функции $\varphi = \varphi(x, y)$

Для отображения на одном чертеже поверхности и карты линий уровня следует выполнить команды:

```
>> figure(5);
>> meshc(X1,Y1,M)
```

Поверхность и соответствующая карта линий уровня представлены на рис. 5.6.

Рис. 5.6. Результат выполнения функции `meshc(X1,Y1,M)`

В связи с тем, что поверхность, представленая на рис. 5.7, практически полностью закрывает карту линий уровня, мы перешли в режим редактирования графика (`Edit Plot`) и, предварительно выделив щелчком мыши изображение поверхности, удалили ее. Результат выполнения описанной выше последовательности действий представлен на рис. 5.7.

Рис. 5.7. Проекция карты линий уровня на плоскость $z = 0$

Для построения трехмерной карты линий уровня (эвипотенциалей) функции $\varphi = \varphi(x, y)$ следует выполнить команды:

```
>> figure(6);
>> contour3(X1,Y1,M,11);
```

Результаты выполнения данной последовательности команд представлены на рис. 5.8.

Рис. 5.8. Проекция карты линий уровня на плоскость $z = 0$

В отличие от функции $\vec{\varphi} = \varphi(\vec{R})$, напряженность электростатического поля $\vec{E}(\vec{R})$ является векторной функцией, которая в каждой точке пространства характеризуется величиной поля и направлением. Для описания векторного поля будем использовать силовые линии — линии, касательные к которым в каждой точке параллельны вектору напряженности электрического поля. В связи с тем, что силовые линии несут только информацию о направлении вектора напряженности электрического поля, но не его величине, для анализа изменения величины напряженности электрического поля в пространстве будем использовать функцию $|\vec{E}(\vec{R})|$.

Таким образом, для исследования особенностей напряженности электрического поля, создаваемого произвольной конфигурацией электрических зарядов, следует:

1) задать в пространстве дискретную координатную сетку;

2) вычислить в узлах сетки координаты напряженности электрического поля $\vec{E}(\vec{R}) = (E_x, E_y)$;

3) построить в каждом узле сетки единичные векторы $\vec{n} = \left(\frac{E_x}{|\vec{E}|}, \frac{E_y}{|\vec{E}|}\right)$, касательные к силовой линии электрического поля;

4) построить график поверхности и карту линий уровня функции $|\vec{E}(\vec{R})|$.

Данный алгоритм в MATLAB может быть реализован следующей последовательностью команд, сохраненной нами в файле `Glava5_2.m`:

```
>> e=1.6*10^-16; % заряд электрона
>> R0=10^-6; % единица измерения расстояния
>> N=50; % число зарядов
>> i=1:N;
>> q(i)=e; % заполнение вектора,
           % содержащего значения зарядов
>> x1=-5*R0; % x-я координата левого конца
             % системы зарядов
>> x2=5*R0; % x-я координата правого конца
            % системы зарядов
>> xq=x1+(x2-x1)/N*i; % вычисление x-x координат
                      % системы зарядов
|>> yq=zeros(1,N); % задание y-x координат
                   % системы зарядов
>> N1=23; % число узлов прямоугольной
          % координатной сетки
>> Xmin=-10*R0; Ymin=-10*R0; % задание координат
                             % нижнего левого
                             % угла координатной сетки
>> Xmax=10*R0; Ymax=10*R0; % задание координат
                           % верхнего правого
                           % угла координатной сетки
>> i=1:N1;
>> X=Xmin+(Xmax-Xmin)/N1*i; % вычисление x-x координат
                            % узлов сетки
>> j=1:N1;
>> Y=Ymin+(Ymax-Ymin)/N1*j; % вычисление y-x координат
                            % узлов сетки
>> M=Potential(q,xq,yq,X,Y); % вычисление значений потенциала в
                             % узлах координатной сетки
>> [X1,Y1]=meshgrid(X,Y); % вычисление матриц X1,Y1,
                          % используемых функциями визуализации
                          % двумерных и трехмерных зависимостей
>> [px,py]=gradient(-M,0.1,0.1); % вычисление градиента функции
% вычисление координат единичных векторов, касательных
% к силовой линии
>> px1=px./(px.^2+py.^2).^0.5;
>> py1=py./(px.^2+py.^2).^0.5;
```

```
>> figure(1);
>> quiver(X1,Y1,px1,py1,0.5) % построение векторного
                             % поля единичных
                             % векторов, касательных к
                             % силовой линии,
                             % в узлах координатной сетки
>> mp=(px.^2+py.^2).^0.5; % вычисление абсолютных
                          % значений вектора
                          % напряженности
                          % в узлах координатной сетки
>> figure(2);
>> contour(X1,Y1,mp,17); % построение линий
                         % равной напряженности
```

Результаты выполнения приведенной выше последовательности команд представлены на рис. 5.9 (векторное поле), рис. 5.10 (линии равной напряженности).

Рис. 5.9. Визуализация силовых линий напряженности электростатического поля, создаваемого линейной системой зарядов

Рис. 5.10. Карта линий равной напряженности электростатического поля, создаваемого линейной системой зарядов

***Задача* 5.1**

Создайте документ, позволяющий строить потенциал и напряженность электрического поля диполя, считая, что составляющие его электрические заряды одинаковы по величине и противоположны по знаку.

1. *Имеет ли в данной задаче значение, в каких единицах измеряется заряд и расстояние?*

2. *Как изменится электрическое поле, если заряды будут иметь одинаковые знаки?*

3. *Убедитесь в том, что силовые линии никогда не заканчиваются на положительных зарядах и всегда идут к отрицательным зарядам. Почему силовые линии никогда не пересекаются?*

Рис. 5.11. К вычислению напряженности электростатического поля, создаваемого квадруполем

***Задача* 5.2**

Создайте файл, позволяющий строить потенциал и напряженность электрического поля квадруполя, то есть электрическую систему, состоящую из четырех равных по абсолютной величине электрических зарядов, расположенных в вершинах квадрата, считая, что электрические заряды, расположенные в соседних вершинах, имеют разный знак (рис. 5.11).

1. *Как выглядит электрическое поле вблизи квадруполя (ближняя зона)?*

2. *Как выглядит электрическое поле на большом расстоянии от квадруполя (дальняя зона)?*

3. *Исследуйте зависимость электрического поля в ближней и дальней зонах от расстояния до центра системы и расстояния между зарядами.*

***Задача* 5.3**

Создайте файл, позволяющий строить потенциал и напряженность электрического поля протяженного диполя, то есть диполя, состоящего из N (>10) зарядов, знаки которых последовательно чередуются.

1. *Сравните электрическое поле при четном и нечетном числе зарядов, составляющих диполь, в ближней и дальней зонах.*

2. *Сравните электрическое поле при четном и нечетном числе зарядов, составляющих диполь, с электрическим полем точечного заряда.*

3. *Исследуйте зависимость электрического поля протяженного диполя в ближней и дальней зонах от расстояния между зарядами.*

***Задача* 5.4**

Рассмотрите однородно заряженную квадратную пластину со стороной L и полным зарядом Q, лежащую в плоскости XOY (рис. 5.12). Как известно, в пределе при $L \to \infty$ при постоянной плотности заряда $\sigma = Q/L^2$ вектор напряженности электрического поля $\vec{E}$ направлен по нормали к пластине, $|\vec{E}| = \sigma/2\varepsilon_0$. Для нахождения электрического поля заряженной пластины конечных размеров разбейте пластину сеткой из $N \times N$ ячеек так, чтобы каждую ячейку можно было считать точечным зарядом величиной $dq = Q/N^2$.

Рис. 5.12. К расчету электростатического поля, создаваемого заряженной пластиной

1. *Создайте документ, позволяющий вычислять потенциал электрического поля заряженной пластины конечных размеров.*

2. *Выберите точку и постройте зависимость потенциала электрического поля от числа ячеек. Оцените размер ячейки, начиная с которого его уменьшение оказывает заметное влияние на точность вычислений. Как зависит этот размер ячейки от координаты точки наблюдения?*

3. *Для выбранных размеров ячейки исследуйте электрическое поле на различных расстояниях от пластины.*

4. *Проведите сравнение электрического поля заряженной пластины конечных размеров с электрическим полем, создаваемым бесконечной пластиной и точечным зарядом.*

5.2. МАГНИТНОЕ ПОЛЕ ВИТКА С ПОСТОЯННЫМ ТОКОМ

Изучение электрических полей систем неподвижных электрических зарядов, проведенное нами в предыдущем разделе, было основано на законе Кулона (5.3) и принципе суперпозиции (5.4), (5.5). Закон Кулона в электростатике имеет аналог в магнитостатике — закон Био–Савара–Лапласа, в соответствии с которым напряженность магнитного поля, создаваемого элементом тока $I\vec{dl}$, имеет вид

$$d\vec{B} = \frac{\mu_0}{4\pi} I \frac{[\vec{dl} \times \vec{R}]}{|\vec{R}|^3}, \tag{5.6}$$

где I — величина тока, измеряется в амперах A, напряженность магнитного поля $\vec{B}$ — в теслах (Тл), $\mu_0 = 4\pi \cdot 10^{-7} м/A$ — магнитная проницаемость вакуума.

Закон Био–Савара–Лапласа имеет в магнитостатике совершенно общий характер и вместе с принципом суперпозиции может быть использован для нахождения напряженности магнитного поля, создаваемого произвольной системой постоянных токов, в любой точке пространства. Действительно, так как напряженность магнитного поля, создаваемого элементом тока Idl, расположенном в точке с радиусом-вектором $\vec{r}$, в точке наблюдения с радиусом-вектором $\vec{R}$ равна

$$d\vec{B} = \frac{\mu_0}{4\pi} I \frac{[\vec{dl} \times (\vec{R} - \vec{r})]}{|\vec{R} - \vec{r}|^3}, \tag{5.7}$$

то для вычисления напряженности магнитного поля, создаваемого произвольной системой постоянных токов, следует задать ориентацию каждого элемента тока , входящего в структуру, и вычислить напряженности магнитных полей, создаваемых каждым элементом тока $I\vec{dl}$, и, в соответствии с принципом суперпозиции, произвести суммирование напряженностей магнитных полей:

$$\vec{B} = \frac{\mu_0}{4\pi} I \sum_{i=1}^{N} \frac{[\vec{dl_i} \times (\vec{R} - \vec{r_i})]}{|\vec{R} - \vec{r_i}|^3}, \tag{5.8}$$

где N — число элементов, $\vec{r_i}$ — радиус-вектор соответствующего элемента.

Отметим, что вычислить магнитное поле аналитически, используя (5.7), (5.8), оказывается возможным только для структур, имеющих достаточно высокую степень симметрии (прямой провод, поле кольца на оси симметрии и т. д.). Для большинства токовых конфигураций расчет магнитного поля может быть проведен только численно. Наиболее важными геометрическими конфигурациями токовых структур, для которых необходимо знать распределение напряженности магнитного поля, являются: прямой провод, петля, соленоид, тороидальная обмотка. Далее мы опишем подход,

позволяющий вычислять магнитное поле петли с постоянным током, и проведем критический анализ известных моделей, используемых при расчете магнитных полей соленоида и тороидальной обмотки с постоянным током.

Для расчета напряженности магнитного поля петли с током выберем систему координат, в которой плоскость XOY совпадает с плоскостью петли (рис. 5.13).

Рис. 5.13. К расчету напряженности магнитного поля петли с током

В выбранной системе координат положение i-го элемента тока $Id\vec{l_i}$,

$$d\vec{l_i} = Ia \begin{bmatrix} \sin\varphi_i \\ -\cos\varphi_i \\ 0 \end{bmatrix} d\varphi, \qquad (5.9)$$

где $d\varphi = \frac{2\pi}{N}$, $\varphi_i = id\varphi$, $i = 0, 1, \ldots, N-1$, N — число элементов тока, описывается радиусом-вектором $\vec{r_i}$:

$$\vec{r_i} = a \begin{bmatrix} \cos\varphi_i \\ \sin\varphi_i \\ 0 \end{bmatrix}. \qquad (5.10)$$

Подставив (5.9), (5.10) в (5.8), нетрудно получить выражения для составляющих вектора напряженности магнитного поля B_x, B_y, B_z, однако для расчета напряженности и визуализации магнитного поля в MATLAB этого не требуется, так как в пакет встроены функции для вычисления стандартных действий векторного анализа: вычисление скалярного произведения, векторного произведения и т. д.

На примере вычисления и визуализации напряженности магнитного поля, создаваемого витком с током в плоскости $x = 0$, опишем последовательность действий, позволяющих решить поставленную задачу. В рассматриваемой плоскости, как очевидно из соображений симметрии, составляющая напряженности магнитного поля вдоль оси OX равна нулю, поэтому далее мы проводим вычисления только составляющих напряженности вдоль осей OY, OZ. Так как далее нас будет интересовать форма силовых линий, но не абсолютное значение напряженностей, множитель будем полагать равным единице. Сначала необходимо создать файл `Ring.m`, содержащий описание функции, возвращающей значения составляющих напряженности магнитного поля, листинг которого мы приводим ниже:

```
function [By,Bz]=Ring(a,Y,Z)
% функция, возвращающая значения
% проекций напряженности магнитного
% поля на оси OY, OZ, вычисляемые
% в соответствии с (5.7)
Nstep=100; % количество элементов,
           % аппроксимирующих кольцо
```

```
X=0;
Ny=length(Y); % число узлов
              % координатной сетки
              % по оси OY
Nz=length(Z); % число узлов
              % координатной сетки
              % по оси OZ
deltaphi=2*pi/Nstep; % шаг по углу
n=1:Nstep+1;
phi(n)=deltaphi*(n-1); вычисление вектора
% вычисление составляющих
% напряженности магнитного поля
% в соответствии с (5.7)
for i=1:Nz
   for j=1:Ny
      s=[0 0 0];
      for n=1:Nstep+1
         dL=[-a*sin(phi(n))*deltaphi...
             a*cos(phi(n))*deltaphi...
             0];
          r=[a*cos(phi(n)) a*sin(phi(n)) 0];
          R=[X Y(j) Z(i)];
          s=s+cross(dL,(R-r))./((R-r)*(R-r)').^(3/2);
      end;
      by(i,j)=s(2);
      bz(i,j)=s(3);
   end;
end;
By=by;
Bz=bz;
```

Далее необходимо выполнить следующую последовательность команд, сохраненную нами в файле `Glava5_3.m`:

```
N1=21; % число узлов сетки
i=1:N1+1;
Ymin=-5; Zmin=-5; % нижний левый угол
                  % координатной сетки
Ymax=5; Zmax=5; % верхний правый угол
                % координатной сетки
% вычисление координат узлов сетки
Y(i)=Ymin+(Ymax-Ymin)/N1*(i-1);
Z(i)=Zmin+(Zmax-Zmin)/N1*(i-1);
a=1; % радиус кольца
[By Bz]=Ring(a,Y,Z); % вычисление проекций
                     % напряженности магнитного поля
```

```
mp=(By.^2+Bz.^2).^0.5; % вычисление значений модуля
                       % напряженности магнитного поля
                       % в узлах координатной сетки
% вычисление координат единичных векторов, касательных к
% силовой линии магнитного поля, в узлах координатной сетки
by1=By./mp;
bz1=Bz./mp;
quiver(Y,Z,by1,bz1); % визуализация векторного поля
```

Результаты выполнения приведенной выше последовательности команд представлены на рис. 5.14. Анализ рис. 5.14 позволяет сделать вывод о том, что поле единичных

Рис. 5.14. Поле единичных векторов, касательных к силовой линии магнитного поля, создаваемого витком с током

векторов позволяет на качественном уровне получать информацию о форме силовых линий и направления напряженности магнитного поля в данной точке координатной сетки.

Определенный недостаток технологии визуализации силовых линий магнитного поля состоит в том, что линия, являющаяся по своей физической сути непрерывной, заменяется набором векторов, касательных к данной линии. Для получения более адекватного изображения силовых линий магнитного поля можно использовать векторный

потенциал магнитного поля $\vec{A}$, связанный с напряженностью магнитного поля следующим соотношением:

$$\vec{B} = rot\vec{A}. \tag{5.11}$$

Используя (5.11), можно показать [22], что силовая линия есть линия, на которой $|\vec{A}(x, y, z)| = \text{const}$. Следовательно, для построения силовой линии магнитного поля необходимо вычислить модуль векторного потенциала в узлах координатной сетки и затем построить карту эквипотенциалей.

Векторный потенциал магнитного поля, создаваемого элементом тока $Id\vec{l}$, равен

$$d\vec{A} = \frac{\mu_0}{4\pi}\frac{Id\vec{l}}{|\vec{R} - \vec{r}|}, \tag{5.12}$$

где $\vec{R}$ — радиус-вектор точки наблюдения, $\vec{r}$ — радиус-вектор элемента тока.

Векторный потенциал магнитного поля $\vec{A}$, создаваемого произвольной системой токов $Id\vec{l}_i$, в соответствии с принципом суперпозиции, равен

$$\vec{A} = \sum_{i=1}^{N} d\vec{A}_i = \frac{\mu_0}{4\pi}\sum_{i=1}^{N}\frac{Id\vec{l}_i}{|\vec{R} - \vec{r}_i|}. \tag{5.13}$$

Для практической реализации предложенного алгоритма, как и ранее, сначала необходимо создать файл `RingA.m`, содержащий описание функции, возвращающей значения составляющих напряженности магнитного поля, листинг которого мы приводим ниже:

```
function A=RingA(a,Y,Z)
% функция, возвращающая
% значения модуля вектора-потенциала
% магнитного поля, вычисляемого
% в узлах координатной сетки,
% расположенной в плоскости YOZ,
% в соответствии с (5.13)
Nstep=100; % количество элементов,
           % аппроксимирующих кольцо
X=0;
Ny=length(Y); % число узлов
              % координатной сетки по оси OY
Nz=length(Z); % число узлов
              % координатной сетки по оси OZ
deltaphi=2*pi/Nstep; % шаг по углу
n=1:Nstep+1;
phi=deltaphi*(n-1); % вычисление вектора
% вычисление модуля напряженности
% вектора-потенциала магнитного поля
for i=1:Nz
  for j=1:Ny
```

```
    s=[0 0 0];
    for n=1:Nstep+1
      dL=[-a*sin(phi(n))*deltaphi a*cos(phi(n))*deltaphi 0];
      r=[a*cos(phi(n)) a*sin(phi(n)) 0];
      R=[X Y(j) Z(i)];
      s=s+dL./((R-r)*(R-r)').^(1/2);
    end;
    ax(i,j)=s(1);
    ay(i,j)=s(2);
    az(i,j)=s(3);
  end;
end;
A=(ax.^2+ay.^2+az.^2).^0.5;
```

Далее необходимо выполнить следующую последовательность команд:

```
N1=21; % число узлов сетки
i=1:N1+1;
Ymin=-5; Zmin=-5; % нижний левый угол
                  % координатной сетки
Ymax=5; Zmax=5; % верхний правый угол
                % координатной сетки
% вычисление координат узлов сетки
Y=Ymin+(Ymax-Ymin)/N1*(i-1);
Z=Zmin+(Zmax-Zmin)/N1*(i-1);
a=1; % радиус кольца
mp=RingA(a,Y,Z); % вычисление модуля
                 % векторного потенциала
contour(Y,Z,mp,55);
```

Результаты выполнения приведенной выше последовательности команд представлены на рис. 5.15.

Рис. 5.15. Карта эквипотенциалей функции $|\vec{A}|$

Анализ карты эквипотенциальных поверхностей, представленных на рис. 5.15, позволяет выявить два недостатка. Во-первых, на карте неправильно отображаются силовые линии, проходящие вблизи центра кольца, что объясняется особенностями вычислительного алгоритма. Во-вторых, силовые линии оказываются негладкими, что обусловлено использованием координатной сетки с достаточно большим шагом и использованием MATLAB при построении силовых линий магнитного поля последовательных отрезков прямых. Устранить второй недостаток

достаточно просто, используя какую-либо интерполяцию исходных данных на координатной сетке меньшего размера. Для этого необходимо выполнить следующую последовательность команд, сохраненную нами в файле `Glava5_4.m`:

```
% задание новой координатной сетки
>>  yi=Ymin:(Ymax-Ymin)/1000:Ymax;
>>  zi=Zmin:(Zmax-Zmin)/1000:Zmax;
% формирование матриц,
% используемых функцией interp2
>>  [y1 z1]=meshgrid(Y,Z);
>>  [y2 z2]=meshgrid(yi,zi);
% вычисление значений
% векторного потенциала в узлах
% координатной сетки меньшего
% размера с использованием
% бикубической сплайн-интерполяции
>>   zi3 = interp2(y1,z1,mp,y2,z2,'bicubic');
>>  contour(yi,zi,zi3,55); % визуализация силовых линий
```

Результат выполнения приведенной выше последовательности команд представлен на рис. 5.16.

Рис. 5.16. Карта эквипотенциалей функции $|\vec{A}|$

Таким образом, сочетание обоих описанных в настоящем разделе методов визуализации силовых линий магнитного поля позволяет получить адекватное представление об их геометрии.

***Задача* 5.5** Как известно зависимость напряженность поля, создаваемого магнитным диполем $\vec{M}$, описывается формулой

$$\vec{B}(\vec{R}) := \frac{3(\vec{M}\cdot\vec{R})}{|\vec{R}|^5} - \frac{\vec{M}}{|\vec{R}|^3}. \tag{5.14}$$

1. *Исследуйте особенности магнитного поля, создаваемого магнитным диполем. Рассмотрите магнитные поля на различных расстояниях от диполя.*

2. *Сравните магнитные поля, создаваемые кольцом с током и магнитным диполем. Какой вывод можно сделать?*

5.3. МАГНИТНОЕ ПОЛЕ СОЛЕНОИДА С ПОСТОЯННЫМ ТОКОМ

Вопрос о структуре магнитного поля однослойного соленоида с постоянным током обсуждается практически во всех учебниках — от школьных до ставших в настоящее время классическими вузовских учебников по электричеству, магнетизму [1–5] и электродинамике [6–10]. Расчет поля на оси соленоида, допускающий решение в аналитическом виде, в случае бесконечно длинного соленоида и соленоида конечной длины предлагается в качестве задач во многих задачниках [11–13].

Соленоид, согласно устоявшейся точке зрения, рассматривается как источник магнитного поля. Один конец соленоида является северным полюсом, другой — южным, поле вне соленоида считается аксиально симметричным, а внутри катушки в приближении бесконечно длинного соленоида — однородным. Общий вид силовых линий данного магнитного поля показан на рис. 5.17, что подтверждается элементарными опытами со стрелкой компаса и магнитными опилками, располагаемыми на горизонтальной плоской поверхности с малым коэффициентом трения, проходящей через продольную ось соленоида.

Рис. 5.17. Карта эквипотенциалей функции $|\vec{A}|$

Известный метод расчета напряженности магнитного поля однослойного соленоида состоит в представлении соленоида набором последовательно расположенных колец и суммировании в соответствии с принципом суперпозиции напряженностей магнитных полей, создаваемых каждым кольцом в выбранной точке пространства. При таком подходе неявно предполагается, что ток при плотной намотке вдоль продольной оси соленоида отсутствует или исчезающе мал по сравнению с током вдоль кольца. Это послужило причиной, по которой магнитное поле, обусловленное составляющей тока вдоль оси соленоида, считалось малым по сравнению с магнитным полем, обусловленным движением электрических зарядов вдоль кольца. Данное допущение, однако, недостаточно корректно.

Доказательство высказанного утверждения проведем в два этапа; на первом построим качественную картину протекания тока по соленоиду, на втором этапе проведем количественные расчеты. Причем, в отличие от традиционного подхода,

Рис. 5.18. К вычислению скорости движения электрона в спиральной намотке

обсужденного выше, рассмотрим изменение условий движения носителей заряда при последовательном усложнении геометрии токовой структуры (прямой ток—спираль—соленоид). Оставаясь в рамках электронной теории, рассмотрим ток в проводе как последовательность движущихся с постоянной скоростью зарядов. Очевидно, что намотка провода не меняет скорость движения носителей заряда вдоль провода $\vec{v}_0$, однако появляются составляющие скорости носителей заряда вдоль координатных осей (рис. 5.18). При этом движение носителей заряда вдоль оси OZ будет движением с постоянной скоростью $\vec{v}_\perp$, в плоскости XOY — с постоянной скоростью по окружности. Выразим данные скорости через радиус спирали a и шаг спирали h, для чего рассмотрим развертку одного витка спирали в плоскости XOZ (рис. 5.19).

Из рис. 5.19 очевидно, что

$$v_z = v_0 \cos\alpha = v_0 \frac{h}{\sqrt{(2\pi a)^2 + h^2}}, \tag{5.15}$$

$$v_\perp = v_0 \frac{2\pi a}{\sqrt{(2\pi a)^2 + h^2}}. \tag{5.16}$$

Учитывая, что шаг спирали h выражается через длину соленоида L_s и число витков N как $h = L_s/N$, и переходя в (5.15), (5.16) к токам, получаем:

Рис. 5.19. Развертка одного витка спиральной обмотки

$$I_z = N_e e v_0 \frac{L_s}{\sqrt{(2\pi a N)^2 + L_s^2}}, \tag{5.17}$$

$$I_z = N_e e v_0 \frac{2\pi a N}{\sqrt{(2\pi a N)^2 + L_s^2}}, \tag{5.18}$$

где N_e — линейная плотность электронов в соленоиде (число электронов в проводе, намотанном на соленоид, отнесенное к длине соленоида); e — заряд электрона.

Кроме того, намотка провода приводит к увеличению длины провода с током L, которая выражается через число витков и шаг намотки

$$L = N\sqrt{(2\pi a)^2 + h^2} = \sqrt{(2\pi a N)^2 + L_s^2}. \tag{5.19}$$

Анализ выражений (5.17), (5.18) показывает, что поле соленоида с однослойной намоткой является суперпозицией полей двух токов, один из которых является круговым, второй — продольным. Увеличение числа витков приводит, с одной стороны, к уменьшению шага спирали и, следовательно, к уменьшению величины продольного тока (см. выражения (5.17), (5.18)), с другой стороны, к увеличению длины провода, намотанного на соленоид, что приводит к возрастанию количества носителей заряда, двигающихся в соленоиде, т. е. возрастанию линейной плотности заряда. Из выражений (5.17) и (5.18) видно, что количество электричества, перемещающееся в направлении оси OZ в единицу времени остается неизменным при увеличении числа витков соленоида. Таким образом, при любом количестве витков (любом шаге намотки) количество электричества, перемещающегося в направлении продольной оси соленоида в единицу времени оказывается равным количеству электричества в прямом проводе. Следовательно, соленоид с однослойной обмоткой должен обладать аксиально симметричным магнитным полем в плоскости, перпендикулярной продольной оси соленоида, подобным полю прямого тока. Учитывая аналогию с магнитным полем прямого тока, естественно в дальнейшем называть его поперечным полем — в отличие от магнитного поля, приведенного на рис. 5.17. При большом числе витков соленоид можно рассматривать как проводящий цилиндр конечных размеров, по которому течет продольный ток, следовательно, можно ожидать появления отличия поперечного магнитного поля соленоида и магнитного поля прямого тока, которое должно проявляться при увеличении числа витков.

Для проверки рассуждений, приведенных выше, получим аналитическое выражение для напряженности магнитного поля соленоида на основе закона Био–Савара–Лапласа (5.6) и принципа суперпозиции, проинтегрировав (5.6) по всем элементам тока.

Как известно [14], положение любой точки спирали полностью определяется заданием единичного вектора $\vec{n}$, направленного вдоль оси спирали (ось OZ), и вектора $\vec{a}$, отмечающего положение точки спирали в плоскости, перпендикулярной к вектору $\vec{n}$ (рис. 5.20).

Тогда положение любой точки спирали определяется радиусом-вектором $\vec{R}$, зависящим от одного параметра — угла поворота φ:

$$\vec{R}(\varphi) = \vec{n}h\frac{\varphi}{2\pi} + \vec{a}\cos\varphi + [\vec{n}\times\vec{a}]\sin\varphi. \tag{5.20}$$

Рис. 5.20. К вычислению напряженности магнитного поля спиральной обмотки

Элемент тока $Id\vec{l}$ выражается через вектор $\vec{t}(\varphi)$, касательный к спирали, как

$$Id\vec{l} = I\vec{t}(\varphi)d\varphi, \tag{5.21}$$

где

$$\vec{t}(\varphi) = \frac{h}{2\pi}\vec{n} - \vec{a}\sin\varphi + [\vec{n}\times\vec{a}]\cos\varphi. \tag{5.22}$$

Обозначим вектор, направленный из начала координат в точку наблюдения, через R_0 , тогда вектор $r(\varphi) = \vec{R}_0 - \vec{R}(\varphi)$. Таким образом, напряженность магнитного поля в точке наблюдения записывается в виде

$$\vec{B} = \frac{\mu_0 I}{4\pi} \int\limits_0^{2\pi N} \frac{\left[\vec{t}(\varphi) \times \left(\vec{R}_0 - \vec{R}(\varphi)\right)\right]}{\left|\vec{t}(\varphi) \times \left(\vec{R}_0 - \vec{R}(\varphi)\right)\right|^3} d\varphi. \tag{5.23}$$

Для вычисления интеграла (5.23) перейдем к координатной записи. Заметив, что в выбранной системе координат векторы $\vec{n}$, $\vec{a}$, $[\vec{n} \times \vec{a}]$, входящие в (5.20)–(5.23), имеют координаты $\vec{n} = (0, 0, 1)$, $\vec{a} = (1, 0, 0)$, $[\vec{n} \times \vec{a}]$, получим выражения для вычисления координат векторов $\vec{R}(\varphi)$, $\vec{t}(\varphi)$, $\vec{R}_0 - \vec{R}(\varphi)$:

$$\vec{R}(\varphi) = \left(a\cos\varphi, a\sin\varphi, \frac{h}{2\pi}\varphi\right), \tag{5.24}$$

$$\vec{t}(\varphi) = \left(-a\sin\varphi, a\cos\varphi, \frac{h}{2\pi}\right), \tag{5.25}$$

$$\vec{R}_0 - \vec{R}(\varphi) = \left(x_0 - a\cos\varphi, y_0 - a\sin\varphi, z_0 - \frac{h}{2\pi}\varphi\right). \tag{5.26}$$

Подставляя (5.24)–(5.26) в (5.23), получаем для составляющих напряженности магнитного поля соленоида вдоль соответствующих координатных осей:

$$B_x = \frac{\mu_0 I}{4\pi} \int\limits_0^{2\pi N} \frac{\left(\left(z_0 - \frac{h}{2\pi}\varphi\right) a\cos\varphi - \frac{h}{2\pi}\left(y_0 - a\sin\varphi\right)\right)}{\left(R_0^2 + a^2 + \left(\frac{h}{2\pi}\varphi\right)^2 - 2\left(x_0 a\cos\varphi + a y_0 \sin\varphi + \frac{h}{2\pi} z_0 \varphi\right)\right)^{3/2}} d\varphi, \tag{5.27}$$

$$B_y = \frac{\mu_0 I}{4\pi} \int\limits_0^{2\pi N} \frac{\left(\left(x_0 - a\cos\varphi\right) \frac{h}{2\pi} + a\sin\varphi\left(z_0 - \frac{h}{2\pi}\varphi\right)\right)}{\left(R_0^2 + a^2 + \left(\frac{h}{2\pi}\varphi\right)^2 - 2\left(x_0 a\cos\varphi + a y_0 \sin\varphi + \frac{h}{2\pi} z_0 \varphi\right)\right)^{3/2}} d\varphi, \tag{5.28}$$

$$B_z = -\frac{\mu_0 I}{4\pi} \int\limits_0^{2\pi N} \frac{\left(\left(y_0 - a\sin\varphi\right) a\sin\varphi + a\cos\varphi\left(x_0 - a\cos\varphi\right)\right)}{\left(R_0^2 + a^2 + \left(\frac{h}{2\pi}\varphi\right)^2 - 2\left(x_0 a\cos\varphi + a y_0 \sin\varphi + \frac{h}{2\pi} z_0 \varphi\right)\right)^{3/2}} d\varphi. \tag{5.29}$$

В связи с тем, что соленоид имеет цилиндрическую форму, представляется целесообразным записать компоненты поля в цилиндрической системе координат. Для этого положение точки наблюдения будем характеризовать в плоскости, перпендикулярной продольной оси соленоида, радиусом-вектором $\vec{r}_0$ и углом поворота φ_0, положение относительно продольной соленоида — z_0 координатой. Тогда выражения для напряженности составляющих поля в цилиндрической системе координат находятся соответствующей заменой переменных в (5.27)–(5.29): $x_0 \to r_0\cos\varphi_0$, $y_0 \to r_0\sin\varphi_0$,

$R_0^2 \to r_0^2 + z_0^2$ и переходом из декартовой в цилиндрическую систему координат, определяемым матрицей поворота относительно оси OZ [10]:

$$\begin{pmatrix} B_r \\ B_\varphi \\ B_z \end{pmatrix} = \begin{pmatrix} \cos\varphi_0 & \sin\varphi_0 & 0 \\ -\sin\varphi_0 & \cos\varphi_0 & 0 \\ 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} B_x \\ B_y \\ B_z \end{pmatrix}. \tag{5.30}$$

Описанная выше замена переменных и выражение (5.30) позволяют легко получить окончательные выражения для составляющих напряженности магнитного поля в цилиндрической системе координат. В связи с тем, что процедура нахождения составляющих напряженности магнитного поля в цилиндрической системе координат теперь является тривиальной, а окончательные выражения весьма громоздки, автор их не приводит.

Отметим, что выражения (5.27)–(5.29) получены абсолютно строго, без использования каких-либо приближений, и позволяют проводить вычисления для соленоидов с произвольными параметрами (длиной, поперечным сечением и числом витков). Их анализ показывает, что интегралы в (5.27)–5.29 всегда отличны от нуля. Следовательно, однослойный соленоид имеет поперечное магнитное поле, аналогичное полю прямого тока.

Так как интегралы (5.27)–(5.29) не берутся в элементарных функциях, целесообразно, задавшись конкретными параметрами обмотки (шагом, числом витков и радиусом поперечного сечения), провести численное вычисление компонент вектора напряженности магнитного поля и провести его визуализацию, как это было описано в предыдущем разделе. Так как далее нас будет интересовать форма силовых линий, но не абсолютное значение напряженностей, множитель будем полагать равным единице в (5.27)–(5.29).

Обсудим более подробно особенности вычисления интегралов (5.27)–(5.29). В MATLAB для нахождения значения определенного интеграла от некоторой функции $f(x)$ на интервале $[a, b]$ необходимо выполнить следующую последовательность действий:

1. провести разбиение отрезка $[a, b]$ интегрирования на N отрезков и сохранить координаты их концов в векторе x_i, $i = 1, \ldots, N$;

2. вычислить значения подынтегральной функции $f(x)$ в узловых точках x_i, $i = 1, \ldots, N$: $F_i = f(x_i)$;

3. использовать функцию MATLAB, реализующую метод трапеций:

   `trapz(x,F)`.

Таким образом, для вычисления и визуализации напряженности магнитного поля соленоида следует:

1. Создать m-файлы, возвращающие значения составляющих напряженности магнитного поля вдоль каждой из координатных осей:

```
% листинг файл Bx.m
function z=Bx(x0,y0,z0,a,h,N)
```

```
% функция, возвращающая значение
% проекции напряженности
% магнитного поля на ось OX
phi=0:(50*N):2*pi*N; % вычисление координат
                     % узловых точек
F=f(phi,x0,y0,z0,a,h); % вычисление значений
                       % функции в узловых точках
z=trapz(phi,F); % вычисление значений интеграла

function F=f(psi,x0,y0,z0,a,h)
% описание подынтегральной функции в (5.27)
F=((z0-psi*h/(2*pi))*a.*cos(psi)...
  -h/(2*pi)*(y0-a*sin(psi)))./...
  (x0^2+y0^2+z0^2+a^2+(h/(2*pi)*psi).^2-...
   2*(x0*a*cos(psi)+a*y0*sin(psi)...
   +h/(2*pi)*z0*psi)).^(3/2);

% листинг файл By.m
function z=By(x0,y0,z0,a,h,N)
% функция, возвращающая значение
% проекции напряженности
% магнитного поля на ось OY
phi=0:pi/50:2*pi*N; % вычисление координат
                    % узловых точек
F=f(phi,x0,y0,z0,a,h,I); % вычисление значений
                         % функции в узловых точках
z=trapz(phi,F); % вычисление значений интеграла

function F=f(psi,x0,y0,z0,a,h);
% описание подынтегральной функции в (5.28)
F=((x0-a*cos(psi))*h/(2*pi)...
   +a*sin(psi).*(z0-psi*h/(2*pi)))./...
   (x0^2+y0^2+z0^2+a^2+(h/(2*pi)*psi).^2-...
   2*(x0*a*cos(psi)+a*y0*sin(psi)+...
   h/(2*pi)*z0*psi)).^(3/2);

% листинг файл Bz.m
function z=Bz(x0,y0,z0,a,h,N)
% функция, возвращающая значение
% проекции напряженности магнитного
% поля на ось OZ
phi=0:pi/50:2*pi*N; % вычисление координат
                    % узловых точек
F=f(phi,x0,y0,z0,a,h,I); % вычисление значений
                         % функции в узловых точках
```

```
z=-trapz(phi,F); % вычисление значений интеграла

function F=f(psi,x0,y0,z0,a,h);
% описание подынтегральной функции в (5.29)
F=((y0-a*sin(psi))*a.*sin(psi)+...
  a*cos(psi).*(x0-a*cos(psi)))./...
  (x0^2+y0^2+z0^2+a^2+(h/(2*pi)*psi).^2-...
   2*(x0*a*cos(psi)+a*y0*sin(psi)+...
  h/(2*pi)*z0*psi)).^(3/2);
```

2. Задать координатную сетку (вычислить координаты узлов сетки).

3. Вычислить в каждом узле сетки значение составляющих напряженностей магнитного поля соленоида.

4. Построить векторное поле единичных векторов, касательных к силовым линиям магнитного поля, выполнив последовательность действий, описанную в предыдущем разделе.

Для построения непрерывных силовых линий магнитного поля, как было показано выше, следует вычислить в узлах координатной сетки значения векторного потенциала $\vec{A}(x,y,z)$:

$$\vec{A} = \frac{\mu_0 I}{4\pi} \int \frac{d\vec{l}(\varphi)}{|\vec{R}_0 - \vec{R}(\varphi)|}, \tag{5.31}$$

где $d\vec{l}(\varphi)$, $\vec{R}(\varphi)$ — векторы, определенные согласно (5.21)–(5.22), $\vec{R}_0$ — радиус-вектор точки наблюдения, и построить карту эквипотенциальных поверхностей функции $|\vec{A}(x,y,z)|$.

Подставляя (5.21)-(5.22) в (5.31), получаем выражения для составляющих векторного потенциала магнитного поля соленоида вдоль соответствующих координатных осей:

$$A_x = \frac{\mu_0 I}{4\pi} \int\limits_0^{2\pi N} \frac{-a\sin\varphi}{\left(R_0^2 + a^2 + \left(\frac{h}{2\pi}\varphi\right)^2 - 2\left(x_0 a\cos\varphi + a y_0 \sin\varphi + \frac{h}{2\pi} z_0 \varphi\right)\right)^{1/2}} d\varphi, \tag{5.32}$$

$$A_y = \frac{\mu_0 I}{4\pi} \int\limits_0^{2\pi N} \frac{a\cos\varphi}{\left(R_0^2 + a^2 + \left(\frac{h}{2\pi}\varphi\right)^2 - 2\left(x_0 a\cos\varphi + a y_0 \sin\varphi + \frac{h}{2\pi} z_0 \varphi\right)\right)^{1/2}} d\varphi, \tag{5.33}$$

$$A_z = \frac{\mu_0 I}{4\pi} \int\limits_0^{2\pi N} \frac{\frac{h}{2\pi} z_0}{\left(R_0^2 + a^2 + \left(\frac{h}{2\pi}\varphi\right)^2 - 2\left(x_0 a\cos\varphi + a y_0 \sin\varphi + \frac{h}{2\pi} z_0 \varphi\right)\right)^{1/2}} d\varphi. \tag{5.34}$$

Таким образом, для вычисления и визуализации силовых линий магнитного поля соленоида нужно:

1. Создать m-файлы, возвращающие значения составляющих векторного потенциала магнитного поля вдоль каждой из координатных осей:

```
% листинг файла Ax.m
function z=Ax(x0,y0,z0,a,h,N)
% функция, возвращающая значение
% проекции векторного потенциала
% магнитного поля на ось OX
phi=0:pi/50:2*pi*N; % вычисление координат
                    % узловых точек
F=f(phi,x0,y0,z0,a,h); % вычисление значений функции
                       % в узловых точках
z=trapz(phi,F); % вычисление значений интеграла

function F=f(psi,x0,y0,z0,a,h);
% описание подынтегральной функции в (5.32)
F=-(a*sin(psi))./...
  ((x0-a*cos(psi)).^2+...
  (y0-a*sin(psi)).^2+...
  (z0-h./(2*pi)*psi).^2).^0.5;

% листинг файла Ay.m
function z=Ay(x0,y0,z0,a,h,N)
% функция, возвращающая значение
% проекции векторного потенциала
% магнитного поля на ось OY
phi=0:pi/50:2*pi*N; % вычисление координат
                    % узловых точек
F=f(phi,x0,y0,z0,a,h); % вычисление значений функции
                       % в узловых точках
z=trapz(phi,F); % вычисление значений интеграла

function F=f(psi,x0,y0,z0,a,h);
% описание подынтегральной функции в (5.33)
F=(a*cos(psi))./...
  ((x0-a*cos(psi)).^2+...
  (y0-a*sin(psi)).^2+...
  (z0-h./(2*pi)*psi).^2).^0.5;

% листинг файла Az.m
function z=Az(x0,y0,z0,a,h,N)
% функция, возвращающая значение
% проекции векторного потенциала
% магнитного поля на ось OZ
phi=0:pi/50:2*pi*N; % вычисление координат
                    % узловых точек
```

```
F=f(phi,x0,y0,z0,a,h); % вычисление значений функции
                       % в узловых точках
z=trapz(phi,F); % вычисление значений интеграла

function F=f(psi,x0,y0,z0,a,h);
% описание подынтегральной функции в (5.34)
F=(h/(2*pi))./...
  ((x0-a*cos(psi)).^2+...
  (y0-a*sin(psi)).^2+...
  (z0-h./(2*pi)*psi).^2).^0.5;
```

2. Задать координатную сетку.

3. Вычислить в каждом узле сетки значение проекций векторного потенциала магнитного поля соленоида на соответствующие координатные оси.

4. Построить карту линий уровня функции $|\vec{A}(x, y, z)|$.

Для примера на рис. 5.21, 5.22 представлены проекции силовых линий однослойного соленоида с параметрами $a = 0,5$, $L = 6$, $N = 200$, нижний конец которого расположен в плоскости $Z = 0$, на плоскости $Z = 0$ и $Z = 3$ соответственно. Проекция

Рис. 5.21. Визуализация силовых линий магнитного поля соленоида с постоянным током в плоскости $X = 0$

Рис. 5.22. Визуализация силовых линий магнитного поля соленоида с постоянным током в плоскости $Z = 3$

векторного потенциала на плоскость YOZ вычислялась на квадратной координатной сетке со следующими параметрами: левый нижний угол $(-5, -5)$, правый верхний угол $(5, 5)$, число узлов $N = 17 \times 17$. Проекция векторного потенциала на плоскость $Z = 3$ вычислялась на прямоугольной координатной сетке со следующими параметрами: левый нижний угол $(-5, -5)$, правый верхний угол $(5, 8)$, число узлов $N = 17 \times 17$. На рис. 5.21, 5.22 более толстыми линиями показаны соответствующие сечения соленоида.

Рис. 5.23. Зависимость отношения проекций напряженностей на плоскость $Z = 3$ и ось OZ от расстояния до оси OZ

Для сравнения проекции напряженности магнитного поля на плоскость $Z = 3$ и проекции напряженности магнитного поля на ось OZ вычислено их отношение на различных расстояниях от оси OZ (рис. 5.23). Как очевидно, данная величина есть тангенс угла между силовой линией и осью OZ, который характеризует искривленность силовых линий. Как показывает анализ зависимости, представленной на рис. 5.23, у соленоида с указанными размерами составляющая напряженности магнитного поля, лежащая в плоскости $Z = 3$, то есть достаточно далеко от обоих концов соленоида, оказывается одного порядка с составляющей напряженности, параллельной оси OZ. Следовательно, силовые линии магнитного поля, создаваемые однослойным соленоидом, являются не плоскими, но пространственными кривыми.

***Задача* 5.6**

Используя выражения (5.27)–(5.29)*, исследуйте магнитное поле внутри и снаружи прямого соленоида с постоянным током при различной длине соленоида и шаге намотки.*

1. *Сравните значения контурного интеграла* $4\pi\mu_0 J \oint \vec{B}d\vec{l}$*, вычисляемого по круговым контурам, один из которых находится внутри, а другой — вне соленоида.*

2. *Сравните значения радиальной* B_r *и нормальной (то есть перпендикулярной к радиусу-вектору точки наблюдения)* B_φ *составляющих напряженности магнитного поля.*

3. *Как зависит нормальная составляющая напряженности магнитного поля соленоида* B_φ *от азимутального угла* φ*?*

4. *Вычислите зависимость напряженности поперечного магнитного поля от расстояния до продольной оси соленоида при неизменной длине соленоида и различном шаге намотки. Постройте полученные зависимости и сравните их с аналогичной зависимостью напряженности магнитного поля, создаваемого постоянным прямым током.*

5. *Можно ввести еще один параметр* η*, характеризующий магнитное поле соленоида с постоянным током, — отношение нормальной составляющей напряженности* B_φ *к продольной составляющей напряженности магнитного* B_z *в данной точке. Этот параметр определяет «степень вертикальности» силовой линии магнитного поля, то есть показывает, можно ли пренебречь напряженностью поперечной составляющей магнитного поля по сравнению с напряженностью его продольной составляющей:*

(a) *Вычислите в выбранной точке пространства значения параметра η при различном шаге намотки и фиксированных размерах соленоида. Изобразите данную зависимость графически.*

(b) *Вычислите при выбранном шаге намотки и длине соленоида зависимость параметра η от расстояния до продольной оси соленоида. Изобразите данную зависимость графически.*

(c) *Можно ли считать силовую линию вертикальной? Почему известные опыты с металлическими опилками не позволяют выявить обнаруженную особенность магнитного поля прямого соленоида с постоянным током?*

Для упрощения процедуры расчета и визуализации магнитного поля соленоида с постоянным током автором создан соответствующий `GUI`, представленный на рис. 5.24. Описание соответствующего интерфейса находится в файле (в папке `Glava5`). Для запуска `GUI` следует открыть соответствующую папку, далее выбрать и открыть файл `Solenoid1.fig` или выполнить в командном окне команду:

```
>> Solenoid1
```

После заданий расчетных параметров и нажатия на одну из кнопок интерфейса будет исполняться файл `Solenoind1.m`, листинг которого приведен ниже.

```
% листинг файла Solenoid1.m
function varargout = solenoid1(varargin)
% SOLENOID1 Application M-file for solenoid1.fig

%    FIG = SOLENOID1 launch solenoid1 GUI.
%    SOLENOID1('callback_name', ...) invoke the named callback.
% Last Modified by GUIDE v2.0 17-Jun-2009 10:29:42
% Copyright 2003-2009 S. V. Porshnev

if nargin == 0  % LAUNCH GUI

    fig = openfig(mfilename,'reuse');

    % Generate a structure of handles to pass
    % to callbacks, and store it.
    handles = guihandles(fig);
    guidata(fig, handles);

    if nargout > 0
        varargout{1} = fig;
    end

elseif ischar(varargin{1}) % INVOKE NAMED SUBFUNCTION
                           % OR CALLBACK
```

Рис. 5.24. Зависимость отношения проекций напряженностей на плоскость $Z = 3$ и ось OZ от расстояния до оси OZ

```
    try
        if (nargout)
            % FEVAL switch-yard
            [varargout{1:nargout}] = feval(varargin{:});
        else
            feval(varargin{:}); % FEVAL switchyard
        end
    catch
        disp(lasterr);
    end

end
```

```
%| ABOUT CALLBACKS:
%| GUIDE automatically appends subfunction
%| prototypes to this file, and sets objects'
%| callback properties to call them through the FEVAL
%| switchyard above. This comment describes that mechanism.
%| Each callback subfunction declaration has
%| the following form:
%| <SUBFUNCTION_NAME>(H, EVENTDATA, HANDLES, VARARGIN)
%|
%| The subfunction name is composed using
%| the object's Tag and the
%| callback type separated by '_', e.g. 'slider2_Callback',
%| 'figure1_CloseRequestFcn', 'axis1_ButtondownFcn'.
%|
%| H is the callback object's handle (obtained using GCBO).
%|
%| EVENTDATA is empty, but reserved for future use.
%|
%| HANDLES is a structure containing handles of components
%| in GUI using tags as fieldnames, e.g. handles.
%| figure1, handles.slider2. This structure is created
%| at GUI startup using GUIHANDLES and stored in
%| the figure's application data using GUIDATA.
%| A copy of the structure is passed to each callback.
%| You can store additional information int his structure
%|  at GUI startup, and you can change the structure
%| during callbacks.  Call guidata(h, handles) after
%| copy to replace the stored original so that subsequent
%| callbacks see changing your the updates.
%| Type "help guihandles" and  "help guidata"
%| for more information.
%|
%| VARARGIN contains any extra arguments you have
%| passed to thecal lback. Specify the extra arguments
%| by editing the callback property in the inspector.
%| By default, GUIDE sets the property to:
%| <MFILENAME>('<SUBFUNCTION_NAME>', gcbo, [ ], guidata(gcbo))
%| Add any extra arguments after the last argument,
%|  before the final closing parenthesis.
% -------------------------------------------------------

function z=Bx(x0,y0,z0,a,h,N,I)
f=inline('I*((z0-psi*h/(2*pi))*a.*cos(psi)-...
          h/(2*pi)*(y0-a*sin(psi)))...
          /(x0^2+y0^2+z0^2+a^2+(h/(2*pi)*psi).^2-...
```

```
          2*(x0*a*cos(psi)+a*y0*sin(psi)+...
          h/(2*pi)*z0*psi)).^(3/2)',...
          'psi','x0','y0','z0','a','h','I');
phi=0:pi/50:2*pi*N;
F=f(phi,x0,y0,z0,a,h,I);
z=trapz(phi,F);
function mutual_exclude(off)
set(off,'Value',0)

% ---------------------------------------------------------
function z=By(x0,y0,z0,a,h,N,I)
f=inline('I*((x0-a*cos(psi))*h/(2*pi)+...
              a*sin(psi).*(z0-psi*h/(2*pi))).... 
              /(x0^2+y0^2+z0^2+a^2+(h/(2*pi)*psi).^2-...
              2*(x0*a*cos(psi)+a*y0*sin(psi)+...
              h/(2*pi)*z0*psi)).^(3/2)',...
              'psi','x0','y0','z0','a','h','I');
phi=0:pi/50:2*pi*N;
F=f(phi,x0,y0,z0,a,h,I);
z=trapz(phi,F);

% ---------------------------------------------------------
function z=Bz(x0,y0,z0,a,h,N,I)
f=inline('I*((y0-a*sin(psi))*a.*sin(psi)+...
              a*cos(psi).*(x0-a*cos(psi))). ...
              /(x0^2+y0^2+z0^2+a^2+(h/(2*pi)*psi).^2-...
               2*(x0*a*cos(psi)+a*y0*sin(psi)+...
               h/(2*pi)*z0*psi)).^(3/2)',...
              'psi','x0','y0','z0','a','h','I');
phi=0:pi/50:2*pi*N;
F=f(phi,x0,y0,z0,a,h,I);
z=trapz(phi,F);

% ---------------------------------------------------------
function z=Ax(x0,y0,z0,a,h,N,I)
f=inline('-I*(a*sin(psi))./((x0-a*cos(psi)).^+...
               (y0-a*sin(psi)).^2+...
               (z0-h./(2*pi)*psi).^2).^0.5',...
               'psi','x0','y0','z0','a','h','I');
phi=0:pi/50:2*pi*N;
F=f(phi,x0,y0,z0,a,h,I);
z=trapz(phi,F);

% ---------------------------------------------------------
function z=Ay(x0,y0,z0,a,h,N,I)
```

```
f=inline('I*(a*cos(psi))./((x0-a*cos(psi)).^2+...
               (y0-a*sin(psi)).^2+...
               (z0-h./(2*pi)*psi).^2).^0.5',...
               'psi','x0','y0','z0','a','h','I');
phi=0:pi/50:2*pi*N;
F=f(phi,x0,y0,z0,a,h,I);
z=trapz(phi,F);

% -------------------------------------------------------
function z=Az(x0,y0,z0,a,h,N,I)
f=inline('(I*h/(2*pi))./((x0-a*cos(psi)).^2+...
               (y0-a*sin(psi)).^2.+...
               (z0-h./(2*pi)*psi).^2).^0.5',...
               'psi','x0','y0','z0','a','h','I');
phi=0:pi/50:2*pi*N;
F=f(phi,x0,y0,z0,a,h,I);
z=trapz(phi,F);
% -------------------------------------------------------
function varargout = edit1_Callback(h,...
                                    eventdata,...
                                    handles,...
                                    varargin)

% -------------------------------------------------------
function read_data(handles)
global Radius...
       Solenoid_Length...
       N_Coil I...
       Xmin Xmax...
       Ymin Ymax...
       Zmin Zmax...
       N_Knot...
       X0 Y0 Z0...
       Mod_R Phi...
       dR...
       N_Step
Radius = str2double(get(handles.edit1,'String'));

Solenoid_Length = str2double(get(handles.edit2,'String'));

N_Coil = str2double(get(handles.edit3,'String'));

I = str2double(get(handles.edit4,'String'));
Xmin = str2double(get(handles.edit5,'String'));
Xmax = str2double(get(handles.edit6,'String'));
```

```
Ymin = str2double(get(handles.edit7,'String'));
Ymax = str2double(get(handles.edit8,'String'));

Zmin = str2double(get(handles.edit9,'String'));
Zmax = str2double(get(handles.edit10,'String'));

N_Knot = str2double(get(handles.edit11,'String'));

X0 = str2double(get(handles.edit12,'String'));
Y0 = str2double(get(handles.edit13,'String'));
Z0 = str2double(get(handles.edit14,'String'));

Mod_R = str2double(get(handles.edit15,'String'));

Phi = str2double(get(handles.edit16,'String'));

dR = str2double(get(handles.edit17,'String'));

N_Step = str2double(get(handles.edit18,'String'));

% -------------------------------------------------------
function varargout = edit2_Callback(h,...
                                    eventdata,...
                                    handles,...
                                    varargin)

% -------------------------------------------------------
function varargout = edit3_Callback(h,...
                                    eventdata,...
                                    handles,...
                                    varargin)
% -------------------------------------------------------

function varargout = edit4_Callback(h,...
                                    eventdata,...
                                    handles,...
                                    varargin)
% -------------------------------------------------------

function varargout = edit5_Callback(h,...
                                    eventdata,...
                                    handles,...
                                    varargin)
% -------------------------------------------------------
```

```
function varargout = edit6_Callback(h,...
                                    eventdata,...
                                    handles,...
                                    varargin)
% --------------------------------------------------------

function varargout = edit7_Callback(h,...
                                    eventdata,...
                                    handles,...
                                    varargin)
% --------------------------------------------------------

function varargout = edit8_Callback(h,...
                                    eventdata,...
                                    handles,...
                                    varargin)
% --------------------------------------------------------

function varargout = edit9_Callback(h,...
                                    eventdata,...
                                    handles,...
                                    varargin)
% --------------------------------------------------------

function varargout = edit10_Callback(h,...
                                    eventdata,...
                                    handles,...
                                    varargin)

% --------------------------------------------------------
function varargout = edit11_Callback(h,...
                                    eventdata,...
                                    handles,...
                                    varargin)

% --------------------------------------------------------
function varargout = edit12_Callback(h,...
                                    eventdata,...
                                    handles,...
                                    varargin)

% --------------------------------------------------------
function varargout = figure1_ResizeFcn(h,...
                                    eventdata,...
```

```
                                    handles,...
                                    varargin)
% -------------------------------------------------------
function varargout = edit13_Callback(h,...
                                    eventdata,...
                                    handles,...
                                    varargin)

% -------------------------------------------------------
function varargout = edit14_Callback(h,...
                                    eventdata,...
                                    handles,...
                                    varargin)

% -------------------------------------------------------
function varargout = radiobutton1_Callback(h,...
                                    eventdata,...
                                    handles,...
                                    varargin)
off = [handles.radiobutton2,...
       handles.radiobutton3,...
       handles.radiobutton4];
mutual_exclude(off)

% -------------------------------------------------------
function varargout = radiobutton2_Callback(h,...
                                    eventdata,...
                                    handles,...
                                    varargin)
off = [handles.radiobutton2,...
       handles.radiobutton3,
       handles.radiobutton4]
mutual_exclude(off)

% -------------------------------------------------------
function varargout = radiobutton3_Callback(h,...
                                    eventdata,...
                                    handles,...
                                    varargin)
off = [handles.radiobutton1,...
       handles.radiobutton2,...
       handles.radiobutton4];
mutual_exclude(off)

% -------------------------------------------------------
```

```
function varargout = radiobutton4_Callback(h,...
                                    eventdata,...
                                    handles,...
                                    varargin)
off = [handles.radiobutton1,...
       handles.radiobutton2,...
       handles.radiobutton3];
mutual_exclude(off)

% -------------------------------------------------------
% ПОСТРОЕНИЕ СЕТКИ
function varargout = pushbutton1_Callback(h,...
                                    eventdata,...
                                    handles,...
                                    varargin)
  global Radius Solenoid_Length N_Coil I ...
             Xmin Xmax...
             Ymin Ymax...
             Zmin Zmax...
             N_Knot...
             X0 Y0 Z0...
             Mod_R...
             Phi dR...
             N_Step
  read_data(handles);
  i=1:N_Knot+1;
  if get(handles.radiobutton1,'Value')==1
  set(handles.text34,'String',...
        strcat('КООРДИНАТНАЯ СЕТКА В ПЛОСКОСТИ X0=',...
                mat2str(X0)));

    y(i)=Ymin+(Ymax-Ymin)/N_Knot*(i-1);
    z(i)=Zmin+(Zmax-Zmin)/N_Knot*(i-1);

    X1=-Radius/2;Y1=0;
    X2=Radius/2;Y2=0;
    X3=Radius/2;Y3=Solenoid_Length;
    X4=-Radius/2;Y4=Solenoid_Length;

    h_Coil=plot([X1,X2,X3,X4,X1],...
                [Y1,Y2,Y3,Y4,Y1]);
    hold on
    for i=1:N_Knot+1
      for j=1:N_Kn ot+1
        h_Knot=line([y(i),y(i)],...
```

```
                    [z(j),z(j)],...
                    'Color','Red',...
                    'Marker','o',...
                    'MarkerSize',3);
      end
    end;
    hold off
  end
  if get(handles.radiobutton2,'Value')==1
    set(handles.text34,'String',...
        strcat('КООРДИНАТНАЯ СЕТКА В ПЛОСКОСТИ Y0=',...
                mat2str(Y0)));

    x(i)=Xmin+(Xmax-Xmin)/N_Knot*(i-1);
    z(i)=Zmin+(Zmax-Zmin)/N_Knot*(i-1);

    X1=-Radius/2;
    Y1=0;

    X2=Radius/2;
    Y2=0;

    X3=Radius/2;
    Y3=Solenoid_Length;

    X4=-Radius/2;
    Y4=Solenoid_Length;

    h_coil=plot([X1,X2,X3,X4,X1],...
                [Y1,Y2,Y3,Y4,Y1]);
    hold on
    for i=1:N_Knot+1
      for j=1:N_Knot+1
        h_Knot=line([x(i),x(i)],...
                    [z(j),z(j)],...
                    'Color','Red',...
                    'Marker','o',...
                    'MarkerSize',3);
      end
    end;
    hold off
  end
  if get(handles.radiobutton3,'Value')==1
  set(handles.text34,'String',...
        strcat('КООРДИНАТНАЯ СЕТКА В ПЛОСКОСТИ Z0=',...
```

```
              mat2str(Z0)));
  x(i)=Xmin+(Xmax-Xmin)/N_Knot*(i-1);
  y(i)=Ymin+(Ymax-Ymin)/N_Knot*(i-1);
  alpha=0:pi/100:2*pi;
  h_Coil=plot(Radius*cos(alpha),Radius*sin(alpha));
  hold on
   for i=1:N_Knot+1
     for j=1:N_Knot+1
       h_Knot=line([x(i),x(i)],...
                   [y(j),y(j)],...
                   'Color','Red','...
                   Marker','o',...
                   'MarkerSize',3);
     end
   end;
   hold off
 end
 if get(handles.radiobutton4,'Value')==1
   set(handles.text34,'String',...
       'ПРОСТРАНСТВЕННАЯ КООРДИНАТНАЯ СЕТКА');
   h=Solenoid_Length/N_Coil;
   phi=0:pi/50:N_Coil*2*pi;

   x1=Radius*sin(phi);
   y1=Radius*cos(phi);
   z1=phi/(2*pi)*h-Solenoid_Length/2;

   h_Coil=plot3(x1,y1,z1);
   view([30,30])

   x(i)=Xmin+(Xmax-Xmin)/N_Knot*(i-1);
   y(i)=Ymin+(Ymax-Ymin)/N_Knot*(i-1);
   z(i)=Zmin+(Zmax-Zmin)/N_Knot*(i-1);
   hold on

   for i=1:N_Knot+1
     for j=1:N_Knot+1
       for k=1:N_Knot+1
         h_Knot=line([x(i),x(i)],...
                     [y(j),y(j)],...
                     [z(k),z(k)],...
                     'Color','Red',...
                     'Marker','o',...
                     'MarkerSize',3);
       end
```

```
      end
    end;
    axis square; grid on
    hold off
  end;

% -------------------------------------------------------
function varargout = pushbutton2_Callback(h,...
                                  eventdata,...
                                  handles,...
                                  varargin)
set(handles.pushbutton3,'Enable','off');
global Radius Solenoid_Length N_Coil I...
            Xmin Xmax...
            Ymin Ymax...
            Zmin Zmax...
            N_Knot...
            X0 Y0 Z0...
            Mod_R Phi dR N_Step ...
            y1 z1 a1...
            a2 x2 z2...
            x3 y3 a3...
            x4 y4 z4 a4...
            ax ay az

  read_data(handles);
  h=Solenoid_Length/N_Coil;
  i=1:N_Knot;

  % ВЫЧИСЛЕНИЕ ПРОЕКЦИИ
  % ВЕКТОРА-ПОТЕНЦИАЛА НА ПЛОСКОСТЬ YOZ
  if get(handles.radiobutton1,'Value')==1

    y1(i)=Ymin+(Ymax-Ymin)/(N_Knot-1)*(i-1);
    z1(i)=Zmin+(Zmax-Zmin)/(N_Knot-1)*(i-1);

    for k=1:N_Knot
      for j=1:N_Knot
        ax=Ax(X0,y1(j),z1(k),Radius,h,N_Coil,I);
        ay=Ay(X0,y1(j),z1(k),Radius,h,N_Coil,I);
        az=Az(X0,y1(j),z1(k),Radius,h,N_Coil,I);
        a1(k,j)=(ax.^2+ay.^2+az.^2).^0.5;
      end
    end;
  end;
```

```
% ВЫЧИСЛЕНИЕ ПРОЕКЦИИ
% ВЕКТОРА-ПОТЕНЦИАЛА НА ПЛОСКОСТЬ XOZ
  if get(handles.radiobutton2,'Value')==1

    x2(i)=Xmin+(Xmax-Xmin)/(N_Knot-1)*(i-1);
    z2(i)=Zmin+(Zmax-Zmin)/(N_Knot-1)*(i-1);

    for k=1:N_Knot
      for i=1:N_Knot
        ax=Ax(x2(i),Y0,z2(k),Radius,h,N_Coil,I);
        ay=Ay(x2(i),Y0,z2(k),Radius,h,N_Coil,I);
        az=Az(x2(i),Y0,z2(k),Radius,h,N_Coil,I);
        a2(k,i)=(ax.^2+ay.^2+az.^2).^0.5;
      end
    end;
  end;

  % ВЫЧИСЛЕНИЕ ПРОЕКЦИИ
  % ВЕКТОРА-ПОТЕНЦИАЛА НА ПЛОСКОСТЬ XOY
  if get(handles.radiobutton3,'Value')==1

    x3(i)=Xmin+(Xmax-Xmin)/(N_Knot-1)*(i-1);
    y3(i)=Ymin+(Ymax-Ymin)/(N_Knot-1)*(i-1);

    for j=1:N_Knot
      for i=1:N_Knot
        ax=Ax(x3(i),y3(j),Z0,Radius,h,N_Coil,I);
        ay=Ay(x3(i),y3(j),Z0,Radius,h,N_Coil,I);
        az=Az(x3(i),y3(j),Z0,Radius,h,N_Coil,I);
        a3(i,j)=(ax.^2+ay.^2+az.^2).^0.5;
      end
    end;
  end;

 % ВЫЧИСЛЕНИЕ ВЕКТОРА-ПОТЕНЦИАЛА
 % В УЗЛАХ ТРЕХМЕРНОЙ СЕТКИ

  if get(handles.radiobutton4,'Value')==1
    x4(i)=Xmin+(Xmax-Xmin)/(N_Knot-1)*(i-1);
    y4(i)=Ymin+(Ymax-Ymin)/(N_Knot-1)*(i-1);
    z4(i)=Zmin+(Zmax-Zmin)/(N_Knot-1)*(i-1);

    for i=1:N_Knot
      for j=1:N_Knot
```

```
        for k=1:N_Knot
          ax(i,j,k)=Ax(x4(i),y4(j),z4(k),Radius,h,N_Coil,I);
          ay(i,j,k)=Ay(x4(i),y4(j),z4(k),Radius,h,N_Coil,I);
          az(i,j,k)=Az(x4(i),y4(j),z4(k),Radius,h,N_Coil,I);
          a4(i,j,k)=(ax(i,j,k).^2+ay(i,j,k).^2+az(i,j,k).^2).^0.5;
        end
      end
    end;
 end;
set(handles.pushbutton3,'Enable','on');

% -------------------------------------------------------
function varargout = pushbutton3_Callback(h,...
                                  eventdata,...
                                  handles,...
                                  varargin)

global Radius Solenoid_Length N_Coil I...
           Xmin Xmax...
           Ymin Ymax...
           Zmin Zmax...
           N_Knot...
           X0 Y0 Z0...
           Mod_R Phi dR N_Step...
           y1 z1 a1...
           x2 z2 a2...
           x3 y3 a3...
           x4 y4 z4 a4

% НАПРЯЖЕННОСТЬ В ПЛОСКОСТИ YOZ
  if get(handles.radiobutton1,'Value')==1
    if ~isempty(y1)
    set(handles.text34,'String',...
        strcat('НАПРЯЖЕННОСТЬ МАГНИТНОГО...
                ПОЛЯ В ПЛОСКОСТИ X0=',...
                mat2str(X0)));
    contour(y1,z1,a1,17);
    hold on
    X1=-Radius/2;Y1=0;
    X2=Radius/2;Y2=0;
    X3=Radius/2;Y3=Solenoid_Length;
    X4=-Radius/2;Y4=Solenoid_Length;
    line([X1,X2,X3,X4,X1],[Y1,Y2,Y3,Y4,Y1])
    hold off
    end;
```

```
  end;

  % НАПРЯЖЕННОСТЬ В ПЛОСКОСТИ XOZ
  if get(handles.radiobutton2,'Value')==1
    if ~isempty(x2) %x2~=[]
    set(handles.text34,'String',strcat('НАПРЯЖЕННОСТЬ...
                                       МАГНИТНОГО ПОЛЯ...
                                       В ПЛОСКОСТИ Y0=',...
                                       mat2str(Y0)));
    contour(x2,z2,a2,17);
    hold on

    X1=-Radius/2;Y1=0;
    X2=Radius/2;Y2=0;
    X3=Radius/2;Y3=Solenoid_Length;
    X4=-Radius/2;Y4=Solenoid_Length;

    line([X1,X2,X3,X4,X1],...
         [Y1,Y2,Y3,Y4,Y1])
    hold off
    end;
  end;

% НАПРЯЖЕННОСТЬ В ПЛОСКОСТИ XOY
  if get(handles.radiobutton3,'Value')==1
    if ~isempty(x3) % x3~=[]
    set(handles.text34,'String',strcat('НАПРЯЖЕННОСТЬ...
                                       МАГНИТНОГО ПОЛЯ...
                                       В ПЛОСКОСТИ Z0=',...
                                       mat2str(X0)));
    contour(x3,y3,a3,17);
    hold on
    alpha=0:pi/100:2*pi;
    plot(Radius*cos(alpha),Radius*sin(alpha))
    hold off
    end;
  end;
  % НАПРЯЖЕННОСТЬ В ПРОСТРАНСТВЕ
  if get(handles.radiobutton4,'Value')==1
    if ~isempty(x4) % x4~=[]

      hsurfaces = slice(x4,y4,z4,a4,...
                        Xmax,Ymax,Ymin);
      set(hsurfaces,...
          'FaceColor',...
```

```
          'interp',...
          'EdgeColor',...
          'none')
     hcont =...
          contourslice(x4,y4,z4,a4,...
                       Xmax,Ymax,Ymin);
          set(hcont,...
              'EdgeColor',...
              [.7,.7,.7],...
              'LineWidth',.5)
     [sx,sy,sz] = meshgrid(Xmin:0.1:Xmax,...
                           Ymin:0.1:Ymax,...
                           Zmin);
hlines = streamline(x4,y4,z4,...
                    ax,ay,az,...
                    sx,sy,sz);
set(hlines,...
   'LineWidth',2,...
   'Color','r')
view(3)
daspect([2,2,1])
axis tight
end;
end;
% ---------------------------------------------------------
function varargout = pushbutton4_Callback(,...
                                  eventdata,...
                                  handles,...
                                  varargin)

helpdlg('ПРОГРАММА ПРЕДНАЗНАЧЕНА ДЛЯ РАСЧЕТА И...
              ВИЗУАЛИЗАЦИИ НАПРЯЖЕННОСТИ...
              МАГНИТНОГО ПОЛЯ КАТУШКИ...
              С ПОСТОЯННЫМ ТОКОМ...
              В ОТЛИЧИЕ ОТ ТРАДИЦИОННОГО...
              ПОДХОДА, В КОТОРОМ СОЛЕНОИД...
              ПРЕДСТАВЛЯЕТСЯ НАБОРОМ...
              КРУГОВЫХ ТОКОВ, А ПОТОМУ...
              СОСТАВЛЯЮЩАЯ МАГНИТНОГО ПОЛЯ...
              В ПЛОСКОСТИ ПЕРПЕНДИКУЛЯРНОЙ...
              ОСИ OZ ОКАЗЫВАЕТСЯ РАВНОЙ...
              НУЛЮ, В ПРОГРАММЕ ИСПОЛЬЗУЮТСЯ...
              РАСЧЕТНЫЕ ФОРМУЛЫ, ПОЛУЧЕННЫЕ...
              ПРИ РАССМОТРЕНИИ СОЛЕНОИДА...
              В ВИДЕ СПИРАЛИ...
```

```
              (СМ.: ПОРШНЕВ С.В., ХАРИТОНОВ В.И. ...
              ОСОБЕННОСТИ МАГНИТНОГО ПОЛЯ...
              СОЛЕНОИДА С ПОСТОЯННЫМ ТОКОМ // ...
              ЭЛЕКТРИЧЕСТВО. 1998. Т. 6. С. 71-75)...
              ПРОГРАММА ПОЗВОЛЯЕТ: ...
              1) ВЫЧИСЛЯТЬ ПРОЕКЦИИ НАПРЯЖЕННОСТИ...
               МАГНИТНОГО ПОЛЯ НА СООТВЕТСТВУЮЩИЕ...
               КООРДИНАТНЫЕ ОСИ И СТРОИТЬ...
              НОРМИРОВАННЫЕ ВЕКТОРНЫЕ ПОЛЯ, ДАЮЩИЕ...
              ПРЕДСТАВЛЕНИЕ О НАПРЯЖЕННОСТИ...
              И ГЕОМЕТРИИ СИЛОВЫХ ЛИНИЙ...
              МАГНИТНОГО ПОЛЯ;...
              2) ВЫЧИСЛЯТЬ И ПРЕДСТАВЛЯТЬ ЗАВИСИМОСТИ...
              МОДУЛЯ ПРОЕКЦИИ НАПРЯЖЕННОСТИ...
              МАГНИТНОГО ПОЛЯ НА ПЛОСКОСТИ,...
              ПЕРПЕНДИКУЛЯРНЫЕ ОСИ OZ, ОТ РАССТОЯНИЯ...
             ДО ОСИ OZ;...
             3) ВЫЧИСЛЯТЬ И ПРЕДСТАВЛЯТЬ НА ГРАФИКЕ...
             МОДУЛЬ ОТНОШЕНИЯ НАПРЯЖЕННОСТЕЙ...
             ПРОЕКЦИИ МАГНИТНОГО ПОЛЯ...
             НА ПЛОСКОСТЬ, ПЕРПЕНДИКУЛЯРНУЮ...
            ОСИ OZ, К ПРОЕКЦИИ НАПРЯЖЕННОСТИ...
            НА ОСЬ OZ, ХАРАКТЕРИЗУЮЩЕГО СТЕПЕНЬ...
            НЕПАРАЛЕЛЬНОСТИ СИЛОВОЙ ЛИНИИ...
            ОСИ OZ.','КРАТКАЯ ИНФОРМАЦИЯ О ПРОГРАММЕ');

% ---------------------------------------------------------------------
function varargout = pushbutton5_Callback(h,...
                                          eventdata,...
                                          handles,...
                                          varargin)
close;

% ---------------------------------------------------------------------
function varargout = edit15_Callback(h,...
                                     eventdata,...
                                     handles,...
                                     varargin)

 % ---------------------------------------------------------------------
function varargout = edit16_Callback(h,...
                                     eventdata,...
                                     handles,...
                                     varargin)
```

```
% ---------------------------------------------------------------------
function varargout = edit17_Callback(h,...
                                     eventdata,...
                                     handles,...
                                     varargin)

% ---------------------------------------------------------------------
function varargout = edit18_Callback(h,...
                                     eventdata,...
                                     handles,...
                                     varargin)

% ---------------------------------------------------------------------
function varargout = pushbutton7_Callback(h,...
                                     eventdata,...
                                     handles,...
                                     varargin)
   global   Radius...
            Solenoid_Length...
            N_Coil I...
            Xmin Xmax...
            Ymin Ymax...
            Zmin Zmax ...
            N_Knot...
            X0 Y0 Z0...
            Mod_R...
            Phi...
            dR...
            N_Step...
            R...
            Mh...
            Mode

read_data(handles);
Mode=1;
set(handles.pushbutton9,'Enable','off');
set(handles.pushbutton10,'Enable','off');

X1=Mod_R*cos(Phi);Y1=Mod_R*sin(Phi);
dx=dR*cos(Phi);dy=dR*sin(Phi);h=Solenoid_Length/N_Coil;

for i=1:N_Step
  x1(i)=X1+dR*cos(Phi)*(i-1);
  y1(i)=Y1+dR*sin(Phi)*(i-1);
  Hx(i)=Bx(x1(i),y1(i),Z0,Radius,h,N_Coil,I);
```

```
  Hy(i)=By(x1(i),y1(i),Z0,Radius,h,N_Coil,I);
end;
R=(x1.^2+y1.^2).^0.5;
Mh=(Hx.^2+Hy.^2).^0.5;
Mh=Mh/Mh(1);
set(handles.pushbutton9,'Enable','on');
set(handles.pushbutton10,'Enable','on');

% ---------------------------------------------------------------------
function varargout = pushbutton8_Callback(h,...
                                    eventdata,...
                                    handles,...
                                    varargin)
   global   Radius...
            Solenoid_Length...
            N_Coil I...
            Xmin Xmax...
            Ymin Ymax...
            Zmin Zmax ...
            N_Knot...
            X0 Y0 Z0...
            Mod_R...
            Phi...
            dR...
            N_Step...
            R...
            Mh...
            Mode

read_data(handles);
Mode=2
set(handles.pushbutton9,'Enable','off');
set(handles.pushbutton10,'Enable','off');

X1=Mod_R*cos(Phi);Y1=Mod_R*sin(Phi);
dx=dR*cos(Phi);dy=dR*sin(Phi);h=Solenoid_Length/N_Coil;

for i=1:N_Step
  x1(i)=X1+dR*cos(Phi)*(i-1);
  y1(i)=Y1+dR*sin(Phi)*(i-1);
  Hx(i)=Bx(x1(i),y1(i),Z0,Radius,h,N_Coil,I);
  Hy(i)=By(x1(i),y1(i),Z0,Radius,h,N_Coil,I);
  Hz(i)=Bz(x1(i),y1(i),Z0,Radius,h,N_Coil,I);
end;
R=(x1.^2+y1.^2).^0.5;
```

```
Mh=(Hx.^2+Hy.^2).^0.5;
for i=1:N_Step
  Mh(i)=Mh(i)/abs(Hz(i));
end;
set(handles.pushbutton9,'Enable','on');
set(handles.pushbutton10,'Enable','on');

% ---------------------------------------------------------------------
function varargout = pushbutton9_Callback(h,...
                                    eventdata,...
                                    handles,...
                                    varargin)
 global   Radius...
            Solenoid_Length...
            N_Coil I...
            Xmin Xmax...
            Ymin Ymax...
            Zmin Zmax ...
            N_Knot...
            X0 Y0 Z0...
            Mod_R...
            Phi...
            dR...
            N_Step...
            R...
            Mh...
            Mode
if Mode==1
  set(handles.text34,'String',strcat('МОДУЛЬ ПРОЕКЦИИ...
                                     НАПРЯЖЕННОСТИ НА...
                                     ПЛОСКОСТЬ Z=',mat2str(Z0),...
                                     'НОРМИРОВАННЫЙ НА...
                                     ЗНАЧЕНИЕ НАПРЯЖЕННОСТИ...
                                     В ПЕРВОЙ ТОЧКЕ'));
end;
if Mode==2
  set(handles.text34,'String',strcat('АБСОЛЮТНОЕ ЗНАЧЕНИЕ...
                                    ОТНОШЕНИЯ ПРОЕКЦИИ...
                                    НАПРЯЖЕННОСТИ НА...
                                    ПЛОСКОСТЬ Z=',mat2str(Z0),...
                                    ' К ПРОЕКЦИИ...
                                    НАПРЯЖЕННОСТИ НА ОСЬ OZ'));
end;
plot(R,Mh);
```

```
% ---------------------------------------------------------------------
function varargout = pushbutton10_Callback(h,...
                                      eventdata,...
                                      handles,...
                                      varargin)
 global   Radius...
            Solenoid_Length...
            N_Coil I...
            Xmin Xmax...
            Ymin Ymax...
            Zmin Zmax...
            N_Knot...
            X0 Y0 Z0...
            Mod_R...
            Phi...
            dR...
            N_Step...
            R...
            Mh...
            Mode
hold on
plot(R,Mh);
hold off
```

5.4. МАГНИТНОЕ ПОЛЕ ТОРОИДАЛЬНОЙ ОБМОТКИ С ПОСТОЯННЫМ ТОКОМ

Результаты данного раздела впервые получены автором совместно с В. И. Харитоновым в [19].

Для нахождения магнитного поля тороидальной обмотки с постоянным током (рис. 5.25) используют уравнение Максвелла:

$$rot\vec{B} = \vec{j},$$

имеющее в интегральной форме следующий вид [4]:

$$\oint\limits_L \vec{B}d\vec{l} = \sum_{k=1}^{n} (\pm I_k), \qquad (5.35)$$

где L — контур, охватывающий ток, I_k — сила тока, охватываемого контуром.

Рис. 5.25. Тороидальная обмотка

Рис. 5.26. Традиционно используемая модель тороидальной обмотки с постоянным током

Следующий шаг в нахождении магнитного поля состоит в замене обмотки тора с протекающим по ней током поверхностным током с постоянной линейной плотностью (рис. 5.26).

При использовании данной модели линии тока предполагаются расположенными в меридиональных плоскостях, проходящих через ось симметрии тора, то есть набором колец с током. Циркуляция магнитного поля тора $\vec{B}$ в (5.25), зависящего из соображений симметрии только от расстояния до центра тора, равна $2\pi RB$. Полный ток, пронизывающий площадь, ограниченную этой окружностью, равен $N \cdot I$, где N — число витков тороидальной обмотки. Откуда напряженность магнитного поля $\vec{B}$ внутри тора равна

$$B = \frac{\mu_0}{2\pi R} N \cdot I.$$

Рассмотрение контура, представляющего собой окружность, радиусом, большим радиуса тора, показывает, что полный ток, протекающий через данный контур, равен нулю, так как ток в каждом витке катушки пересекает ее дважды в противоположных направлениях. Отсюда с неизбежностью следует вывод о равенстве нулю напряженности магнитного поля вне тора. Однако количественное обоснование принятого выше допущения о возможности замены тороидальной обмотки поверхностно распределенным током либо не приводится, либо утверждается, что составляющей тока вдоль тора при достаточно плотной намотке можно пренебречь [5, 16]. Справедливость данных подходов можно поставить под сомнение, так как они противоречат закону сохранения заряда. Действительно, при протекании тока через тор и подводящие провода количество электричества, проходящего в единицу времени через выбранное сечение провода 1, должно равняться количеству электричества, проходящего через выбранное поперечное сечение провода 2 (рис. 5.25). Следовательно, количество электричества, перемещаемое в единицу времени вдоль окружности радиуса R тора одинаково при любой крутизне намотки. Так как перемещение заряда вдоль оси тора есть продольный ток, величина которого равна I, то тор должен обладать внешним магнитным полем, аналогичным магнитному полю витка с током.

Для подтверждения проведенного качественного доказательства существования внешнего магнитного поля тора проведем прямые вычисления напряженности магнитного поля на основе закона Био–Савара–Лапласа [4]:

$$d\vec{B} = \frac{\mu_0 I}{4\pi} \frac{\left[d\vec{l} \times \vec{r}\right]}{r^3}, \tag{5.36}$$

где $Id\vec{l}$ — элемент тока, $\vec{r}$ — радиус-вектор от элемента тока в точку наблюдения, и принципа суперпозиции — интегрированием (5.29) по всем элементам тока.

Для однозначного описания положения элементов намотки введем два радиуса-вектора — радиус-вектор $\vec{R}_T$, описывающий положение центра данного сечения тора меридиональной плоскостью, с началом в точке 0, и радиус-вектор $\vec{a}$, описывающий положение элемента спирали в меридиональной плоскости (рис. 5.27). Выберем систему координат XYZ так, что при нулевом угле поворота вектор $\vec{R}_T$ имеет координаты $(\vec{R}_T, 0, 0)$, вектор $\vec{a}$ — $(a, 0, 0)$. Положение каждой точки тороидальной намотки в выбранной системе координат будем описывать радиусом-вектором $\vec{r}$, который является суммой векторов $\vec{R}_T$, $\vec{a}$:

$$\vec{r} = \vec{R}_T + \vec{a}. \qquad (5.37)$$

Рис. 5.27. Традиционно используемая модель тороидальной обмотки с постоянным током

Очевидно, что при постоянном угле намотки положение любой точки спирали будет определяться единственным параметром — углом поворота φ вектора $\vec{R}_T$ в плоскости XOY. Найдем связь угла поворота вектора $\vec{a}$ в меридиональной плоскости φ_1 с углом поворота φ вектора $\vec{R}_T$ в плоскости XOY, для чего рассмотрим плоскостную развертку одного витка тороидальной намотки, представляющую прямоугольный треугольник, длины катетов которого равны длинам дуг окружности радиуса R в плоскости XOY и окружности радиуса a в меридиональной плоскости (рис. 5.28). При N-витковой намотке они равны $R_T \cdot 2\pi N$ и 2π, соответственно, поэтому угол намотки β равен

$$\beta = \arctan \frac{aN}{R_T}. \qquad (5.38)$$

Рис. 5.28. Традиционно используемая модель тороидальной обмотки с постоянным током

Рассмотрим развертку спирали при произвольном угле поворота φ, представляющую, как и для одного витка, прямоугольный треугольник. В этом случае длины соответствующих катетов равны $R_T \cdot \varphi$, $a \cdot \varphi_1$. Используя известные формулы, связывающие катеты прямоугольного треугольника через угол между катетом и гипотенузой, получим

$$R_T \cdot \varphi \cdot \tan\beta = a \cdot \varphi_1. \qquad (5.39)$$

Подставляя выражение для угла β из (5.38) в (5.39), окончательно найдем

$$\varphi = N\varphi_1. \qquad (5.40)$$

Учитывая (5.37), (5.40), запишем зависимость координат радиуса-вектора $\vec{r}$ от угла поворота φ в выбранной системе координат:

$$\vec{r} = ((R_T + a\cos N\varphi)\cos\varphi, (R_T + a\cos N\varphi)\sin\varphi, a\sin N\varphi). \qquad (5.41)$$

Элемент длины тороидальной обмотки $d\vec{l}$ выражается через вектор $\vec{t}$, касательный к обмотке в данной точке, и угол поворота $d\varphi$

$$d\vec{l} = \vec{t}d\varphi, \tag{5.42}$$

где вектор $\vec{t}$ по определению равен [14]

$$\vec{t} = \frac{d\vec{r}}{d\varphi}. \tag{5.43}$$

Из (5.41), (5.43) найдем

$$\vec{t} = \begin{bmatrix} -(R_T \sin\varphi + a\sin\varphi\cos N\varphi + Na\sin N\varphi\cos\varphi) \\ R_T\cos\varphi - aN\sin N\varphi\sin\varphi + a\cos N\varphi\cos\varphi \\ aN\cos N\varphi \end{bmatrix}. \tag{5.44}$$

Выражение для напряженности магнитного поля, создаваемого элементом тороидальной обмотки в точке пространства, положение которой описывается радиусом-вектором $\vec{R}_0 = (x_0, y_0, z_0)$, учитывая (5.36), (5.44), можно записать в виде

$$d\vec{B} = \frac{\mu_0 I}{4\pi}\frac{[\vec{t}\times(\vec{R}_0 - \vec{r})]}{|\vec{R}_0 - \vec{r}|^3}d\varphi, \tag{5.45}$$

где

$$\vec{R}_0 - \vec{r} = \begin{bmatrix} x_0 - (R_T + a\cos N\varphi)\cos\varphi \\ y_0 - (R_T + a\cos N\varphi)\sin\varphi \\ z_0 - a\sin N\varphi \end{bmatrix}^T, \tag{5.46}$$

$$|\vec{R}_0 - \vec{r}| = \\ = \sqrt{(x_0-(R_T+a\cos N\varphi)\cos\varphi)^2+(y_0-(R_T+a\cos N\varphi)\sin\varphi)^2+(z_0-a\sin N\varphi)^2}\ . \tag{5.47}$$

Составляющие магнитного поля, создаваемого тороидальной обмоткой с постоянным током в точке с радиусом-вектором R_0, вдоль координатных осей находятся непосредственным интегрированием (5.45). Окончательные выражения компонентов напряженности магнитного поля через интегралы элементарно получаются подстановкой выражений (5.46), (5.47) и (5.37) в (5.45):

$$B_x = \tfrac{\mu_0 I}{4\pi}\times \\ \times\int\limits_0^{2\pi}\frac{(R_T\cos\varphi - aN\sin N\varphi\cos\varphi + a\cos N\varphi\cos\varphi)(z_0 - a\sin N\varphi) - aN\cos N\varphi(y_0-(R_T+a\cos N\varphi)\cos\varphi)}{[\sqrt{(x_0-(R_T+a\cos N\varphi)\cos\varphi)^2+(y_0-(R_T+a\cos N\varphi)\sin\varphi)^2+(z_0-a\sin N\varphi)^2}]^3}d\varphi\ , \tag{5.48}$$

$$B_y = \tfrac{\mu_0 I}{4\pi}\times \\ \times\int\limits_0^{2\pi}\frac{(R_T\sin\varphi + a\sin\varphi\cos N\varphi + Na\sin N\varphi\cos\varphi)(z_0 - a\sin N\varphi) + aN\cos N\varphi(x_0-(R_T+a\cos N\varphi)\cos\varphi)}{[\sqrt{(x_0-(R_T+a\cos N\varphi)\cos\varphi)^2+(y_0-(R_T+a\cos N\varphi)\sin\varphi)^2+(z_0-a\sin N\varphi)^2}]^3}d\varphi\ , \tag{5.49}$$

$$B_z = -\tfrac{\mu_0 I}{4\pi}\int\limits_0^{2\pi}\frac{(R_T\sin\varphi + a\sin\varphi\cos N\varphi + Na\sin N\varphi\cos\varphi)(y_0-(R_T+a\cos N\varphi)\sin\varphi)}{[\sqrt{(x_0-(R_T+a\cos N\varphi)\cos\varphi)^2+(y_0-(R_T+a\cos N\varphi)\sin\varphi)^2+(z_0-a\sin N\varphi)^2}]^3}d\varphi\ . \tag{5.50}$$

Как видно из выражений (5.48)–(5.50), интегралы, описывающие составляющие магнитного поля тороидальной обмотки с постоянным током, не выражаются через элементарные функции, поэтому возможно только численное исследование особенностей обсуждаемого поля.

Важно отметить, что несмотря на громоздкость выражений (5.48)–(5.50), вычислительные средства MATLAB позволяют записать решение данной задачи в компактной форме. При составлении программы можно использовать подход, описанный в предыдущем разделе при расчете и визуализации магнитного поля соленоида с постоянным током. Однако результаты тестирования соответствующей программы показали, что для расчета напряженности магнитного поля соленоида даже для сетки с небольшим числом узлов требуется неприемлемо много времени. Как показала пошаговая прогонка программы, наибольшие временные затраты требует операция вычисления векторного произведения, входящего в выражение (5.45), которое приходится вычислять многократно для каждого значения переменной φ. По этой причине мы отказались от использования встроенной в MATLAB функции `cross`, записывая явные выражения для каждой координаты векторного произведения. Ниже приводится листинг файла `B_Tor.m`, содержащий описание функции, возвращающей значения составляющих напряженностей вдоль соответствующих координатных осей.

```
% листинг файла B_Tor.m
function [Bx,By,Bz]=B_Tor(x0,y0,z0,Rt,a,N);
% функция, возвращающая значение
% составляющих напряженности
% магнитного поля вдоль соответствующих
% координатных осей
dphi=pi/(5*N); % шаг интегрирования
phi=0:dphi:2*pi; % вектор, содержащий значения
                 % узлов интегрирования
[rx ry rz]=r_Tor(Rt,a,N,phi); % вычисление координат вектора r
                              % в соответствии с (5.41) в узлах
                              % интегрирования
[tx ty tz]=t_Tor(Rt,a,N,phi); % вычисление координат вектора t,
                              % в соответствии с (5.44)
N2=length(rx); % длина вектора
% вычисление составляющих
% напряженности магнитного поля
% в соответствии с (5.48)-(5.50)
X0(1:N2)=x0;
Y0(1:N2)=y0;
Z0(1:N2)=z0;

Rx=X0-rx;
Ry=Y0-ry;
Rz=Z0-rz;

Z=(Rx.^2+Ry.^2+Rz.^2).^(3/2);
```

```
f1=(ty.*Rz-tz.*Ry)./Z;
f2=(tx.*Rz-tz.*Rx)./Z;
f3=(tx.*Ry-ty.*Rx)./Z;

Bx=trapz(phi,f1);
By=trapz(phi,f2);
Bz=trapz(phi,f3);

function [rx,ry,rz]=r_Tor(Rt,a,N,phi)
% функция, возвращающая координаты вектора
% r в соответствии с (5.41)
rx=(Rt+a*cos(N*phi)).*cos(phi);
ry=(Rt+a*cos(N*phi)).*sin(phi);
rz=a*sin(N*phi);

function [tx,ty,tz]=t_Tor(Rt,a,N,phi)
% функция, возвращающая координаты
% вектора t в соответствии с (5.44)
tx=-(Rt*sin(phi)+...
     a*sin(phi).*cos(N*phi)+...
     a*N*sin(N*phi).*cos(phi));
ty=Rt*cos(phi)-...
   N*a*sin(phi).*sin(N*phi)+...
   a*cos(N*phi).*cos(phi);
tz=a*N*cos(N*phi);
```

Далее для вычисления и визуализации напряженности магнитного поля в плоскости YOZ необходимо выполнить следующую последовательность команд, сохраненную нами в файле `Glava5_6.m`:

```
N1=21; % число узлов
i=1:N1+1;
Ymin=-100; Zmin=-100; % левый нижний угол
                      % координатной сетки
Ymax=100; Zmax=100; % правый верхний угол
                    % координатной сетки

% вычисление координат узлов сетки
Y=Ymin+(Ymax-Ymin)/N1*(i-1);
Z=Zmin+(Zmax-Zmin)/N1*(i-1);
Rt=1; % радиус большого кольца тора
a=0.5; % радиус малого кольца тора
N=100; % число витков

% вычисление составляющих
% напряженности магнитного
```

```
% поля в узлах координатной сетки
for i=1:N1+1
  for j=1:N1+1
    [bx by bz]=B_Tor(0,Y(i),Z(j),Rt,a,N);
    Bx(i,j)=bx;
    By(i,j)=by;
    Bz(i,j)=bz;
  end;
end;

% нормировка векторов
mp=(By.^2+Bz.^2).^0.5;
y1=By./mp;
bz1=Bz./mp;
quiver(Y,Z,by1,bz1,0.5); % визуализация поля
                         % нормированных векторов,
                         % касательных к силовым линиям
                         % магнитного поля
```

Рис. 5.29. Поле нормированных векторов, касательных к силовым линиям внешнего магнитного поля тороидальной обмотки

Результаты выполнения описанной последовательности команд представлены на рис. 5.29.

Отметим, что как и в случае с прямым соленоидом, использование многослойной тороидальной обмотки с числом слоев n, намотанной в одном направлении, приведет к пропорциональному увеличению (в n раз) количества электричества, перемещающегося в единицу времени вдоль окружности тора радиуса R, по сравнению с однослойной обмоткой. Следовательно, напряженность поля многослойной тороидальной обмотки будет в n раз больше поля однослойной обмотки B_1:

$$B = NB_1.$$

При использовании многослойной тороидальной обмотки, в которой направление намотки в каждом следующем слое противоположно предыдущему, при четном количестве слоев суммарное количество электричества, перемещающегося в единицу времени вдоль окружности тора радиуса R, равно нулю, при нечетном количестве слоев — равно количеству электричества, перемещающегося в единицу времени вдоль оси радиуса R, в однослойной тороидальной обмотке. Следовательно, тороидальная обмотка с четным количеством слоев обмотки не имеет внешнего магнитного поля, а тороидальная обмотка с нечетным числом слоев имеет внешнее магнитное поле, равное полю однослойной тороидальной обмотки.

***Задача* 5.7**

Как известно [13], *напряженность магнитного поля витка с током имеет нулевые составляющие вдоль осей OX, OY и ненулевую составляющую вдоль оси OZ, зависящую от расстояния до центра витка:*

$$B_z = \frac{\mu_0 I}{4\pi} \frac{2\pi R^2}{\left(R^2 + z^2\right)^{3/2}},$$

где R — радиус катушки, z — расстояние между точкой наблюдения и центром витка.

1. *Вычислите зависимость напряженности магнитного поля тороидальной обмотки на оси OZ от расстояния до центра тора (радиус тора 5 см, радиус поперечного сечения 0,5 см, число витков $N = 500$) и сравните с аналогичной зависимостью для витка с током.*
2. *Объясните причину отсутствия проявлений внешнего магнитного поля тороидальной обмотки в опытах по визуализации магнитного поля с помощью магнитных опилок* [17]. *(В этих опытах тороидальная обмотка и подстилающая поверхность с металлическими опилками размещались в одной плоскости.)*

5.5. ЧИСЛЕННОЕ РЕШЕНИЕ УРАВНЕНИЙ ЛАПЛАСА И ПУАССОНА

В предыдущих разделах мы находили потенциал и напряженность электрических полей, обусловленных стационарным распределением электрических зарядов, используя принцип суперпозиции. Во многих случаях приходится решать более сложные задачи, из которых наиболее типичными являются следующие:

1. Нахождение потенциала электрического поля при неизвестном местоположении исходных зарядов, но заданном электрическом потенциале на границах области. (Например, задача о распределении потенциала электрического поля, создаваемого системой неподвижных проводников, помещенных в вакуум и подключенных к батареям. Здесь можно измерить потенциал каждого проводника, но задать распределение электрических зарядов на проводниках, зависящее от их формы, весьма сложно.)

2. Нахождение потенциала электрического поля, создаваемого заданным распределением электрических зарядов $\rho(x, y, z)$ в пространстве.

Хорошо известно, что прямой метод вычисления потенциала электрического поля $\varphi(x, y, z)$ в этих задачах состоит в решении уравнения Лапласа

$$\Delta\varphi(x, y, z) = \frac{\partial^2\varphi}{\partial x^2} + \frac{\partial^2\varphi}{\partial y^2} + \frac{\partial^2\varphi}{\partial z^2} = 0, \tag{5.51}$$

$$\Delta\varphi(x, y, z) = \frac{\partial^2\varphi}{\partial x^2} + \frac{\partial^2\varphi}{\partial y^2} + \frac{\partial^2\varphi}{\partial z^2} = -\rho(x, y, z), \tag{5.52}$$

соответственно.

Уравнения (5.51), (5.52) относятся к классу дифференциальных уравнений в частных производных эллиптического типа.

В дальнейшем мы будем рассматривать только частный случай эллиптических уравнений для потенциала φ, зависящего от двух пространственных переменных. Совершенно очевидно, что для полного решения задачи уравнения (5.51), (5.52) необходимо дополнить граничными условиями. Различают три типа граничных условий:

1) граничные условия Дирихле (значения φ задаются на некоторой замкнутой кривой в плоскости (x, y) и, возможно, на некоторых дополнительных кривых, расположенных внутри области (рис. 5.30));

2) граничные условия Неймана (на границе задается нормальная производная потенциала φ);

3) смешанная краевая задача (на границе задается линейная комбинация потенциала φ и его нормальной производной).

Рассмотрим методы численного решения уравнения (5.52), оказывающегося более общим. Не теряя общности будем предполагать, что решение 5.52 ищется в единичном квадрате. В качестве первого шага получения численного решения преобразуем уравнение (5.52) к виду, удобному для численного решения. Для этого в плоскости (x, y) задаем сетку из $(N + 1) \times (N + 1)$ узлов, покрывающую рассматриваемую область. (Для простоты выбираем шаг сетки h по каждой координатной оси равномерным и одинаковыми.) Узлы сетки будем обозначать парой индексов (i, j), пробегающих значения от 0 до N. В выбранных обозначениях координаты точки равны $(x_i = ih,\ y_j = jh)$. Обозначив значения функций $\varphi(x, y)$, $\rho(x, y)$ в узловых точках $\varphi_{ij} = \varphi(x_i, y_j)$, $\rho_{ij} = \rho(x_i, y_j)$, и используя для аппроксимации производной трехточечную формулу [18, 20], получаем разностную аппроксимацию уравнения (5.52):

Рис. 5.30. Традиционно используемая модель тороидальной обмотки с постоянным током

$$\frac{\varphi_{i+1j} + \varphi_{i-1j} - 2\varphi_{ij}}{h^2} + \frac{\varphi_{ij+1} + \varphi_{ij-1} - 2\varphi_{ij}}{h^2} = -\rho_{ij}. \tag{5.53}$$

Уравнение (5.53) может быть также получено и другим способом, основанном на вариационном принципе. Рассмотрим функционал E поля $\varphi(x, y)$ следующего вида:

$$E = \int\limits_0^1 dx \int\limits_0^1 dy \left[\frac{1}{2}\left(\vec{\nabla}\phi\right)^2 - \rho\phi\right]. \tag{5.54}$$

Так как величина $\left(\vec{\nabla}\varphi\right)$ есть электрическое поле, а ρ — плотность заряда, то величина E равна полной энергии системы. Найдем вариацию функционала E

$$\delta E = \int\limits_0^1 dx \int\limits_0^1 dy \left[\vec{\nabla}\varphi \cdot \vec{\nabla}\delta\varphi - \rho\delta\varphi\right]. \tag{5.55}$$

Интегрируя по частям первый член (5.55), получаем

$$\delta E = \int\limits_C dl \delta\varphi \vec{n} \cdot \vec{\nabla}\varphi + \int\limits_0^1 dx \int\limits_0^1 dy \delta\varphi \left[-\nabla^2\varphi - \rho\right], \tag{5.56}$$

где криволинейный интеграл берется по границе рассматриваемой области (C), a — единичная нормаль к границе. Для вариаций, удовлетворяющих граничным условиям на границе (C) $\delta\varphi = 0$, поэтому криволинейный интеграл в (5.56) также равен 0. Требование равенства δE нулю для всех таких вариаций означает, что $\varphi(x, y)$ удовлетворяет уравнению (5.52).

Для вывода дискретной аппроксимации уравнения в частных производных на основе вариационного принципа аппроксимируем E через значения поля в узлах сетки, используя двухточечную разностную формулу для производной и правило трапеций для вычисления интеграла:

$$E = \frac{1}{2}\sum_{i=1}^{N}\sum_{j=1}^{N}\left[(\varphi_{ij}-\varphi_{i-j})^2 + (\varphi_{ij}-\varphi_{ij-1})^2\right] - h^2\sum_{i=1}^{N-1}\sum_{j=1}^{N-1}\rho_{ij}\varphi_{ij}. \tag{5.57}$$

Накладывая условие

$$\frac{\partial E}{\partial \varphi_{ij}} = 0, \tag{5.58}$$

для всех i, j, приходим к разностному уравнению (5.53).

Предваряя решение двумерных уравнений Лапласа и Пуассона, рассмотрим методы решения одномерной краевой задачи, которая может быть описана уравнением второго порядка

$$\frac{d^2\phi}{dx^2} = -\rho(x), \tag{5.59}$$

с заданными значениями $\varphi(0)$ и $\varphi(1)$. Соответствующий функционал E для данного уравнения имеет вид

$$E = \int\limits_0^1 dx\left[\frac{1}{2}\left(\frac{d\varphi}{dx}\right)^2 - \rho\varphi\right], \tag{5.60}$$

а его дискретный аналог на равномерной сетке с шагом $h = 1/N$

$$E = \frac{1}{2h}\sum_{i=1}^{N}(\varphi_i - \varphi_{i-1})^2 - h\sum_{i=1}^{N-1}\rho_i\varphi_i. \tag{5.61}$$

Варьируя (5.61) по всем φ_i[1], получаем разностное уравнение

$$\varphi_{i+1} + \varphi_{i-1} - 2\varphi_i = -h^2\rho_i, \tag{5.62}$$

где $\varphi_0 = \varphi(0)$, $\varphi_N = \varphi(1)$.

Уравнение (5.62) задает систему линейных уравнений относительно неизвестных переменных φ_i $(i = 1, N-1)$, причем матрица системы уравнений является трехдиагональной (то есть в матрице отличными от нуля оказываются только элементы, расположенные на главной диагонали и двух диагоналях, расположенных выше и ниже главной диагонали). Для небольшого числа точек (например, $N \leq 100$) данную систему уравнений можно решить прямыми методами [21] или использовать специальный прямой метод, разработанный для решения «трехдиагональных» систем [20]. Однако на практике при численном решении эллиптических уравнений приходится использовать сетки, имеющие значительно большее количество узлов, поэтому мы считаем

[1] В выражении для энергии φ_i встречается дважды: в членах $(\varphi_i - \varphi_{i-1})^2$ и $(\varphi_{i+1} - \varphi_i)^2$.

целесообразным ограничиться рассмотрением только итерационных методов (на примере метода релаксации), применяемым в случае больших разряженных матриц.

Перепишем уравнение (5.62), разрешив его относительно переменной φ_i:

$$\varphi_i = \frac{1}{2}\left[\varphi_{i+1} + \varphi_{i-1} + h^2\rho_i\right]. \tag{5.63}$$

Несмотря на то что значения $\varphi_{i-1}, \varphi_{i+1}$, входящие в правую часть (5.63), нам неизвестны, его можно интерпретировать как «уточнение» значений φ_i через значения в соседних точках.

Следовательно, метод решения уравнения (5.63) (метод Гаусса—Зейделя) состоит в следующем:

1) выбрать некоторое начальное приближение для решения уравнения (5.62);

2) продвигаясь по сетке (например, слева направо), уточнить решения уравнения в соответствии с (5.63).

При многократном повторении описанного процесса начальное приближение может сойтись («срелаксировать») к точному решению.

На практике вместо уравнения (5.63) используют обобщенное уравнение, в котором на каждом шаге релаксации заменяется линейной комбинацией из своего старого значения и «улучшенного» по формуле (5.63):

$$\varphi_i \to \varphi_i' = (1-\omega)\varphi_i + \frac{\omega}{2}(\varphi_{i+1} + \varphi_{i-1} + h^2\rho_i). \tag{5.64}$$

Для исследования сходимости описанной процедуры вычислим изменение функционала энергии E. Подставляя (5.64) в (5.61), после ряда преобразований найдем

$$E' - E = -\frac{\omega(2-\omega)}{h}\left[\frac{1}{2}(\varphi_{i+1} + \varphi_{i-1} - h^2\rho_i) - \varphi_i\right]^2. \tag{5.65}$$

Из (5.65) видно, что при $\omega \in]0, 2[$ энергия E не возрастает и, следовательно, итерационный процесс будет сходиться к требуемому минимальному значению. Существование посторонних минимумов энергии свидетельствует о плохой обусловленности линейной системы. Параметр ω определяет скорость сходимости итерационного процесса — скорость релаксации. Принято различать два случая: при $\omega < 1$ говорят, что используется метод «нижней релаксации», при $\omega > 1$ — метод «верхней релаксации».

В качестве примера, иллюстрирующего релаксационный метод, рассмотрим одномерную краевую задачу для уравнения (6.59) с

$$\rho(x) = 12x^2, \quad \varphi(0) = \varphi(1) = 0,$$

имеющую точное решение

$$\varphi(x) = x(1 - x^3)$$

с энергией

$$E = -\frac{9}{14} = -0,64286.$$

Для решения поставленной задачи необходимо создать m-файл, содержащий описание функции, возвращающей матрицу, в n-м столбце которой находятся решения, полученные на n-м итерационном шаге. Листинг файла `Iteration.m`, который содержит описание данной функции, приводится ниже.

```
% листинг файла Iteration.m
function z=iteration(N,Omega,Number_of_Iteration)
% функция, реализующая
% итерационный процесс
h=1/N; % шаг сетки
i=1:N+1;

% начальные значения
x(i)=(i-1)*h;
f=ro(x);
phi(i)=0;

% интерационный цикл
for k=1:Number_of_Iteration
    for i=1:N-1 % прохождение по узлам сетки
      phi(i+1)=(1-Omega)*phi(i+1)+...
                 Omega/2*(phi(i)+...
                 phi(i+2)+...
                 h.^2*f(i+1));
    end;
  if k==1
    q=phi';  % сохранение решения
             % на первом шаге итерации
  else
     q=cat(2,q,phi'); % сохранение
                      % каждого следующего решения
                      % в новый столбец матрицы q
  end;
end;
z=q;

function z=ro(x)
% начальное приближение
z=12*x.^2;
```

Здесь для сохранения решений уравнения, получаемых на каждом итерационном шаге, в соответствующем столбце матрицы `q` мы используем функцию `cat`. Первый аргумент функции определяет способ размещения добавляемого вектора. Если значение первого элемента установить равным единице, то вектор будет добавлен к существующему вектору снизу, если 2 — то новый вектор размещается справа от существующего, то есть формируется матрица.

Для нахождения численных решений, а также отображения точного решения и решений, полученных на выбранных итерационных шагах, на графике следует выполнить следующую последовательность команд, сохраненную нами в файле `Glava5_7.m`:

```
N = 21; % число узлов сетки,
        % в которых
        % ищется решение уравнения
i = 1:N+1;
x=1/N*(i-1);
Omega=1; % параметр итерационного процесса
Niteration=200; % число итераций

y = Iteration(N,Omega,Niteration);

N1=200; % число точек для построения графика функции
j = 1:N1+1;
x1(j) = (j-1)/N1; % координаты узлов сетки
                  % для вычисления
                  % значений точного решения
y1(j) = x1(i).*(1-x1(i).^3); % точное решение
plot(x1,y1,...
     x,y(:,10),'--',...
     x,y(:,50),':',...
     x,y(:,180),'-.');
```

Рис. 5.31. Демонстрация сходимости итерационного решения к точному

Результаты выполнения приведенной выше последовательности команд представлены на рис. 5.31.

Можно продемонстрировать сходимость итерационного решения, построив зависимость функционала энергии как функцию числа итераций. Для решения поставленной задачи необходимо создать m-файл, содержащий описание функции, возвращающей значения функционала энергии на n-м итерационном шаге. Ниже приводится листинг файла `Iteration_of_Energy.m`, который содержит описание данной функции.

```
% листинг файла Iteration_of_Energy.m
function z=Iteration_of_Enegy(N,Omega,Number_of_Iteration)
% функция, возвращающая значения энергии на каждом
% шаге итерационного процесса
h=1/N; % шаг сетки
i=1:N+1;
```

```
% начальные условия
x(i)=(i-1)*h;
f=ro(x);
phi(i)=0;
% итерационный цикл
for k=1:Number_of_Iteration %
   for i=1:N-1 % прохождение по узлам
               % сетки для вычисления решения
      phi(i+1)=(1-Omega)*phi(i+1)+...
               Omega/2*(phi(i)+phi(i+2)+...
               h.^2*f(i+1));
    end;
    e=0;
    for i=2:N+1 % прохождение по узлам
                % сетки для вычисления энергии
        e=e+(phi(i)-phi(i-1)).^2/(2*h)-h*phi(i)*f(i);
    end;
    En(k)=e;
end;
z=En;

function z=ro(x)
% начальное приближение
z=12*x.^2;
```

Для нахождения зависимости энергии от номера шага итерации, а также отображения данной зависимости на графике следует выполнить следующую последовательность команд:

```
>> Ni = 200;
>> i=1:Ni;
>> z=Iteration_of_Energy(21,1,Ni);
>> z1(i)=-9/14;
>> plot(i,z,i,z1,'--');
```

Результаты выполнения перечисленной выше последовательности команд представлены на рис. 5.32.

***Задача* 5.8**

1. *Используя выражения* (5.61), (5.64), *получите формулу* (5.65).
2. *Получите аналитическое решение рассмотренной выше краевой задачи и найдите точное значение энергии.*
3. *Исследуйте зависимость скорости сходимости релаксационного процесса от значения параметра. Сравните скорость сходимости методов верхней и нижней релаксации.*

Рис. 5.32. Зависимость энергии от номера итерации

В практических расчетах читателю могут пригодиться несколько полезных советов, приведенных в [18]:

1. Выбор хорошего начального приближения уменьшает необходимое число итераций.

2. Необходимо использовать оптимальное значение параметра релаксации, которое может быть оценено аналитически или получено эмпирическим путем.

3. Процесс может оказаться более эффективным, если несколько итераций провести по некоторой сеточной подобласти, в которой сеточное решение известно наиболее слабо, что позволяет сэкономить затраты на релаксацию уже отрелаксировавших частей решения.

4. Провести расчет на относительно грубой сетке, для которой объем вычислений невелик, а затем экстраполировать полученное решение на более мелкую сетку и использовать эти значения в качестве начального приближения для последующих итераций.

Для решения двух- и трехмерных задач также может быть использована описанная выше итерационная процедура. В двухмерном случае аналог формулы (5.64) получается решением уравнения (5.53) относительно φ_{ij} и заменой $\varphi_i \to \varphi_{ij}$ в (5.64):

$$\varphi_{ij} \to \varphi_{ij} = (1-\omega)\varphi_{ij} + \frac{\omega}{4}[\varphi_{i+1j} + \varphi_{i-1j} + \varphi_{ij+1} + \varphi_{ij-1} + h^2\rho_{ij}]. \qquad (5.66)$$

Если данный алгоритм применяется последовательно к каждому сеточному узлу (например, все узлы пробегаются последовательно слева направо и сверху вниз), то можно показать, что функционал энергии (5.61), при условии, что ω лежит в соответствующих пределах, всегда уменьшается и процесс сходится к искомому решению.

Продемонстрируем метод релаксации на примерах решения двумерного уравнения Лапласа (5.51) и двумерного уравнения Пуассона (5.52).

Поставим следующую краевую задачу для уравнения Лапласа: найти решение уравнения (5.51) в квадратной области ($0 \leq x \leq 1$ см,$0 \leq y \leq 1$ см) при заданных значениях потенциалов на границах: $u(x,0) = u(x,1) = 10\,B$, $u(0,y) = u(1,y) = 10\,B$. Для решения поставленной задачи необходимо создать m-файл, содержащий описание функции, возвращающей значения потенциала на каждом шаге итерационного процесса. Ниже приводится листинг файла `IterationL.m`, содержащий описание данной функции.

```
% листинг функции IterationL.m
function z=IterationL(N,Omega,Number_of_Iteration,phi)
% функция, возвращающая
% значения потенциала на каждом шаге
% итерационного процесса

h=1/N; % шаг сетки
% вычисление координат узлов сетки
i=1:N+1;
x(i)=(i-1)*h;
j=1:N+1;
y(j)=(j-1)*h;

% итерационный цикл
for k=1:Number_of_Iteration
  % прохождение по узлам сетки
  for j=2:N
      for i=2:N
        phi(i,j)=(1-Omega)*phi(i,j)+...
                 Omega/4*(phi(i+1,j)+...
                 phi(i-1,j)+...
                 phi(i,j+1)+...
                 phi(i,j-1)-...
                 h.^2*f(i+1,j+1)); % релаксация
      end;
  end;
  if k==1
     q=phi;
  else
     q=cat(2,q,phi);
  end;
end;
z=q;

function z=f(x,y)
z=0; % Для решения уравнения Пуассона
     % следует задать функцию, стоящую
     % в правой части уравнения (5.52)
```

Далее необходимо выполнить следующую последовательность команд, сохраненную нами в файле Glava5_8.m:

```
N=15;
i=1:N+1;
j=1:N+1;

% задание граничных условий
mu(i,1)=10; % потенциал на левой границе
mu(i,N+1)=10; % потенциал на правой границе
mu(1,j)=-10; % потенциал на нижней границе
mu(N+1,j)=10; % потенциал на верхней границе

% задание начального приближения
kx=2:N;
ky=2:N;
mu(kx,ky)=12; % параметр релаксации
Omega=1;
Niter=200; % число итераций
z=IterationL(N,Omega,Niter,mu); % решение
                                % уравнения Лапласа
% вычисление векторов и матриц для
% построения карты линий уровня
x(i)=(i-1)/N;
y(j)=(j-1)/N;
[x1 y1]=meshgrid(x,y);
K=100; % номер итерации для построения
       % карты линий уровня
N1=(N+1)*K+1;
N2=(N+1)*(K+1);
A=z(1:N+1,N1:N2); % выделение K-го решения
                  % из общей матрицы
                  % решений
[C,h] = contour(x1,y1,A,17);
```

Результат выполнения приведенной выше последовательности команд представлен на рис. 5.33.

Для просмотра процесса релаксации решения уравнения Лапласа следует выполнить следующую последовательность команд, размещенную нами в конце файла Glava5_8.m:

```
set(gca,'nextplot','replacechildren'); % задание режима
                                       % перерисовки карты
                                       % в одном и том же окне
% создание кадров анимационного клипа
for K=2:Niter-1
```

Рис. 5.33. Карта эквипотенциалей решения уравнения Лапласа на 100-м шаге итерации

```
  N1=(N+1)*(K-1)+1;
  N2=(N+1)*K;
  A=z(1:N+1,N1:N2);
  [C,h]=contour(x1,y1,A,15);
  F(K-1)=getframe; % создание одного кадра
end;
```

MATLAB, в процессе выполнения цикла по переменной K, выводит на экран каждый вновь создаваемый кадр. Для повторного воспроизведения анимационного клипа используется команда `movie(F,n)`, где `F` — имя переменной, в которую помещены кадры клипа, n — число повторений клипа, если значение n не указано, то клип воспроизводится бесконечное число раз (до закрытия графического окна). Для получения стоп-кадра следует щелкнуть по любому пункту меню графического окна. Для продолжения воспроизведения нужно щелкнуть мышью в поле графика. Можно сохранить созданный анимационный клип в файле, используя команду `save`. Например, для созданного выше анимационного клипа данная команда имеет следующий синтаксис:

```
>> save имя_файла F
```

Для загрузки и выполнения ранее созданного анимационного клипа используется следующая последовательность команд:

```
>> load имя_файла имя_переменной
>> movie(имя_переменной)
```

Задача 5.9

1. *Рассмотрите процесс эволюции карты эквипотенциальных поверхностей в процессе итераций.*

2. *Исследуйте скорость сходимости методов нижней и верхней релаксации.*

3. *Измените описанный выше документ, используя для разностной аппроксимации лапласиана 5-точечную формулу, имеющую более высокий порядок точности:*

$$f_i'' = \frac{1}{12h^2}(-f_{i-2} + 16f_{i-1} - 30f_i + 16f_{i+1} - f_{i+2}).$$

4. *Используя описанный выше документ, вычислите зависимость разности потенциалов от расстояния между двумя линейными зарядами. Сравните полученные численные результаты с известным аналитическим решением этой задачи. Исследуйте зависимость скорости сходимости метода релаксаций и точности численного решения от шага сетки.*

5. *Помимо граничных условий Дирихле можно задавать периодические граничные условия, при которых потенциалы на правой и левой, а также на верхней и нижней границах задаются произвольными, но равными по величине друг другу, то есть* $\varphi_{i0} = \varphi_{iN}$, $\varphi_{0j} = \varphi_{Nj}$.

   Эти условия могут описывать пространственно периодическое распределение плотности заряда в кристалле. Модифицируйте соответствующим образом описанный выше документ и решите уравнение Пуассона с периодическими граничными условиями. Как будет выглядеть решение уравнения в данном случае?

Рассмотрим решения краевой задачи двумерного уравнения Лапласа (5.51) для квадратной области ($0 \le x \le 1$ *см*, $0 \le y \le 1$ *см*) с известными потенциалами на границах

$$u(x, 0) = u(x, 10) = 10,$$

$$u(0, y) = u(10, y) = -10,$$

полученного на сетке, состоящей из 15×15 узлов (рис. 5.33). Из рис. 5.33 видно, что линии, расположенные на карте эквипотенциальных уровней оказываются негладкими (имеют в некоторых точках изломы). Наличие изломов, в свою очередь, у линий равных потенциалов свидетельствует о наличии разрывов у производной функции $\vec{\nabla}\varphi(x, y)$, описывающей напряженность электрического поля. С другой стороны, как известно из теории функций комплексного переменного, функция $\varphi(x, y)$, удовлетворяющая уравнению Лапласа, является аналитической [22]. Необходимым и достаточным условием аналитичности функции $\varphi(x, y)$ является непрерывность ее производных, которому, как очевидно, не отвечает полученное нами численное решение. Обнаруженный «дефект» численного решения связан с большим шагом сетки, на которой ищется решения уравнения (5.51). Для его устранения можно использовать два способа:

1) находить численное решение на сетке с меньшим шагом;

2) используя найденное числовое решение на сетке, состоящей из 15×15 узлов, находить значения в точках, не совпадающих с узлами сетки, с помощью интерполяционной процедуры.

Подробное рассмотрение методов интерполяции функций и связанных с этой задачей проблем выходит за рамки данного раздела, поэтому мы ограничиваемся только одним методом, получившим в последние годы наибольшее распространение в прикладных исследованиях — методом сплайн-интерполяции. Происхождение этого метода и самого термина «сплайн» связывают с техническим приемом чертежников. При необходимости провести непрерывную кривую через сеточный график x_n, y_n на бумаге наносят точки x_n, y_n. Около каждой из них втыкают две булавки и через образовавшийся коридор пропускают гибкую и упругую стальную линейку («сплайн»). Форма, которую принимает эта линейка, является решением задачи о сплайн-интерполяции сеточной функции x_n, y_n. Можно показать, что функция $y(x)$, описывающая форму линейки, сводится к уравнению $y^{(IV)} = 0$. Из уравнения Эйлера следует, что функция $y = y(x)$ является кубическим полиномом, при этом на каждом интервале $[x_n, x_{n+1}]$ имеется свой кубический многочлен. Для обеспечения гладкости интерполяционной процедуры накладывается требование непрерывности в узловых первой и второй производных функции $y(x)$. Аппарат сплайн-интерполяции в настоящее время существенно развит (в частности, разработаны методы интерполяции функций, зависящих от двух переменных) и доведен до уровня стандартных функций, включенных в современные программные средства для математических вычислений.

Для использования сплайн-интерполяции следует дополнить приведенную выше последовательность команд следующими командами:

```
% задание координатной сетки
% для вычисления значений сплайна
M=100;
n=1:M;
x2(n)=(n-1)/M;
y2(n)=(n-1)/M;
[X2 Y2]=meshgrid(x2,y2);
zi3 = interp2(x1,y1,A,X2,Y2,'splain'); % вычисление
                                       % значений сплайна
[C,h]=contour(X2,Y2,zi3,17);
```

Результаты выполнения дополнительных команд представлены на рис. 5.34.

Сравнение зависимостей, представленных на рис. 5.33, 5.34, показывает, что, используя сплайн-интерполяцию, удалось устранить недостатки численного решения, проявляющиеся в наличии изломов линий равного потенциала.

***Задача* 5.10**

1. *Используя массив интерполированных значений решения уравнения Лапласа, вычислите напряженность электрического поля и постройте карту нормированных векторов единичной длины, касательных к силовой линии электрического поля. (Указание: см. раздел 5.1.)*

2. *Постройте карту эквипотенциалей функции* $|\vec{E}(x,y)|$. *Сравните карты линий уровня функций* $\varphi(x,y)$ *и* $\left|\vec{E}\,(x,y)\right|$.

Рис. 5.34. Карта эквипотенциалей решения уравнения Лапласа на 100-м шаге итерации при использовании сплайн-интерполяции

Последовательность действий, позволяющая найти и провести последущую визуализацию численного решения уравнения Пуассона, аналогична описанной выше последовательности действий. Однако требуется внести некоторые изменения в m-файл, содержащий определение функции, возвращающей численное решение, поскольку процесс релаксации не должен касаться области с заданной плотностью заряда. Ниже приводится листинг файла `IterationP.m`, содержащий модифицированную функцию, возвращающую численное решение уравнение Пуассона с правой частью вида

$$f(x) = \begin{cases} 7, & \text{if} \quad x \in [0.4, 0.6],\ y \in [0.4, 0.6], \\ 0, & \text{if} \quad x \notin [0.4, 0.6],\ y \notin [0.4, 0.6]. \end{cases} \tag{5.67}$$

```
% листинг файла IterationP.m
function z=IterationP(N,Omega,Number_of_Iteration,phi)
% функция, возвращающая значения
% потенциала на каждом шаге
% итерационного процесса
h=1/N; % шаг сетки
% вычисление координат
% узлов сетки
i=1:N+1;
x(i)=(i-1)*h;
j=1:N+1;
y(j)=(j-1)*h;
f=ro(N,x,y); % вычисление значений
             % функции, описывающей
             % заданное
             % распределение заряда
for k=1:Number_of_Iteration % итерационная процедура
  for j=2:N % проход по узлам сетки
     for i=2:N
```

```
        % проверка, где находится текущий узел
        if (x(i)>=0.4)&(x(i)<=0.6)&(y(j)>=0.4)&(y(j)<=0.6)
           phi(i,j)=f(i,j)*h^2; % если в области
           % с заданным распределением,
           % то равно значению потенциала
        else
           % если вне области, то релаксация
           phi(i,j)=(1-Omega)*phi(i,j)+...
                    Omega/4*(phi(i+1,j)+...
                    phi(i-1,j)+...
                    phi(i,j+1)+...
                    phi(i,j-1)-...
                    phi(i,j)+...
                    h^2*f(i+1,j+1));
        end;
    end;
  end;
  if k==1
    q=phi;
  else
    q=cat(2,q,phi);
  end;
end;
z=q;

function z=ro(N,x,y)
% функция, описывающая
% плотность
% распределения заряда
for j=1:N+1
   for i=1:N+1
      if (x(i)>=0.4)&(x(i)<=0.6)&(y(j)>=0.4)&(y(j)<=0.6)
        z(i,j)=700;
      else
        z(i,j)=0;
      end;
   end;
end;
```

Для нахождения распределения потенциала в рассматриваемой задаче достаточно заменить в строке

```
z=IterationL(N,Omega,Niter,mu);
```

файла `Glava5_8.m` название функции `IterationL` на `IterationP`.

Численное решение рассматриваемой задачи представлено на рис. 5.35.

***Задача* 5.11** *Найдите численные решения граничной задачи* $u(x,0)=u(x,10)=0$, $u(0,y)=u(10,y)=0$ *уравнения Пуассона с заданным распределением*

Рис. 5.35. Карта эквипотенциалей решения уравнения Пуассона на 100-м шаге итерации

плотности заряда (5.67) *и проведите их визуализацию, аналогично тому, как это было сделано выше в данном разделе.*

ЛИТЕРАТУРА К ГЛАВЕ 5

1. *Калашников С. Г.* Электрические и магнитные поля. М.: Наука, 1970.

2. *Фриш С. Э., Тиморева А. В.* Курс общей физики. Т. II. М.: Физматгиз, 1961.

3. *Фейман Р., Лейтон Р., Сэндс М.* Феймановские лекции по физике. Вып. 5, 6. М.: Наука, 1977.

4. *Парселл Э.* Электричество и магнетизм. Берклеевский курс физики. М.: Наука, 1975.

5. *Говорков В. А.* Электрические и магнитные поля. Л.: Гостехиздат, 1960.

6. *Тамм И. Е.* Основы теории электричества. М.: Наука, 1966.

7. *Смайт В.* Электростатика и электродинамика. М.: Иностранная литература, 1954.

8. *Стрэттон Дж. А.* Теория электромагнетизма. М.: Иностранная литература, 1948.

9. *Джексон Дж.* Классическая электродинамика. М.: Иностранная литература, 1965.

10. *Зоммерфельд А.* Электродинамика. М.: ИЛ, 1958.

11. Сборник задач по общему курсу физики. Электричество и магнетизм. / Сб. задач под ред. И. А. Яковлева. М.: Наука, 1977.

12. *Батыгин В. В., Топтыгин И. Н.* Сборник задач по электродинамике. М.: Наука, 1970.

13. *Векштейн Е. Г.* Сборник задач по электродинамике. М.: Наука, 1966.

14. *Смирнов В. И.* Курс высшей математики. Т. 3. М.: Физматгиз, 1958.

15. *Поршнев С. В., Харитонов В. И.* Особенности магнитного поля соленоида с постоянным током // Электричество, 1998. № 6.

16. *Сивухин Д. В.* Общий курс физики. Электричество. Т. III. М.: Наука, 1977.

17. *Поль Р. В.* Учение об электричестве. М.: Физматгиз, 1962.

18. *Кунин С.* Вычислительная физика. М.: Мир, 1992.

19. *Поршнев С. В., Харитонов В. И.* Магнитное поле тороидальной обмотки // Преподавание физики в высшей школе, 1998. № 14.

20. *Федоренко Р. П.* Введение в вычислительную физику. М.: Изд-во Моск. физ.-техн. ин-та, 1994.

21. *Самарский А. А.* Введение в численные методы. М.: Наука, 1987.

22. *Морс Ф. М, Фешбах Г.* Методы теоретической физики. Т. 1. М.: ИЛ, 1958.

Глава 6

МОДЕЛИРОВАНИЕ ДВИЖЕНИЯ ЭЛЕКТРИЧЕСКИХ ЗАРЯДОВ В ПОСТОЯННЫХ ЭЛЕКТРИЧЕСКИХ И МАГНИТНЫХ ПОЛЯХ

6.1. ПОСТАНОВКА ЗАДАЧИ

Как известно, на электрический заряд q, движущийся в скрещенных электрическом $\vec{E}$ и магнитном $\vec{B}$ полях со скоростью $\vec{v}$, действует сила Лоренца $\vec{F}$

$$\vec{F} = q\left(\vec{E} + \left[\vec{v} \times \vec{B}\right]\right). \tag{6.1}$$

В соответствии со вторым законом Ньютона

$$\frac{d\vec{p}}{dt} = \frac{d\left(m\vec{v}\right)}{dt} = \vec{F}, \tag{6.2}$$

где m — масса заряда, при нерелятивистском движении равная массе покоя m_0, и зависящая от скорости по закону

$$m = \frac{m_0}{\sqrt{1 - |\vec{v}|^2/c^2}} = m_0\gamma \tag{6.3}$$

при движении со скоростями, близкими к скорости света.

Следовательно, для полного описания движения заряженной частицы необходимо при заданных напряженностях электрического и магнитного полей решить задачу Коши системы, состоящей из трех дифференциальных уравнений второго порядка:

$$\begin{aligned}
\frac{d}{dt}p_x &= \frac{d}{dt}(mv_x) = q\left(E_x(x,y,z) + \left[\vec{v} \times \vec{B}(x,y,z)\right]_x\right),\\
\frac{d}{dt}p_y &= \frac{d}{dt}(mv_y) = q\left(E_y(x,y,z) + \left[\vec{v} \times \vec{B}(x,y,z)\right]_y\right),\\
\frac{d}{dt}p_z &= \frac{d}{dt}(mv_z) = q\left(E_z(x,y,z) + \left[\vec{v} \times \vec{B}(x,y,z)\right]_z\right).
\end{aligned} \tag{6.4}$$

где индексы x, y, z обозначают проекции векторов на соответствующую ось выбранной системы координат.

Аналитическое решение системы (6.4) может быть найденного для относительно небольшого числа случаев, поэтому для описания движения заряженных частиц в электромагнитных полях необходимо проведение численных расчетов.

6.2. РАССЕИВАНИЕ ЧАСТИЦ В ЦЕНТРАЛЬНОМ ПОЛЕ. ОПЫТ РЕЗЕРФОРДА

Как известно [1], задача об описании результата столкновения двух частиц с массами m_1, m_2 может быть сведена к эквивалентной задаче об отклонении одной частицы с массой m (m — приведенная масса) в поле $u(|\vec{R}|)$ неподвижного силового центра, расположенного в центре инерции. Явление рассеивания состоит в том, что частица с начальной кинетической энергией E и прицельным расстоянием ρ (рис. 6.1) влетает в потенциальное поле на большом удалении от его центра. По мере прохождения через поле частица отклоняется от первоначальной траектории и после удаления от центра поля на достаточное расстояние частица продолжает движение с той же энергией E, но под углом χ к исходному направлению движения. Для полного определения результата столкновения двух частиц необходимо решение уравнений движения (6.2) с учетом конкретного закона взаимодействия частиц. Так как потенциал зависит лишь от расстояния частицы до центра поля, угловой момент частицы является интегралом движения, а вся ее траектория будет лежать в одной плоскости. Поэтому система уравнений движения в данном случае будет состоять из двух уравнений второго порядка или четырех уравнений первого порядка (для двух координат и двух скоростей).

Рис. 6.1. Модель рассеивания частицы на неподвижном центре

Проведем подробное исследование движения альфа-частицы ($m_\alpha = 6.65{\cdot}10^{-27}$ *кг*, $q_\alpha = 2e$, $e = 1.60 \cdot 10^{-19}$ *Кл*) в окрестности ядра атома золота ($m_{Au} = 3.27 \cdot 10^{-25}$ *кг*, $q_{Au} = +79e$) (ядро атома мишени в первом эксперименте Резерфорда по рассеиванию). Так как $m_\alpha \ll m_{Au}$, будем принимать при численных расчетах, что $m \cong m_{Au}$, и рассматривать ядро атома золота в акте рассеяния как неподвижное. Поскольку альфа-частица не проникает в ядро, то можно считать, что взаимодействие между частицей и ядром описывается законом Кулона

$$\vec{F} = \frac{1}{4\pi\varepsilon_0} \cdot \frac{qQ}{|\vec{R}|^3}\vec{R}, \tag{6.5}$$

где ε_0 — диэлектрическая проницаемость вакуума ($\varepsilon_0 \approx 8.85 \cdot 10^{-12}$ *ф/м*), q, Q — заряды рассеиваемой частицы и кулоновского центра соответственно, $\vec{R}$ — радиус-вектор заряда q.

Следовательно, система уравнений движения альфа-частицы в прямоугольной системе координат, центр которой совпадает с зарядом Q, будет иметь следующий вид:

$$\begin{aligned} m\frac{d^2x}{dt^2} &= \frac{1}{4\pi\varepsilon_0}\frac{qQ}{(x^2+y^2)^{3/2}}x, \\ m\frac{d^2y}{dt^2} &= \frac{1}{4\pi\varepsilon_0}\frac{qQ}{(x^2+y^2)^{3/2}}y. \end{aligned} \tag{6.6}$$

Для дальнейшего численного решения удобно записать (6.6) в виде системы 4-х уравнений первого порядка

$$\begin{aligned} v_x &= \frac{dx}{dt}, \\ m\frac{dv_x}{dt} &= \frac{1}{4\pi\varepsilon_0}\frac{qQ}{(x^2+y^2)^{3/2}}x, \\ v_y &= \frac{dy}{dt}, \\ m\frac{dv_y}{dt} &= \frac{1}{4\pi\varepsilon_0}\frac{qQ}{(x^2+y^2)^{3/2}}y. \end{aligned} \tag{6.7}$$

Для однозначного вычисления траектории движения рассеиваемой частицы необходимо задать следующие начальные условия: $x(0)$, $y(0)$, $v_x(0)$, $v_y(0)$.

Для численных расчетов оказывается удобным записать систему уравнений (6.6) в безразмерных переменных $\tilde{t}$, $\tilde{x}$, $\tilde{y}$, $\tilde{v}_x$, $\tilde{v}_y$, $\tilde{m}$, $\tilde{q}$, $\tilde{Q}$, связанных с размерными переменными следующими соотношениями:

$$\tilde{t} = \frac{t}{T}, \tag{6.8}$$

$$\tilde{x} = \frac{x}{a}, \tag{6.9}$$

$$\tilde{y} = \frac{y}{a}, \tag{6.10}$$

$$\tilde{v}_x = \frac{v_x}{(a/T)}, \tag{6.11}$$

$$\tilde{v}_y = \frac{v_y}{(a/T)}, \tag{6.12}$$

$$\tilde{m} = \frac{m}{M}, \tag{6.13}$$

$$\tilde{q} = \frac{q}{Q_0}, \tag{6.14}$$

$$\tilde{Q} = \frac{Q}{Q_0}, \tag{6.15}$$

здесь T, a, M, Q_0 — единицы измерений времени, расстояния, массы и заряда соответственно.

Подставив (6.8)–(6.15) в (6.7), найдем

$$\begin{aligned} \tilde{v}_x &= \frac{d\tilde{x}}{d\tilde{t}}, \\ \frac{\tilde{m}}{M}\frac{a}{T^2}\frac{d\tilde{v}_x}{d\tilde{t}} &= \frac{Q_0^2}{4\pi\varepsilon_0 a^2}\frac{\tilde{q}\tilde{Q}}{(\tilde{x}^2+\tilde{y}^2)^{3/2}}\tilde{x}, \\ \tilde{v}_y &= \frac{d\tilde{y}}{d\tilde{t}}, \\ \frac{\tilde{m}}{M}\frac{a}{T^2}\frac{d\tilde{v}_y}{d\tilde{t}} &= \frac{Q_0^2}{4\pi\varepsilon_0 a^2}\frac{\tilde{q}\tilde{Q}}{(\tilde{x}^2+\tilde{y}^2)^{3/2}}\tilde{y}. \end{aligned} \tag{6.16}$$

Оставляя в левых частях уравнений (6.16) только безразмерные величины, преобразуем их к следующему виду:

$$\begin{aligned} \tilde{v}_x &= \frac{d\tilde{x}}{d\tilde{t}}, \\ \tilde{m}\frac{d\tilde{v}_x}{d\tilde{t}} &= K\frac{\tilde{q}\tilde{Q}}{(\tilde{x}^2+\tilde{y}^2)^{3/2}}\tilde{x}, \\ \tilde{v}_y &= \frac{d\tilde{y}}{d\tilde{t}}, \\ \tilde{m}\frac{d\tilde{v}_y}{d\tilde{t}} &= K\frac{\tilde{q}\tilde{Q}}{(\tilde{x}^2+\tilde{y}^2)^{3/2}}\tilde{y}. \end{aligned} \tag{6.17}$$

где

$$K = \frac{Q_0^2 M T^2}{4\pi\varepsilon_0 a^3}. \tag{6.18}$$

Анализ выражения (6.18) показывает, что численное значение коэффициента K зависит от величин T, a, M, Q_0. Выбор величин M, Q_0 достаточно очевиден: в качестве единиц измерения массы следует выбрать массу протона

$$M = 1,6710^{-27}\ кг,$$

а в качестве единиц измерения заряда — заряд электрона

$$Q_0 = e.$$

Для выбора числовых значений величин T, a заметим, что отношение a/T имеет размерность скорости, поэтому их значения выберем так, чтобы обеспечить выполнение условия

$$\frac{a}{T} = c, \tag{6.19}$$

где c — скорость света в вакууме ($c = 3\cdot 10^8\ м/с$).

Из сравнения (6.8)–(6.15) и (6.19) очевидно, что сделанный выбор единиц измерения длины и времени автоматически обеспечивает измерение скорости частицы в долях скорости света. Подставляя (6.19) в (6.18), найдем

$$K = \frac{Q_0^2 M}{4\pi\varepsilon_0 c^2 a}. \tag{6.20}$$

Подставляя в (6.20) числовые значения M, Q_0, ε_0, c, найдем

$$K \approx \frac{1,53\cdot 10^{-18}\ м}{a}. \tag{6.21}$$

Из (6.21) очевидно, что выбирая значение $a = 10^{-18}\ м$, мы обеспечиваем численное значение коэффициента K близким к единице, то есть записываем систему (6.17) в виде наиболее удобном для численного решения. Отметим, что в выбранных единицах измерения размер ядра атома золота ($10\cdot 10^{-15}\ м$) составляет 10^4. Следовательно,

при моделировании рассеивания альфа-частицы на ядре атома золота значения начальных координат частицы по оси OX нужно выбирать из условия $x(0) \gg 10^4$.

Таким образом, в выбранных единицах измерения заряд рассеивающего центра

$$Q = 79,$$

заряд рассеиваемой частицы

$$q = 2,$$

ее масса

$$m = 4,$$

значения начальной скорости движения частицы находятся в интервале $[0, 1]$.

Весьма важным является вопрос о выборе длительности временного интервала, на котором находится численное решение системы ДУ (6.17), так как только при его правильном выборе можно точно определить угол рассеивания χ. Совершенно очевидно, что длительность временного интервала зависит от начального положения частицы и начальной скорости, а также величин электрических зарядов взаимодействующих частиц и должна подбираться для конкретных параметров моделируемой системы. Для выбора оптимальной длительности временного интервала можно использовать следующий алгоритм, основанный на постоянстве угла между вектором скорости и осью OX, в областях невозмущенного движения частицы:

1. Найти численное решение системы ДУ (6.17) на временном интервале длительностью $\approx 2\tilde{x}(0)/\tilde{v}_x(0)$.

2. Построить зависимость угла между вектором скорости и осью OX $\chi = \chi(t)$ от времени:
$$\chi(t) = \arccos\left(\tilde{v}_x(t)/\sqrt{(\tilde{v}_x(t))^2 + (\tilde{v}_y(t))^2}\right).$$

3. При отсутствии участков, на которых
$$\chi(t) \approx \text{const},$$
ищется численное решение на временном интервале большей длительности и повторно проводится анализ зависимости $\chi = \chi(t)$.

В связи с тем, что в ходе решения рассматриваемой задачи приходится решать систему дифференциальных уравнений, на первом шаге решения следует создать m-файл, содержащий описание функции, возвращающей значения правой части системы ДУ (6.17). Далее приводится листинг соответствующего файла `CenterPole.m`.

```
% листинг файла CenterPole.m
function dy=CenterPole(t,z)
% описание функции,
% возвращающей значения
% правой части системы (6.17)
```

```
global q Q M % задание глобальных переменных
             % для получения
             % их значений
             % из вызывающей программы
A=1.53*q*Q/M;
dy=zeros(4,1); % задание вектора-столбца
               % размерности 4x1
% задание координат вектора-функции,
% стоящей в правой части (6.17)
dy(1)=z(2);
dy(2)=A*z(1)/(z(1)^2+z(3)^2)^(3/2);
dy(3)=z(4);
dy(4)=A*z(3)/(z(1)^2+z(3)^2)^(3/2);
```

Для нахождения численного решения системы ДУ (6.17) необходимо выполнить приведенную ниже систему команд, сохраненную нами в файле `Glava6_1.m`:

```
% листинг файла Glava6_1.m
global q M Q % задание глобальных переменных
q=2; % заряд рассеиваемой частицы
M=4; % масса рассеиваемой частицы
Q=79; % заряд рассеивающего центра
x0=-10^6; % x-я координата начальной точки
y0=4*10^4; % значение прицельного параметра
vx0=0.1; % x-я составляющая
         % начальной скорости
vy0=0; % y-я составляющая
       % начальной скорости
Tmax=2*10^7 % длительность временного интервала,
            % на котором ищется численное решение системы ДУ (6.17)
dt=Tmax/2000; % шаг интегрирования системы ДУ (6.17)
% решение системы ДУ
[T Y]=ode45('CenterPole',[0:dt:Tmax],[x0 vx0 y0 vy0]);
% визуализация численного решения ДУ (6.17)
figure(1);
plot(Y(:,1),Y(:,3)); % построение траектории движения

xi=acos(Y(:,2)./(Y(:,2).^2+Y(:,4).^2).^0.5); % вычисление угла
                                             % наклона вектора
                                             % скорости к оси OX
figure(2);
plot(T,xi) % построение зависимости
           % угла рассеивания
           % от времени
```

Результаты выполнения приведенной выше последовательности команд представлены на рис. 6.2, 6.3.

Рис. 6.2. Траектория движения частицы, рассеиваемой на неподвижном ядре

Рис. 6.3. Зависимость угла рассеивания от времени

Задача 6.1

1. *Проведите численные расчеты и постройте на одном чертеже семейство траекторий и углов рассеивания альфа-частицы при различных значениях прицельного параметра. Объясните полученные результаты.*

2. *Постройте зависимость угла рассеивания от прицельного параметра* $\chi = \chi(\rho)$. *(Для этого можно выбрать диапазон изменения прицельного параметра, выбрав шаг, разбить его на конечное число точек и для каждого значения прицельного параметра, решив уравнения движения, определить угол рассеивания.) Объясните, почему угол рассеивания является возрастающей или убывающей функцией прицельного параметра.*

Задача 6.2

Для упрощения процесса вычислений и построения графиков зависимостей, характеризующих процесс рассеивания заряженной частицы на неподвижном центре, разработайте проект графического интерфейса пользователя и создайте его практическую реализацию, используя средства визуального проектирования MATLAB.

В физических применениях обычно приходится иметь дело не с индивидуальным рассеиванием одной частицы, но с рассеиванием целого пучка одинаковых частиц, падающих на центр с одинаковой начальной скоростью и различными прицельными параметрами. Различие в прицельных параметрах ρ обусловливает рассеивание частиц пучка под различными углами χ. Для характеристики исследуемого процесса принято вводить число частиц dN, рассеиваемых в единицу времени на углы, лежащие в интервале $\chi + d\chi$. Однако для характеристики процесса рассеивания более удобной

оказывается величина $d\sigma$ (называемая «эффективным сечением рассеивания»):

$$d\sigma = \frac{dN}{n}, \tag{6.22}$$

где n — число частиц, проходящих в единицу времени через единицу площади поперечного сечения пучка. Эта величина определяется видом рассеивающего поля.

Связь между χ и ρ взаимно однозначна, в чем вы могли убедиться, решив задачу 6.1. Поэтому в заданный интервал углов между χ и $\chi + d\chi$ рассеиваются лишь те частицы, прицельный параметр у которых лежит в интервале между $\rho(\chi)$ и $\rho(\chi)+$ $+d\rho(\chi)$. Число рассеянных частиц равно произведению n на площадь кольца между окружностями с радиусами $\rho(\chi)$ и $\rho(\chi) + d\rho(\chi)$, то есть $dN = 2\pi\rho d\rho n$. Подставив это выражение в (6.22), найдем

$$d\sigma = 2\pi\rho d\rho = 2\pi\rho\left(\chi\right)\left|\frac{d\rho\left(\chi\right)}{d\chi}\right| d\chi. \tag{6.23}$$

Разделив обе части выражения (6.23) на $d\chi$, получаем формулу для дифференциального сечения рассеивания:

$$\frac{d\sigma}{d\chi} = 2\pi\rho\left(\chi\right)\left|\frac{d\rho\left(\chi\right)}{d\chi}\right|. \tag{6.24}$$

Анализ выражения (6.24) показывает, что для нахождение дифференциального сечения рассеивания необходимо знать зависимость $\rho = \rho(\chi)$. Выше мы научились, решая уравнения движения альфа-частицы в кулоновском поле ядра атома золота, вычислять зависимость $\chi = \chi(\rho)$. В связи с тем, что функция $\rho = \rho(\chi)$ является однозначной, нахождение функции $\chi = \chi(\rho)$, обратной к данной, не вызывает затруднений. Однако в выражение (6.24) помимо функции $\rho = \rho(\chi)$ входит и ее производная, которая не может быть вычислена без дополнительных действий, так как функция $\rho = \rho(\chi)$ определена на неравномерной по переменной χ сетке. Выход из создавшейся ситуации состоит в решении классической интерполяционной задачи: нахождении значений функции в промежуточных точках по известным в конечном числе точек значениям функции (узловым точкам). Один из наиболее популярных в настоящее время подходов к решению данной задачи основан на использовании сплайнов (полином третьей степени) [2]. В связи с тем, что подробное рассмотрение теории сплайн-интерполяции выходит за рамки нашей книги, мы ограничиваемся только описанием примера, демонстрирующего использование функции одномерной сплайн-интерполяции. Для вычисления дифференциального сечения рассеивания нами используется следующий алгоритм:

- Задать интервал изменения прицельного параметра ρ.
- Задать шаг изменения дискретного параметра $\Delta\rho$.
- Вычислить для каждого значения прицельного параметра ρ_i численные решения системы ДУ (6.17) и определить соответствующие углы рассеивания χ_i.
- Решить задачи сплайн-интерполяции для функции $\rho_i = \rho(\chi_i)$ на равномерной сетке по переменной χ.

- Вычислить на равномерной сетке по переменной χ численным дифференцированием производную функции $\rho = \rho(\chi)$.
- Вычислить в соответствии с (6.24) дифференциальное сечение рассеивания.

Для реализации описанного алгоритма необходимо выполнить следующую последовательность команд, сохраненную нами в файле `Glava6_2.m`:

```
% листинг файла Glava6_2.m
clear all % очистка рабочего пространства
global Q q M % задание глобальных переменных
Q=79; % заряд рассеивающего центра
q=2; % заряд рассеиваемой частицы
M=4; % масса частицы
Tmax=2*10^7; % длительность временного интервала, на котором
             % ищется численное решение системы ДУ (6.17)
dt=Tmax/2000; % шаг интегрирования
0=-10^6; % x-я координата начальной точки
vx0=0.1; % x-я составляющая начальной скорости
y0=0; % y-я составляющая начальной скорости
Ymin=0; % минимальное значение прицельного параметра
Ymax=10^5; % максимальное значение прицельного параметра
N=100; % число интервалов разбиения отрезка [Ymin,Ymax]
dy=(Ymax-Ymin)/N; % шаг разбиения
% вычисление значений угла рассеивания для N+1
% значения прицельного параметра
for i=1:N+1
      y(i)=(i-1)*dy;
      [T Y]=ode45('CenterPole',[0:dt:Tmax],[x0 vx0 y(i) vy0]);
      M1=length(T);
      xi(i)=acos(Y(M1,2)./(Y(M1,2).^2+Y(M1,4).^2).^0.5);
end;
figure(1); plot(y,xi); % визуализация зависимости
                       % угла рассеивания
                       % от прицельного параметра
% задание параметров сетки
% для вычисления интерполированных
% значений зависимости
x1=min(xi);
x2=max(xi);
Nxi=1000;
dxi=(x2-x1)/Nxi;
i=1:Nxi+1;
Xi(i)=x1+dxi*(i-1); % вычисление координат
                    % узлов сетки
Yi=interp1(xi,y,Xi,'splain'); % вычисление интерполированных
                              % значений зависимости ro=ro(hi)
```

```
dro=abs(diff(Yi)./dxi); % численное дифференцирование
                        % зависимости ro=ro(chi)
j=1:Nxi;
dsigma(j)=2*pi*Yi(j).*dro(j); % вычисление дифференциального
                              % сечения рассеивания
xi1(j)=Xi(j);
figure(2)
plot(xi1,dsigma) % визуализация зависимости дифференциального
                 % сечения рассеивания от угла
```

Результаты выполнения описанной выше последовательности команд представлены на рис. 6.4, 6.5.

Рис. 6.4. Зависимость угла рассеивания от прицельного параметра

Рис. 6.5. Зависимость дифференциального сечения рассеивания от угла

Зависимость дифференциального сечения рассеивания от угла рассеивания, представленная на рис. 6.4, выглядит на первый взгляд несколько неожиданно, так как в выражение для вычисления дифференциального сечения (6.24) входят две монотонные функции: $\rho = \rho(\chi)$, $d\rho(\chi)/d\chi$. Причина присутствия колебаний на рис. 6.5 объясняется тем, что решение системы дифференциальных уравнений, возвращенное функцией `ode45`, оказывается недостаточно точным, и, как следствие, с погрешностью вычисляются значения угла рассеивания. Далее зависимость подвергается процедуре сплайн-интерполяции, которая не может устранить присутствующую в исходных данных погрешность. Далее интерполированная последовательность подвергается процедуре численного дифференцирования, которая, как известно, приводит к обязательному увеличению погрешности, содержащейся в исходных данных. Следствием этого и являются результаты, представленные на рис. 6.5. При использовании решений системы ДУ (6.17), полученных более точным численным методом, реализуемым функцией `ode113`, отмеченной особенности не возникает (рис. 6.6). Естественной платой за более точное решение является примерно двукратное увеличение времени счета.

Рис. 6.6. Зависимость дифференциального сечения рассеивания от угла рассеивания, вычисленная по численному решению уравнения функцией `ode113`

Анализ зависимостей дифференциального сечения рассеивания от угла, представленных на рис. 6.5, 6.6, показывает, что данная зависимость меняется в широком динамическом диапазоне. Для их представления удобно использовать графики, при построении которых на оси OY откладываются значения $\ln(\Delta\sigma)$ (полулогарифмический масштаб). Для построения данного графика в MATLAB используется команда `semilogx`:

```
>> semilogy(xi1,dsigma);
```

результат выполнения которой представлен на рис. 6.7.

***Задача* 6.3**

В опытах Резерфорда происходило рассеивание потока альфа-частиц на ядре атома золота, значения прицельных параметров у которых находились в некотором интервале значений $[0, \rho_{\max}]$, *и затем проводился подсчет числа частиц, попавших в интервал углов* $[\chi, \chi+ +d\chi]$. *Если считать, что распределение альфа-частиц в потоке по прицельному параметру было равномерным (то есть любое значение прицельного параметра из интервала является равновероятным), то можно построить математическую модель опыта Резерфорда, используя следующий алгоритм:*

Рис. 6.7. Зависимость дифференциального сечения рассеивания от угла рассеивания в полулогарифмическом масштабе

1. *Задать* Np *чисел, имеющих значения прицельных параметров, равномерно распределенных на интервале* $[0, \rho_{\max}]$ *(плотность потока частиц* $n = N_p/\pi\rho_{\max}^2$*).*

2. *Для каждой* i*-й частицы вычислить угол рассеивания* χ_i.

3. *Задать число интервалов* N_{int}, *на которые будет разбиваться отрезок* $[\chi_{\min}, \chi_{\max}]$.

4. *Вычислить шаг изменения угла рассеивания* $\Delta\chi$

$$\Delta\chi = \frac{\chi_{\max} - \chi_{\min}}{N_{\text{int}}}.$$

5. *Вычислить количество частиц* ΔN_i, *попавших в интервал* $[\chi_i, \chi_i + \Delta\chi]$.

6. *Вычислить «экспериментальные» значения дифференциального сечения рассеивания для каждого интервала углов*

Постройте компьютерную модель опыта Резерфорда, выполнив для этого следующую последовательность действий:

1. *Создайте в MATLAB документ, реализующий описанный выше алгоритм.*

2. *Постройте на одном чертеже «теоретическую» и «экспериментальную» зависимости дифференциального сечения рассеивания от угла. Сравните полученные результаты.*

3. *Исследуйте, как зависит величина отклонения между «теоретической» и «экспериментальной» зависимостями от плотности потока частиц и величины элементарного угла* $\Delta\chi$.

4. *Пересчитайте созданный вами документ не менее 10 раз, сохраняя при этом результаты вычислений в отдельных файлах. Проведите сравнительный анализ полученных результатов. Проведите усреднение полученных зависимостей и сравните отклонение между теоретической и усредненной «экспериментальной» зависимостями дифференциального сечения рассеивания от угла рассеивания.*

***Задача* 6.4**

Задача о рассеивании заряженной частицы в поле кулоновского центра имеет аналитическое решение

$$\frac{d\sigma}{d\chi} = \pi \left(\frac{\alpha}{mv_\infty^2} \right)^2 \frac{\cos(\chi/2)}{\sin^3(\chi/2)}, \tag{6.25}$$

где v_∞ *— скорость частицы на бесконечно большом удалении от рассеивающего центра,* $\alpha = qQ/4\pi\varepsilon_0$.

1. *Запишите формулу* (6.17), *используя введенные выше безразмерные переменные.*

2. *Вычислите значения дифференциального сечения рассеивания для различных углов рассеивания* χ. *Постройте график зависимости дифференциального сечения рассеивания от угла* χ.

3. *Сравните результаты численных расчетов дифференциального сечения рассеивания и расчетов по формуле* (6.17). *Чем можно объяснить обнаруженные отличия? (Указание: В связи с большим динамическим диапазоном изменения функции* $d\sigma/d\chi$ *при сравнении результатов, полученных численным и аналитическим методами, используйте полулогарифмический график, при построении которого по оси ординат откладываются значения* $\log(d\sigma/d\chi)$.*)*

Другой подход к решению рассматриваемой задачи основан на использовании закона сохранения углового момента

$$L = mv_\infty \rho = mr^2 \frac{d\varphi}{dt}, \tag{6.26}$$

где $d\chi/dt$ — угловая скорость частицы, r — расстояние между частицей и рассеивающим центром, и закона сохранения энергии

$$\frac{mv^2}{2} + U(r) = E. \tag{6.27}$$

В цилиндрической системе координат выражение (6.27) принимает вид

$$\frac{m}{2}(\dot{r}^2 + r^2\dot{\varphi}^2) + U(r) = E. \tag{6.28}$$

Выразив $d\varphi/dt$ из (6.26) и подставив в (6.28), получаем

$$\frac{m\dot{r}^2}{2} + \frac{L^2}{2mr^2} + U(r) = E. \tag{6.29}$$

Отсюда

$$\dot{r} = \sqrt{\frac{2}{m}[E - U(r)] - \frac{L^2}{m^2r^2}}. \tag{6.30}$$

Разделяя в (6.30) переменные, получаем:

$$t = \int \frac{dr}{\sqrt{\frac{2}{m}[E - U(r)] - \frac{L^2}{m^2r^2}}} + \text{const}. \tag{6.31}$$

Если в (6.26) в качестве независимой переменной использовать расстояние r, то можно записать

$$\frac{d\varphi}{dr} = \frac{d\varphi}{dt}\left[\frac{dr}{dt}\right]^{-1} = \frac{L}{mr^2}\left[\frac{dr}{dt}\right]^{-1}. \tag{6.32}$$

Подставляя (6.30) в (6.32), разделяя переменные и интегрируя, находим

$$\varphi = \int \frac{\frac{L}{r^2}dr}{\sqrt{2m[E - U(r)] - \frac{L^2}{r^2}}} + \text{const}. \tag{6.33}$$

Формулы (6.31) и (6.33) решают поставленную задачу в общем виде. Формула (6.33) определяет связь между углом поворота и радиусом-вектором, то есть уравнение траектории. Формула (6.31) определяет в неявном виде зависимость расстояния от времени.

В точке поворота, где квадратный корень в (6.30), а вместе с ним и подынтегральные выражения в (6.31), (6.33) меняют знак. Если отсчитывать угол φ от направления радиуса-вектора, проведенного в точку поворота, то примыкающие с двух сторон к этой точке отрезки траектории будут отличаться только знаком угла φ при одинаковых значениях r. Это означает, что траектория движения рассеиваемой частицы будет

симметрична относительно выбранного направления, поэтому обе асимптоты орбиты пересекают указанную прямую под одинаковыми углами (рис. 6.1). Обозначим угол между асимптотой и указанной прямой φ_0. Тогда, как видно из рис. 6.1, угол отклонения частицы χ при ее пролете мимо центра есть

$$\chi = |\pi - 2\chi_0|. \tag{6.34}$$

Угол φ_0 определяется согласно (6.33) интегралом

$$\varphi_0 = \int\limits_{r_{\min}}^{\infty} \frac{\frac{L}{r^2}dr}{\sqrt{2m[E - U(r)] - \frac{L^2}{r^2}}}, \tag{6.35}$$

где $r_{\min}$ — корень уравнения

$$\sqrt{\frac{2}{m}[E - U(r)] - \frac{L^2}{m^2 r^2}} = 0. \tag{6.36}$$

***Задача* 6.5**

1. *Запишите уравнения* (6.30), (6.31), (6.33), (6.35) *в безразмерном виде.*

2. *Используя обезразмеренные уравнения* (6.31), (6.33), *проведите вычисления и постройте траекторию движения альфа-частицы в электростатическом поле ядра атома золота. Сравните траекторию движения с аналогичной зависимостью, полученной выше решением уравнений движения. Объясните полученные результаты.*

3. *Используя уравнение* (6.34) *и обезразмеренное уравнение* (6.35), *вычислите зависимость* $\rho = \rho(\chi)$ *и изобразите ее графически. Сравните данную зависимость с аналогичной зависимостью, полученной выше решением уравнений движения. Объясните полученные результаты.*

4. *Вычислите значения дифференциального сечения рассеивания для различных углов рассеивания* χ. *Постройте график зависимости дифференциального сечения рассеивания от угла* χ.

5. *Сравните результаты численных расчетов дифференциального сечения рассеивания, полученных решением уравнений движения и прямым вычислением угла рассеивания, с результатами, получаемыми по формуле* (6.25). *Чем можно объяснить обнаруженные отличия? Какой метод расчета оказывается более точным?*

6.3. МОДЕЛИРОВАНИЕ ДВИЖЕНИЯ ЭЛЕКТРИЧЕСКИХ ЗАРЯДОВ В ПОСТОЯННОМ МАГНИТНОМ ПОЛЕ

Для описания движения заряда в постоянном магнитном поле при заданной зависимости напряженности магнитного поля от координат необходимо решить следующую систему ДУ:

$$\begin{aligned}
\frac{dp_x}{dt} &= \frac{d(mv_x)}{dt} = q\Big[\vec{v}\times\vec{B}(x,y,z)\Big]_x,\\
\frac{dp_y}{dt} &= \frac{d(mv_y)}{dt} = q\Big[\vec{v}\times\vec{B}(x,y,z)\Big]_y,\\
\frac{dp_z}{dt} &= \frac{d(mv_z)}{dt} = q\Big[\vec{v}\times\vec{B}(x,y,z)\Big]_z.
\end{aligned} \tag{6.37}$$

Далее в данном разделе рассмотрим моделирование движения электрических зарядов в постоянном однородном магнитном поле $\vec{B}$ при релятивистских и нерелятивистских скоростях. При движении со скоростями много меньшими скорости света система ДУ принимает вид

$$\begin{aligned}
\frac{dv_x}{dt} &= \frac{q}{m_0}\Big[\vec{v}\times\vec{B}\Big]_x,\\
\frac{dv_y}{dt} &= \frac{q}{m_0}\Big[\vec{v}\times\vec{B}\Big]_y,\\
\frac{dv_z}{dt} &= \frac{q}{m_0}\Big[\vec{v}\times\vec{B}\Big]_z.
\end{aligned} \tag{6.38}$$

Предваряя численное решение системы ДУ (6.38), введем безразмерные переменные $\tilde{t} = t/T$, $\vec{\tilde{B}} = \vec{B}/|\vec{B}|$, $\vec{\tilde{v}} = \vec{v}/c$ (c — скорость света), $\vec{\tilde{R}} = \vec{R}/cT$, в которых система (6.38) принимает следующий вид:

$$\begin{aligned}
\frac{d}{d\tilde{t}}\tilde{v}_x &= \frac{qT|\vec{B}|}{m_0}\Big[\vec{\tilde{v}}\times\vec{\tilde{B}}\Big]_x,\\
\frac{d}{d\tilde{t}}\tilde{v}_y &= \frac{qT|\vec{B}|}{m_0}\Big[\vec{\tilde{v}}\times\vec{\tilde{B}}\Big]_y,\\
\frac{d}{d\tilde{t}}\tilde{v}_z &= \frac{qT|\vec{B}|}{m_0}\Big[\vec{\tilde{v}}\times\vec{\tilde{B}}\Big]_z.
\end{aligned} \tag{6.39}$$

Для выбора единиц измерения времени заметим, что на электрический заряд, движущийся в плоскости, перпендикулярной напряженности магнитного поля, действует сила, перпендикулярная его скорости. Следовательно, в данном случае заряд будет двигаться по окружности радиуса R (ларморовской окружности), который можно найти из условия равенства центробежной силы и силы Лоренца

$$\frac{m_0 v^2}{R} = qv|\vec{B}|. \tag{6.40}$$

Откуда

$$R = \frac{m_0 v}{q|\vec{B}|}. \tag{6.41}$$

Вспоминая, что период обращения материальной точки, двигающейся по окружности с равномерной скоростью, равен длине окружности, деленной на модуль линейной скорости, и используя (6.41), найдем

$$T = \frac{2\pi R}{v} = \frac{2\pi m_0}{qt|\vec{B}|}. \tag{6.42}$$

Из (6.42) видно, что выбор периода обращения заряда по ларморовской окружности в качестве единицы измерения времени позволяет записать систему ДУ (6.39) в следующем виде:

$$\begin{aligned} \frac{d\tilde{v}_x}{d\tilde{t}} &= 2\pi \left[\vec{\tilde{v}} \times \vec{\tilde{B}}\right]_x = 2\pi \left(\tilde{v}_y \tilde{B}_z - \tilde{v}_z \tilde{B}_y\right), \\ \frac{d\tilde{v}_y}{d\tilde{t}} &= 2\pi \left[\vec{\tilde{v}} \times \vec{\tilde{B}}\right]_y = 2\pi \left(\tilde{v}_z \tilde{B}_x - \tilde{v}_x \tilde{B}_y\right), \\ \frac{d\tilde{v}_z}{d\tilde{t}} &= 2\pi \left[\vec{\tilde{v}} \times \vec{\tilde{B}}\right]_z = 2\pi \left(\tilde{v}_x \tilde{B}_y - \tilde{v}_y \tilde{B}_x\right). \end{aligned} \tag{6.43}$$

Отметим универсальность системы ДУ (6.43), заключающуюся в том, что в выбранной системе единиц коэффициенты системы не зависят от параметров заряда и напряженности магнитного поля, которые входят в безразмерные начальные условия:

1. радиус-вектор начального положения заряда

$$\vec{\tilde{R}}_0 = \frac{\vec{R}}{cT},$$

2. вектор начальной скорости

$$\vec{\tilde{v}}_0 = \frac{\vec{v}_0}{c}.$$

Легко видеть, что в однородном поле система уравнений (6.43) является замкнутой, так как уравнения зависят только от трех неизвестных функций $\tilde{v}_x(t)$, $\tilde{v}_y(t)$, $\tilde{v}_z(t)$. Найдя численное решение (6.43), далее интегрированием зависимостей $\tilde{v}_x(t)$, $\tilde{v}_y(t)$, $\tilde{v}_z(t)$ можно определить зависимости $\tilde{x}(t)$, $\tilde{y}(t)$, $\tilde{z}(t)$. Однако оказывается более удобным дополнить (6.43) тремя уравнениями для координат:

$$\begin{aligned} \frac{d\tilde{x}}{dt} &= \tilde{v}_x, \\ \frac{d\tilde{y}}{dt} &= \tilde{v}_y, \\ \frac{d\tilde{z}}{dt} &= \tilde{v}_z \end{aligned} \tag{6.44}$$

так как, решив более общую систему ДУ, состоящую из 6 дифференциальных уравнений, одновременно можно найти зависимости $\tilde{x}(t)$, $\tilde{y}(t)$, $\tilde{z}(t)$ и $\tilde{v}_x(t)$, $\tilde{v}_y(t)$, $\tilde{v}_z(t)$. Дополнительный аргумент в пользу данного подхода состоит в том, что общая система уравнений распадается на две независимые системы только для однородного магнитного поля. Для неоднородного поля следует решать систему ДУ, состоящую из 6 дифференциальных уравнений первого порядка.

Для нахождения решения общей системы уравнений движения (6.43) следует создать m-файл, содержащий описание вектор-функции, стоящей в правой части системы ДУ. Ниже приводится листинг соответствующего файла `LorenzForce.m`:

```
% листинг файла LorenzForce.m
function f=LorenzForce(t,z)
% задание функции,
% возвращающей значения
% первых производных
% системы ДУ,
% описывающих движение
% заряженной частицы
% в магнитном поле
global H
f=zeros(6,1); % задание вектора-столбца
              % размерности 6x1
% задание координат вектора-функции,
% стоящей в правой части
% общей системы ДУ
f(1)=z(2);
f(2)=2*pi*(H(3)*z(4)-H(2)*z(6));
f(3)=z(4);
f(4)=-2*pi*(H(3)*z(2)-H(1)*z(6));
f(5)=z(6);
f(6)=2*pi*(z(2)*H(2)-z(4)*H(1));
```

Далее необходимо выполнить следующую последовательность команд, сохраненной нами в файле `Glava6_3.m`:

```
>> global H;
>> H=[0 0 1]; % напряженность магнитного поля
% начальные условия
>> x0=0;
>> y0=0;
>> z0=0;
>> vx0=0.1;
>> vy0=0;
>> vz0=0.01;
% решение системы ДУ
>>  [t R]=ode45('LorenzForce',...
                [0:10/1024:10],...
                [x0 vx0 y0 vy0 z0 vz0]);
% построение траектории движения частицы
>> plot3(R(:,1),R(:,3),R(:,5),...
         x0,y0,z0,'o',...
         'MarkerSize',6);
>> axis square; grid on
```

Результат выполнения приведенной последовательности команд представлен на рис. 6.8.

Рис. 6.8. Траектория движения заряда в постоянном магнитном поле

***Задача* 6.6**

1. *Постройте проекции траектории движения на соответствующие координатные плоскости.*

2. *Проекция траектории движения на плоскость, перпендикулярную вектору напряженности магнитного поля, является окружностью. Определите ее радиус и период обращения по окружности. Объясните полученные результаты.*

3. *Исследуйте зависимость величины шага спирали от скорости движения заряда.*

4. *Постройте зависимости скорости и ее проекций на соответствующие координатные плоскости от времени.*

5. *Исследуйте зависимость точности и устойчивости численного решения системы ДУ* (6.43) *от шага интегрирования.*

Обратимся к анализу особенностей движения заряда, двигающегося с релятивистской скоростью, в постоянном электромагнитном поле. Для этого необходимо разрешить уравнение (6.2) с массой, зависящей от скорости движения по закону (6.3), относительно первой производной скорости. Подставив (6.3) в (6.2) и выполнив дифференцирование, получим

$$\frac{d}{dt}\left(m_0\gamma\vec{v}\right) = m_0\gamma\frac{d}{dt}\vec{v} + m_0\vec{v}\frac{d}{dt}\gamma = \vec{F}, \tag{6.45}$$

где сила $\vec{F}$ задается уравнением (6.1).

Так как

$$\frac{d\gamma}{dt} = \frac{d}{dt}\left(\frac{1}{\sqrt{1-|\vec{v}|^2/c^2}}\right) = \frac{1}{(1-|\vec{v}|^2/c^2)^{3/2}}\frac{(\vec{v}\dot{\vec{v}})}{c^2} = \frac{\gamma^3(\vec{v}\dot{\vec{v}})}{c^2},$$

то (6.45) принимает вид

$$m_0\gamma\dot{\vec{v}} + m_0\vec{v}\frac{\gamma^3\left(\vec{v}\dot{\vec{v}}\right)}{c^2} = \vec{F}. \tag{6.46}$$

Умножая обе части выражения (6.46) скалярно на $\vec{v}$, получаем

$$m_0\gamma\left(\vec{v}\dot{\vec{v}}\right) + m_0\gamma^3\frac{|\vec{v}|^2}{c^2}\left(\vec{v}\dot{\vec{v}}\right) = m_0\gamma^3\left(\vec{v}\dot{\vec{v}}\right) = \left(\vec{F}\vec{v}\right).$$

Откуда

$$\left(\vec{v}\dot{\vec{v}}\right) = \frac{\left(\vec{F}\vec{v}\right)}{m_0\gamma^3}. \tag{6.47}$$

Подставив (6.47) в (6.46), получаем уравнение движения электрического заряда, разрешенное относительно производной:

$$m_0\frac{d\vec{v}}{dt} = \frac{1}{\gamma}\left(\vec{F} - \frac{\vec{v}}{c}\left(\vec{F}\frac{\vec{v}}{c}\right)\right). \tag{6.48}$$

Уравнение (6.48) описывает движение электрических зарядов в произвольном постоянном электромагнитном поле. При движении в постоянном магнитном поле сила Лоренца $\vec{F} = q\left[\vec{v}\vec{B}\right]$ перпендикулярна скорости, следовательно, всегда $\left(\vec{F}\cdot\vec{v}\right) = 0$, поэтому уравнение движения принимает вид

$$m_0\frac{d\vec{v}}{dt} = \frac{1}{\gamma}q\left[\vec{v}\vec{B}\right] = q\left[\vec{v}\vec{B}\right]\sqrt{1-\frac{|\vec{v}|^2}{c^2}}, \tag{6.49}$$

или в координатной записи

$$\begin{aligned}
\frac{d}{dt}v_x &= \frac{q}{m_0}\left[\vec{v}\times\vec{B}\right]_x\sqrt{1-\frac{|\vec{v}|^2}{c^2}},\\
\frac{d}{dt}v_y &= \frac{q}{m_0}\left[\vec{v}\times\vec{B}\right]_y\sqrt{1-\frac{|\vec{v}|^2}{c^2}},\\
\frac{d}{dt}v_z &= \frac{q}{m_0}\left[\vec{v}\times\vec{B}\right]_z\sqrt{1-\frac{|\vec{v}|^2}{c^2}}.
\end{aligned} \tag{6.50}$$

При выборе, как и ранее, в качестве единиц измерения времени периода вращения заряда массой m_0 по ларморовской окружности T, скорости — скорость света c, расстояния — cT система ДУ записывается в виде

$$\begin{aligned}
\frac{d}{d\tilde{t}}\tilde{v}_x &= 2\pi\left[\tilde{\vec{v}}\times\tilde{\vec{B}}\right]_x = 2\pi\left(\tilde{v}_y\tilde{B}_z - \tilde{v}_z\tilde{B}_y\right)\sqrt{1-\left|\tilde{\vec{v}}\right|^2},\\
\frac{d}{d\tilde{t}}\tilde{v}_y &= 2\pi\left[\tilde{\vec{v}}\times\tilde{\vec{B}}\right]_y = 2\pi\left(\tilde{v}_z\tilde{B}_x - \tilde{v}_x\tilde{B}_z\right)\sqrt{1-\left|\tilde{\vec{v}}\right|^2},\\
\frac{d}{d\tilde{t}}\tilde{v}_z &= 2\pi\left[\tilde{\vec{v}}\times\tilde{\vec{B}}\right]_z = 2\pi\left(\tilde{v}_x\tilde{B}_y - \tilde{v}_y\tilde{B}_x\right)\sqrt{1-\left|\tilde{\vec{v}}\right|^2}.
\end{aligned} \tag{6.51}$$

***Задача* 6.7**

1. *Создайте файл, содержащий описание функции, возвращающей значения правой части системы дифференциальных уравнений* (6.51). *Промоделируйте движение зарядов в постоянных однородных магнитных полях с релятивистскими скоростями.*

2. *Постройте проекции траектории движения на соответствующие координатные плоскости. Сравните траектории движения, полученные при одинаковых начальных скоростях, решив нерелятивистские и релятивистские уравнения движения.*

3. *Проекция траектории движения на плоскость, перпендикулярную вектору напряженности магнитного поля, является окружностью. Определите ее радиус и период обращения по окружности. Сравните полученные результаты с аналогичными значениями, полученными для нерелятивистских уравнений движения. Объясните полученные результаты.*

4. *Исследуйте зависимость величины шага спирали от скорости движения заряда?*

5. *Постройте зависимости скорости и ее проекций на соответствующие координатные плоскости от времени.*

6. *Исследуйте зависимость точности и устойчивости численного решения системы ДУ* (6.51) *от шага интегрирования.*

6.4. МОДЕЛИРОВАНИЕ ДВИЖЕНИЯ ЭЛЕКТРИЧЕСКИХ ЗАРЯДОВ В ПОСТОЯННЫХ ЭЛЕКТРИЧЕСКИХ И МАГНИТНЫХ ПОЛЯХ

В этом параграфе мы рассмотрим особенности движения электрических зарядов с нерелятивистскими и релятивистскими скоростями в скрещенных постоянных однородных электрических и магнитных полях. Для описания движения зарядов со скоростями много меньшими скорости света необходимо найти решение системы ДУ (6.4), считая, что масса заряда не зависит от его скорости

$$\begin{aligned} m_0 \frac{d}{dt} v_x &= q \left(E_x(x,y,z) + \left[\vec{v} \times \vec{B}(x,y,z) \right]_x \right), \\ m_0 \frac{d}{dt} v_y &= q \left(E_y(x,y,z) + \left[\vec{v} \times \vec{B}(x,y,z) \right]_y \right), \\ m_0 \frac{d}{dt} v_z &= q \left(E_z(x,y,z) + \left[\vec{v} \times \vec{B}(x,y,z) \right]_z \right). \end{aligned} \tag{6.52}$$

Выбирая, как и ранее, в качестве единиц измерения времени период вращения заряда массой m_0 по ларморовской окружности T, скорости — скорость света c, рас-

стояния — cT запишем систему ДУ (6.52) в виде

$$\begin{aligned}
\frac{d}{d\tilde{t}}\tilde{v}_x &= 2\pi\left[\vec{\tilde{v}}\times\vec{\tilde{B}}\right]_x = 2\pi\tilde{E}_x + 2\pi\left(\tilde{v}_x\tilde{B}_z - \tilde{v}_z\tilde{B}_y\right),\\
\frac{d}{d\tilde{t}}\tilde{v}_y &= 2\pi\left[\vec{\tilde{v}}\times\vec{\tilde{B}}\right]_y = 2\pi\tilde{E}_y + 2\pi\left(\tilde{v}_z\tilde{B}_x - \tilde{v}_x\tilde{B}_z\right),\\
\frac{d}{d\tilde{t}}\tilde{v}_z &= 2\pi\left[\vec{\tilde{v}}\times\vec{\tilde{B}}\right]_z = 2\pi\tilde{E}_z + 2\pi\left(\tilde{v}_x\tilde{B}_y - \tilde{v}_y\tilde{B}_x\right).
\end{aligned} \tag{6.53}$$

где $\tilde{E}_i = E_i/c|\vec{B}|,\ (i = x, y, z)$.

Для однозначного описания движения необходимо задать безразмерные начальные условия: радиус-вектор начального положения заряда $\vec{\tilde{R}}_0 = \vec{R}/cT$ и вектор начальной скорости $\vec{\tilde{v}}_0 = \vec{v}_0/c$.

Рассмотрим более подробно случай, допускающий аналитическое решение: движение во взаимно перпендикулярных однородных электрическом и магнитном полях. Направление вектора $\vec{B}$ выбираем за ось OZ, а плоскость, проходящую через векторы $\vec{B}$ и $\vec{E}$, за плоскость YOZ (рис. 6.9).

Рис. 6.9. Система координат, используемая при описании движения электрического заряда в скрещенных электрическом и магнитном полях

Тогда система уравнений движения (6.53) запишется в виде

$$\begin{aligned}
\ddot{\tilde{x}} &= 2\pi\dot{\tilde{y}}\tilde{B}_z,\\
\ddot{\tilde{y}} &= -2\pi\dot{x}\tilde{B}_z + 2\pi\tilde{E}_y,\\
\ddot{\tilde{z}} &= 2\pi\tilde{E}_z.
\end{aligned} \tag{6.54}$$

Из третьего уравнения (6.54) видно, что вдоль оси OZ заряд двигается равномерно ускоренно, то есть

$$z = \pi\tilde{E}_z t^2 + \tilde{v}_{0z}t. \tag{6.55}$$

Умножая второе уравнение системы (6.54) и складывая с первым, находим

$$\frac{d}{dt}\left(\dot{\tilde{x}} + i\dot{\tilde{y}}\right) + 2\pi i\left(\dot{\tilde{x}} + i\dot{\tilde{y}}\right) = 2\pi i\tilde{E}_y. \tag{6.56}$$

Анализ уравнения (6.56) показывает, что его можно рассматривать как уравнение относительно переменной $\dot{\tilde{x}} + i\dot{\tilde{y}}$. Как известно, интеграл уравнения (6.56) равен сумме интеграла однородного уравнения (уравнение (6.56) без правой части) и частного интеграла уравнения (6.56). Первый из них есть $\dot{\tilde{x}} + i\dot{\tilde{y}} = ae^{-2\pi it}$, второй равен $\tilde{E}_y$.

Таким образом,

$$\dot{\tilde{x}} + i\dot{\tilde{y}} = ae^{-2\pi it} + \tilde{E}_y. \tag{6.57}$$

Постоянная a в (6.57) (в общем случае комплексная) находится из начальных условий. Рассмотрим далее случай, когда в начальный момент времени скорость заряда

направлена вдоль оси OX: $\dot{\tilde{x}}(0) = v_{x0}$, $\dot{\tilde{y}}(0) = 0$. В этом случае постоянная a оказывается вещественной:

$$a = v_{x0} - \tilde{E}_y. \tag{6.58}$$

Подставив (6.58) в (6.57) и отделив вещественную и мнимую части, найдем

$$\begin{aligned} \dot{\tilde{x}} &= a\cos(\pi t) + \tilde{E}_y, \\ \dot{\tilde{y}} &= -a\sin(\pi t). \end{aligned} \tag{6.59}$$

Из (6.59) видно, что компоненты скорости частицы в плоскости XOY являются периодическими функциями времени, средние значения которых равны

$$\begin{aligned} \langle\dot{\tilde{x}}\rangle &= \tilde{E}_y, \\ \langle\dot{\tilde{y}}\rangle &= 0. \end{aligned} \tag{6.60}$$

то есть частица в скрещенных электрическом и магнитном полях движется со средней скоростью в направлении, перпендикулярном плоскости YOZ. Эту скорость называют скоростью электрического дрейфа. Вспоминая, что обезразмеренный вектор напряженности магнитного поля есть единичный вектор, можно записать безразмерную скорость дрейфа в векторном виде как

$$\vec{\tilde{v}}_d = \left[\vec{\tilde{E}}\vec{\tilde{B}}\right]. \tag{6.61}$$

Отметим, что все формулы данного параграфа справедливы при условии малости скорости заряда по сравнению со скоростью света, которое в безразмерном виде имеет вид $\left|\vec{\tilde{v}}\right| \ll 1$. Аналогичное требование справедливо и для скорости дрейфа, которое выполняется при условии, что обезразмеренные напряженности электрического и магнитного полей удовлетворяют условию

$$\tilde{E}_y \ll H_z. \tag{6.62}$$

Интегрируя уравнения (6.59) еще раз и выбирая постоянные интегрирования так, чтобы при $t = 0$ $x = y = 0$, получаем

$$\tilde{x} = \frac{a}{2\pi}\sin(2\pi t) + E_y t, \tag{6.63}$$

$$\tilde{y} = \frac{a}{2\pi}(\cos(\pi t) - 1). \tag{6.64}$$

Уравнения (6.63), (6.64) задают в параметрическом виде кривую, называемую трахоидой.

***Задача* 6.8**

1. *Запишите законы изменения координат заряда и выражения для скорости дрейфа в размерных переменных.*

2. *Запишите условие* (6.53) *в размерных единицах.*

***Задача* 6.9**

Проведите моделирование в MATLAB движения нерелятивистского заряда в скрещенных электрическом и магнитном полях:

1. *Выполните численные расчеты для случая* $v_x \neq 0$, $v_y = v_z = 0$.
   (a) *Убедитесь в справедливости полученных выше аналитических результатов (формулы* (6.58), (6.59)–(6.61), (6.63), (6.64).
   (b) *Постройте траекторию движения частицы в трехмерном пространстве и ее проекции на соответствующие координатные плоскости.*
   (c) *Проведите расчеты для значений начальной скорости, удовлетворяющих условиям* $|a| < |\tilde{E}_y|$, $|a| = |\tilde{E}_y|$, $|a| > |\tilde{E}_y|$, *и сравните траектории движения и их проекции на координатные плоскости.*
   (d) *Проведите расчеты для значений начальной скорости, удовлетворяющей условию* $a = \tilde{E}_y$, *и проанализируйте вид траектории заряда, объясните полученные результаты.*
2. *Проведите расчеты и проанализируйте особенности движения при условии* $v_x \neq 0$, $v_y \neq 0$, $v_z \neq 0$.
3. *Исследуйте особенности движения заряда в параллельных однородных постоянных электрическом и магнитном полях.*
4. *Запишите закон сохранения заряда для рассматриваемого случая в безразмерном виде и оцените точность выполнения для полученных численных решений.*

Для описания движения релятивистского заряда в скрещенных электрическом и магнитном полях необходимо находить решение системы ДУ (6.54). При выборе, как и ранее, в качестве единиц измерения времени период вращения заряда массой m_0 по ларморовской окружности T, скорости — скорость света c, расстояния — cT система ДУ (6.54) записывается в виде

$$\begin{aligned}
\tilde{F}_x &= 2\pi\tilde{E}_x + 2\pi\left(\tilde{v}_y\tilde{B}_z - \tilde{v}_z\tilde{B}_y\right),\\
\tilde{F}_y &= 2\pi\tilde{E}_y + 2\pi\left(\tilde{v}_z\tilde{B}_x - \tilde{v}_x\tilde{B}_z\right),\\
\tilde{F}_z &= 2\pi\tilde{E}_z + 2\pi\left(\tilde{v}_x\tilde{B}_y - \tilde{v}_y\tilde{B}_x\right),\\
\dot{\tilde{v}}_x &= \left[\tilde{F}_x - \tilde{v}_x\left(\tilde{F}_x\tilde{v}_x + \tilde{F}_y\tilde{v}_y + \tilde{F}_z\tilde{v}_z\right)\right]\sqrt{1 - \left|\vec{\tilde{v}}\right|^2},\\
\dot{\tilde{v}}_y &= \left[\tilde{F}_y - \tilde{v}_y\left(\tilde{F}_x\tilde{v}_x + \tilde{F}_y\tilde{v}_y + \tilde{F}_z\tilde{v}_z\right)\right]\sqrt{1 - \left|\vec{\tilde{v}}\right|^2},\\
\dot{\tilde{v}}_z &= \left[\tilde{F}_z - \tilde{v}_z\left(\tilde{F}_x\tilde{v}_x + \tilde{F}_y\tilde{v}_y + \tilde{F}_z\tilde{v}_z\right)\right]\sqrt{1 - \left|\vec{\tilde{v}}\right|^2}.
\end{aligned} \tag{6.65}$$

***Задача* 6.10**

Промоделируйте движение релятивистского электрического заряда, описываемое системой ДУ (6.65)*, в скрещенных электрических и магнитных полях:*

1. *Проведите численные расчеты при условии* $v_x = 0$, $v_y = v_z = 0$ *и сравните с аналогичными зависимостями, полученными при решении системы ДУ* (6.65)*. С каких скоростей начинают проявляться значимые отличия в законах движения?*

2. *Опишите релятивистское движение заряда в параллельных однородных электрическом и магнитном полях.*

3. *Опишите релятивистское движение заряда во взаимно перпендикулярных однородных постоянных электрическом и магнитном полях (рассмотрите случаи* $\tilde{E}_y = \tilde{H}_z$, $\tilde{E}_y > \tilde{H}_z$, $\tilde{E}_y < \tilde{H}_z$*. В каком направлении происходит наиболее быстрое возрастание скорости движения заряда?*

6.5. МОДЕЛИРОВАНИЕ ДВИЖЕНИЯ ЭЛЕКТРИЧЕСКИХ ЗАРЯДОВ В НЕОДНОРОДНОМ МАГНИТНОМ ПОЛЕ

Задача об описании движения электрических зарядов (в частности, вычислении траектории движения) в постоянном неоднородном магнитном поле относится к числу классических задач. Интерес к ней стимулировался в первую очередь активными исследованиями в области управляемого термоядерного синтеза, где требовалось найти ответ на вопрос о возможности удержания частиц в магнитных ловушках, имеющих различные конфигурации магнитных полей [5–7]. Кроме того, данная задача возникает в космической астрофизике [8–9] при объяснении причины существования радиационных поясов у космических тел, обладающих магнитным полем.

Для однозначного описания траектории движения нерелятивистского электрического заряда q в постоянном, неоднородном магнитном поле напряженностью $\vec{H}(\vec{r})$ необходимо при заданных начальных условиях $(\vec{r}(0), \vec{v}(0))$ решить систему уравнений движения, которая в системе *СГСЭq* записывается в виде

$$\begin{aligned} \dot{\vec{r}} &= \vec{v}, \\ \ddot{\vec{r}} &= \frac{q}{mc}\left[\vec{v} \times \vec{H}(\vec{r})\right]. \end{aligned} \tag{6.66}$$

где m — масса заряда, $\vec{r}$ — радиус-вектор, проведенный из начала координат в текущее положение заряда, c — скорость света.

Необходимо отметить, что аналитические решения системы ДУ второго порядка (6.66) удается найти только для нескольких частных случаев (постоянное однородное магнитное поле, магнитный монополь). Для нахождения аналитических решений системы ДУ (6.66) в общем случае приходится использовать приближенные либо численные методы.

Аналитически задача описания движения электрического заряда в слабо неоднородном магнитном поле решается методом усреднения, называемого «дрейфовым приближением» [11]. Используя этот метод, в [7] показано, что движение электрического заряда в неоднородном магнитном поле представляет суперпозицию следующих движений:

1. Вращение по ларморовской окружности со скоростью $\vec{v}_\perp \sim \sqrt{|\vec{H}|}$.

2. Движение центра ларморовской окружности вдоль силовой линии со скоростью $v_{||}$, при этом $v_\perp^2 + v_{||}^2 = \mathrm{const}$ в силу закона сохранения энергии.

3. Дрейфовое движение центра ларморовской окружности со скоростью $\vec{u}_d$ $\left(\vec{u}_d || \left[\vec{H}\left(\vec{r}\right) \times \vec{\nabla}\vec{H}\left(\vec{r}\right)\right]\right)$:

$$|\vec{u}_d| = \frac{1}{\omega_H}\left|\vec{H}(\vec{r})\right|\left(\frac{v_\perp^2}{2} + v_{||}^2\right), \tag{6.67}$$

где $\vec{r}$ — радиус-вектор заряда, ω_H — ларморовская частота, соответствующая напряженности магнитного поля в точке положения заряда.

Условие слабой неоднородности состоит в том, что ларморовский радиус ρ должен быть мал по сравнению с размерами области, в пределах которой вектор напряженности магнитного поля заметно меняется по величине или по направлению:

$$\kappa = \rho\frac{\left|\vec{\nabla}|\vec{H}(\vec{r})|\right|}{|\vec{H}(r)|} \ll 1. \tag{6.68}$$

Другой подход к решению данной задачи основан на том, что слабая неоднородность магнитного поля порождает мало изменяющиеся адиабатические инварианты, количество которых совпадает с числом степеней свободы исследуемой системы. Для рассматриваемой задачи существование адиабатических инвариантов обусловлено каждым из перечисленных выше движений [7, 11]. Вопрос о сохранении адиабатических инвариантов наиболее подробно рассматривался для открытых магнитных ловушек: в [12] получено экспериментальное подтверждение сохранения адиабатического инварианта, связанного с вращением заряда по ларморовской окружности; в [13] доказано вечное сохранение данного адиабатического инварианта в идеальной осесимметричной ловушке с двумерным движением зарядов; в [14] исследовано изменение инвариантов, обусловленное взаимодействием трех степеней свободы.

Наиболее часто в качестве адиабатических инвариантов в рассматриваемой задаче используются:

1) инвариант, обусловленный движением заряда по усредненной ларморовской окружности — магнитный момент:

$$I_1 = \pi\rho^2\frac{H}{2\pi}\frac{q^2}{mc^2} = \frac{mv_\perp^2}{2H}; \tag{6.69}$$

2) инвариант, связанный с продольными колебаниями заряда из-за отражения от магнитных зеркал ловушки:

$$I_2 = \frac{1}{2\pi} \oint \sqrt{E - \mu H} dl, \tag{6.70}$$

где E — полная энергия частицы:

$$E = \frac{mv_\perp^2}{2} + \frac{mv_{||}^2}{2}.$$

Интеграл в (6.70) вычисляется вдоль траектории движения заряда.

Модель слабо неоднородного поля принято также использовать для объяснения движения заряженных частиц, захваченных магнитосферой Земли [5, 6, 11]. При этом вопрос об адекватности данной модели либо не обсуждается [5, 11], либо приводится оценка параметра для протонов и электронов с энергией 10 *Мэв*, находящихся в магнитном поле напряженностью $0,3$ *э* [5].

В связи с этим представляется естественным находить и анализировать численное решение.

В данном разделе мы описываем пакет программ, позволяющий исследовать особенности движения электрических зарядов в поле магнитного диполя. Выбор источника магнитного поля обусловлен тем, что магнитное поле Земли по своей геометрии оказывается близким к полю магнитного диполя, наклоненного под углом 11,5° к оси вращения Земли (рис. 6.10) [8]. Центр магнитного диполя смещен приблизительно на 400 *км* относительно центра Земли. В связи с тем, что данное расстояние оказывается значительно меньше радиуса Земли ($6,4 \cdot 10^3$ *км*), далее в математической модели указанным смещением можно пренебречь и считать, что магнитный диполь находится в начале прямоугольной декартовой системы координат, ось OZ которой направлена вдоль оси вращения Земли (рис. 6.10).

Рис. 6.10. Силовые линии магнитного поля Земли

Для получения замкнутой системы ДУ необходимо дополнить систему уравнений (6.66) аналитическим выражением для напряженности магнитного поля диполя [10]:

$$\vec{H}(\vec{r}) = \frac{3(\vec{\mu}\vec{r})\vec{r}}{|\vec{r}|^5} - \frac{\vec{\mu}}{|\vec{r}|^3}, \tag{6.71}$$

где $\vec{\mu}$ — вектор магнитного диполя, $\vec{r}$ — радиус-вектор точки текущего положения электрического заряда.

Задание начальных условий: радиуса-вектора начальной точки $\vec{R}_0$ и вектора начальной скорости $\vec{v}_0$ позволяет решить задачу Коши для системы уравнений движения (6.66), в которой зависимость магнитного поля от координаты задается выражением (6.71).

Для проведения дальнейших численных расчетов, следуя [16], удобно переписать систему ДУ (6.66) и выражение для напряженности магнитного поля (6.71) в безразмерных переменных. Здесь в качестве единиц измерения расстояния выберем модуль радиуса-вектора $\vec{R}_0$, в качестве единиц измерения модуль магнитного момента $\vec{\mu}_0$ — магнитный момент Земли, используя которые можно и ввести следующие безразмерные переменные:

$$\vec{\tilde{r}} = \frac{\vec{r}}{|\vec{R}_0|}, \tag{6.72}$$

$$\vec{\tilde{\mu}} = \frac{\vec{\mu}}{|\vec{\mu}_0|}. \tag{6.73}$$

Из (6.73) видно, что безразмерный магнитный момент $\vec{\tilde{\mu}}$ оказывается единичным вектором. Далее, подставляя (6.72), (6.73) в уравнение (6.71), получаем

$$\vec{H}\left(\vec{\tilde{r}}\right) = \frac{|\vec{\mu}_0|}{|\vec{R}_0|^3}\left(\frac{3\left(\vec{\tilde{\mu}}\vec{\tilde{r}}\right)\vec{\tilde{r}}}{\left|\vec{\tilde{r}}\right|^5} - \frac{\vec{\tilde{\mu}}}{\left|\vec{\tilde{r}}\right|^3}\right) = |\vec{H}_0|\left(\frac{3\left(\vec{\tilde{\mu}}\vec{\tilde{r}}\right)\vec{\tilde{r}}}{\left|\vec{\tilde{r}}\right|^5} - \frac{\vec{\tilde{\mu}}}{\left|\vec{\tilde{r}}\right|^3}\right), \tag{6.74}$$

где $|\vec{H}_0| = |\vec{\mu}_0|/|\vec{R}_0|^3$ — модуль напряженности магнитного поля в точке начального положения заряда. Из (6.74) видно, что наиболее естественно в качестве единицы измерения напряженности магнитного поля использовать величину $|\vec{H}_0|$. Тогда выражение для безразмерной напряженности магнитного поля примет следующий вид:

$$\vec{\tilde{H}}\left(\vec{\tilde{r}}\right) = \vec{\tilde{H}}\left(\vec{\tilde{r}}\right)/|\vec{H}_0|. \tag{6.75}$$

Выбрав в качестве единиц измерения времени $T_0 = 2\pi mc/|q|\,|\vec{H}_0|$ — период движения заряда по ларморовской окружности в магнитном поле в плоскости, перпендикулярной вектору напряженности $\vec{H}_0$, и используя ранее введенные безразмерные переменные, можно записать систему уравнений движения (6.66), состоящую из 6 уравнений, в следующем виде:

$$\begin{aligned} &\dot{\vec{\tilde{r}}} = \vec{\tilde{v}}, \\ &\ddot{\vec{\tilde{r}}} = 2\pi\left[\vec{\tilde{v}} \times \vec{\tilde{H}}\left(\vec{\tilde{r}}\right)\right], \end{aligned} \tag{6.76}$$

где безразмерные начальные условия вычисляются по следующим выражениям:

$$\vec{\tilde{R}}_0 = \frac{\vec{R}_0}{|\vec{R}_0|}, \tag{6.77}$$

$$\vec{\tilde{v}}_0 = \frac{\vec{v}_0}{|\vec{R}_0|/T_0} = \vec{v}_0\frac{2\pi mc}{|q|\,|\vec{H}_0|\,|\vec{R}_0|}. \tag{6.78}$$

Анализ выражений (6.75)–(6.78) позволяет сделать следующие выводы:

1) система безразмерных уравнений движения оказывается универсальной: коэффициенты системы не зависят от конкретных параметров заряда (массы и величины заряда) и напряженности магнитного поля;

2) безразмерный радиус-вектор, проведенный из начала координат в точку начального положения заряда, — есть единичный вектор;

3) траектория движения заряда в постоянном неоднородном поле, создаваемом магнитным диполем, определяется не отдельными значениями параметров системы — заряда и магнитного поля, но комбинацией этих параметров:

$$\vec{v}_0 \frac{2\pi mc}{|q|\,|\vec{H}_0|\,|\vec{R}_0|};$$

4) при выполнении условия $2\pi mc|\vec{v}_0|/|q|\,|\vec{H}_0|\,|\vec{R}_0| = \mathrm{const}$ траектории движения электрических зарядов в постоянных неоднородных полях, создаваемых магнитными диполями различной величины, оказываются одинаковыми;

5) для исследования особенностей движения заряда в неоднородном магнитном поле достаточно провести вычисления при различных значениях безразмерной начальной скорости.

Для проверки правомерности применения в рассматриваемой задаче модели слабо неоднородного магнитного поля целесообразно вычислить в каждой точке траектории значения коэффициента κ, характеризующего неоднородность магнитного поля, который в безразмерных переменных имеет вид

$$\kappa = \frac{\tilde{v}_\perp}{2\pi} \frac{\left|\vec{\tilde{\nabla}}\left|\vec{\tilde{H}}\right|\right|}{\left|\vec{\tilde{H}}\right|} \ll 1, \tag{6.79}$$

где $\tilde{v}_\perp = v_\perp/\left(R_0/T_0\right)$, и значения адиабатического инварианта, связанного с вращением по ларморовской окружности.

Также для проверки правомерности использования адиабатических инвариантов вычислим мгновенные значения первого адиабатического инварианта (6.69), который в выбранных единицах измерения принимает вид

$$\tilde{I}_1 = \frac{v_\perp^2}{\tilde{H}}. \tag{6.80}$$

***Задача* 6.11**

Получите самостоятельно выражения (6.79), (6.80).

Будем находить численные решения системы (6.76) для следующих параметров модели: электрический заряд — протон

$$m = 1,672 \cdot 10^{-24}\ г,$$

$$q = 4,803 \cdot 10^{-10}\ ед.\ СГСЭq$$

радиус-вектор начального положения точечного заряда

$$\vec{R}_0 = (0, 6, 4 \cdot 10^9, 0)\ см,$$

модуль напряженности магнитного поля $|\vec{H}_0|$ в точке с радиусом-вектором $|\vec{R}_0|$ равен $60\gamma\,(\gamma = 10^{-5}\ э)$ [6], вектор магнитного диполя $\vec{m}$ наклонен под углом $11,5°$ к оси OZ (см. рис. 6.10).

Выбранные параметры, с одной стороны, соответствуют параметрам магнитного поля Земли на расстоянии, отсчитываемом от ее центра, равном 10 радиусам Земли, с другой — позволяют выявить наиболее важные особенности движения зарядов в магнитном поле, создаваемом диполем. Здесь в качестве изменяемого параметра модели можно использовать абсолютное значение начальной скорости заряда при неизменном направлении вектора скорости.

Для нахождения решения общей системы уравнений движения (6.76) следует создать следующие m-файлы, листинги которых приведены ниже:

```
% Листинг функции md.m
function z=md(t,y);
% Функция, описывающая правую часть
% системы дифференциальных
% уравнений движения в форме Коши
% Здесь x=y(1),dx/dt=y(2),y=y(3) и т.д.

global M
r=[y(1) y(3) y(5)]; % r - радиус-вектор частицы
% Задание составлющих вектора
% напряженности магнитного поля
% диполя вдоль координатных осей
Hx=3*M*r'*r(1)/(r*r')^2.5-M(1)/(r*r')^1.5;
Hy=3*M*r'*r(2)/(r*r')^2.5-M(2)/(r*r')^1.5;
Hz=3*M*r'*r(3)/(r*r')^2.5-M(3)/(r*r')^1.5;
% Задание вектора-столбца,
% содержащего первые
% и вторые производные системы
% уравнений движения:
% z(1)=dx/dt=y(2),
% z(2)=dvx/dt=2*pi*(Hz*vy-Hy*vz), и т.д.
z(1)=y(2);
z(2)=2*pi*(Hz*y(4)-Hy*y(6));
z(3)=y(4);
z(4)=2*pi*(Hx*y(6)-Hz*y(2));
z(5)=y(6);
z(6)=2*pi*(Hy*y(2)-Hx*y(4));
z=[z(1);z(2);z(3);z(4);z(5);z(6)];
```

```
% Листинг файла dipol_main.m
% Программа для численного решения
% системы уравнений движения заряженной частицы
% в поле магнитного диполя.
% Ts-шаг интегрирования,
% Tmax-предел интегрирования

global M
% Y0=[x0,vx0,y0,vy0,z0,vz0] - вектор начальных условий.
Y0=[1;0;0;V0;0;0];
% Решение системы ДУ (6.76)
% методом Рунге-Кутта четвертого порядка.
[T,Y]=ode45('md',[0:Ts:Tmax],Y0);
% вычисление модуля
% R - радиус-вектор заряда
x=Y(:,1);
y=Y(:,3);
z=Y(:,5);
R=(x.*x+y.*y+z.*z).^0.5;
% вычисление скорости изменения
% радиуса-вектора во времени.
v=diff(R)./(max(T)/length(T));
V=[v;v(length(v))];
% Вычисление нормированного
% амплитудно-частотного спектра
% модуля радиуса-вектора заряда.
Fmax=1/Ts;
df=1/Tmax;
f=0:df:Fmax;
a=fft(R);
b=abs(a);
s=b./max(b);
% Вычисление составляющих
% и модуля вектора напряженности
% магнитного поля
% в месте положения заряда
Hx=(3*(x.*M(1)+y.*M(2)+z.*M(3)).*x)....
   /(x.*x+y.*y+z.*z).^2.5-...
   M(1)./((x.*x+y.*y+z.*z).^(1.5));
Hy=(3*(x.*M(1)+y.*M(2)+z.*M(3)).*y)....
    /(x.*x+y.*y+z.*z).^2.5-...
    M(2)./((x.*x+y.*y+z.*z).^(1.5));
Hz=(3*(x.*M(1)+y.*M(2)+z.*M(3)).*z)....
    /(x.*x+y.*y+z.*z).^2.5-...
    M(3)./((x.*x+y.*y+z.*z).^(1.5));
```

```
H=(Hx.*Hx+...
   Hy.*Hy+...
   Hz.*Hz).^0.5;
% Вычисление скалярного произведения векторов
% напряженности магнитного поля H
% и вектора скорости
vx=Y(:,2);
vy=Y(:,4);
vz=Y(:,6);
HV=Hx.*vx+Hy.*vy+Hz.*vz;
V1=(vx.*vx+vy.*vy+vz.*vz).^0.5;
% Вычисление квадрата скорости,
% нормальной к вектору напряженности.
V2=V1.^2-(HV./H).^2;
% I - магнитный момент частицы,
% dI - точность сохранения магнитного момента в %.
I=V2./H;
dI=((max(I)-min(I))/mean(I))*100;
% Контроль точности вычислений
% (проверка закона сохранения энергии).
V0=V1(1);
dE=((V0-V1)./V0).*100;
% Вычисление модуля градиента
% напряженности магнитного поля
dr=10^-5;
x1=x-dr;
y1=y-dr;
z1=z-dr;
dHx=(3*(x1.*M(1)+y.*M(2)+z.*M(3)).*x1)./...
    (x1.*x1+y.*y+z.*z).^2.5-...
     M(1)./((x1.*x1+y.*y+z.*z).^(1.5));
dHy=(3*(x.*M(1)+y1.*M(2)+z.*M(3)).*y1)./...
    (x.*x+y1.*y1+z.*z).^2.5-...
     M(2)./((x.*x+y1.*y1+z.*z).^(1.5));
dHz=(3*(x.*M(1)+y.*M(2)+z1.*M(3)).*z1)./...
    (x.*x+y.*y+z1.*z1).^2.5-M(3)./...
    ((x.*x+y.*y+z1.*z1).^(1.5));
H1=(dHx.*dHx+Hy.*Hy+Hz.*Hz).^0.5;
H2=(Hx.*Hx+dHy.*dHy+Hz.*Hz).^0.5;
H3=(Hx.*Hx+Hy.*Hy+dHz.*dHz).^0.5;
G1=(H-H1)./dr; G2=(H-H2)./dr;
G3=(H-H3)./dr;
G=(G1.*G1+G2.*G2+G3.*G3).^0.5;
% Вычисление критерия применимости
% модели слабо неоднородного поля.
```

```
K=(((V2).^0.5).*G)./(2*pi*H);
% Решение уравнений движения
% с новыми начальными условиями.
dx=10^-4;
[T1,Y1]=ode45('md',[0:Ts:Tmax],[1+dx;0;0;V0;0;0]);
% Вычисление расстояния между траекториями,
% близкими в начальный момент времени.
x2=Y1(:,1);
y2=Y1(:,3);
z2=Y1(:,5);
R2=(x2.*x2+y2.*y2+z2.*z2).^0.5;
dR=R-R2;
% Вывод на экран траектории движения частицы,
% зависимости радиуса-вектора заряда R
% от времени и ее спектра,
% а также зависимости dR/dt=f(R)
figure(1);
subplot(2,2,1);
plot(T,R);
grid;
title('Зависимость R=R(t)');
ylabel('R(t)');
xlabel('t');

subplot(2,2,2);
plot(R,V);
grid;title('Зависимость V=V(R)');
ylabel('V');
xlabel('R');

subplot(2,2,3);
semilogy(f,s);
axis([-1 20 0 1]);
grid;
title('Спектр функции R=R(t)');
ylabel('S/Smax');
xlabel('f');

subplot(2,2,4);
plot3(x,y,z);
grid;
title('Траектория движения частицы');
xlabel('x');
ylabel('y');
zlabel('z');
```

```
% Визуализация зависимостей:
% 1) изменения величины полной энергии...
     от номера шага итерации;
% 2) магнитного момента от времени;
% 3) коэффициента, характеризующего...
     неоднородность магнитного поля,
% от времени;
% 4) зависимости расстояния между траекториями...
% близкими в начальный момент времени
figure(2);
subplot(2,2,1);
plot(dE);grid;
title('Проверка закона сохранения энергии');
ylabel('Точность сохранения энергии в %');
xlabel('Номер шага');

subplot(2,2,2);
plot(T,I);
grid;
title('Зависимость M(t)');
ylabel('Магнитный момент');
xlabel('t');

subplot(2,2,3);
plot(T,K);
grid;
title('Коэффициент неоднородности поля');
ylabel('K');
xlabel('t');

subplot(2,2,4);
plot(T,dR);
grid;
title('R(t)-R1(t), где R(0)-R1(0)<<1');
xlabel('t');
ylabel('dR');

% Листинг файла Dipol_menu
% dipol_menu - программа для изменения
% параметров модели в диалоговом режиме.
k=1;
while k<4
   disp(' ')
   disp('Сейчас установлено')
   disp(sprintf('Безразмерная начальная скорость = %g',V0))
```

```
   disp(sprintf('Шаг интегрирования = %g',Ts))
   disp(sprintf('Предел интегрирования = %g',Tmax))
   k=menu('   Что изменить ?   ',...
      sprintf('Безразмерная начальная скорость = %g',V0),...
      sprintf('Шаг интегрирования = %g',Ts),...
      sprintf('Предел интегрирования = %g',Tmax),...
      'Начать вычисления');
   disp(' ')
   if k==1,
      V0=input([sprintf('Текущее значение V0 = %g', V0) '...
                        Новое значение V0= ']);
   elseif k==2,
      Ts=input([sprintf('Текущее значение Ts = %g', Ts) '...
                        Новое значение Ts= ']);
   elseif k==3,
      Tmax=input([sprintf('Текущее значение Tmax = %g', Tmax) '...
                              Новое значение Tmax= ']);
   end;
end;

% Листинг файла dipol_zastavka.m
% Ввод исходных значений

global M
M=[0 sin(11.5*pi/180) cos(11.5*pi/180)]; % M - магнитный
                                          % момент диполя
V0=1.55;
Ts=2*10^-4;
Tmax=13;
% Вывод информационного сообщения
clc
disp('Программа для моделирования движения заряженной...
       частицы в поле магнитного диполя,')
disp('параметры которого соответствуют магнитному полю Земли.')
disp('Векторные безразмерные уравнения движения частицы...
      имеют вид')
disp('                      dR/dt=V')
disp('                      dV/dt=2*pi*[V,H(R)]')
disp('Начальные условия: R(0)=(1,0,0); v(0)=(0,v0,0)')
disp('Ts - шаг интегрирования,Tmax - предел интегрирования.')
disp('Значения V0,Ts и Tmax можно изменять непосредственно...
      в диалоговом режиме')
disp('значения других параметров - в программе dipol_main.')
disp('Для исследования особенностей движения частиц...
      в поле магнитного диполя рекомендуется проанализировать')
```

```
disp('результаты работы программы при различных значениях V0')
disp('Отметим, что при V0 > 1.54 наступает...
      ХАОТИЧЕСКИЙ РЕЖИМ движения частицы.')

% Листинг файла dipol.m
% Пакет для автоматизации моделирования движения частицы
% в поле магнитного диполя.
% Программа dipol_zastavka выводит сведения о пакете,
% список управляющих параметров и их физический смысл.
% dipol_menu позволяет изменять управляющие параметры
% в диалоговом режиме.
% dipol_main.m осуществляет вычисления.

global M
clc
dipol_zastavka;
pause(0.5)
dipol_menu;
dipol_main;
```

Рис. 6.11. Вид монитора после выполнения команды `dipol`

Запуск на выполнение описанных выше модулей осуществляется выполнением команды `dipol` в командном окне MATLAB, после чего появляется окно (рис. 6.11),

в котором можно либо изменить значения параметров модели, установленные по умолчанию, либо начать вычисления.

Непосредственным результатом работы программы являются зависимости соответствующих декартовых координат $x = x(t), y = y(t), z = z(t)$ и скоростей движения заряда от времени $v_x = x'(t), v_y = y'(t), v_z = z'(t)$, совокупность которых задает шестимерное фазовое пространство. Наличие зависимостей $v_x = x'(t)$, $v_y = y'(t)$, $v_z = z'(t)$ позволяет провести вычисление значений параметра κ, характеризующего неоднородность магнитного поля, для каждой точки траектории, а также значений любого из адиабатических инвариантов. Для анализа особенностей движения заряда оказывается более удобным использовать зависимость радиуса-вектора заряда от времени $\left(R(t) = \sqrt{x(t)^2 + y(t)^2 + z(t)^2}\right)$. Это позволяет уменьшить размерность задачи и проводить исследование поведения динамической системы не в шестимерном, а в двумерном фазовом пространстве. Здесь для анализа движения заряда можно использовать:

1) зависимость радиуса-вектора от времени $R = R(t)$;

2) фазовую траекторию (зависимость $R' = R(t)$);

3) траекторию движения в трехмерном пространстве;

4) нормированную спектральную плотность мощности зависимости $R = R(t)$;

5) зависимость, описывающую изменение полной энергии заряда во времени, $E = E(t)$ — величина, характеризующая точность выполнения закона сохранения энергии;

6) зависимость магнитного момента μ, вычисляемого в соответствии с (6.80), от времени;

7) зависимость коэффициента неоднородности магнитного поля κ, вычисляемого в соответствии с (6.79), от времени;

8) зависимость расстояния между траекториями, близкими в начальный момент времени, от времени (данная характеристика позволяет судить об особенностях поведения рассматриваемой динамической системы [16], в частности определять факт перехода динамической системы в хаотический режим).

Для построения последней зависимости задаются близкие начальные условиями $(R(0) - R1(0) \ll 1)$, находится численное решение системы ДУ (6.76) для обоих начальных условий, далее для каждого момента времени вычисляют разность между соответствующими значениями радиусов-векторов первой и второй траекторий.

Результаты работы программы представлены для начальных скоростей

$$\vec{\tilde{v}}_0 = (0, 1{,}54, 0),$$

$$\vec{\tilde{v}}_0 = (0, 0{,}5, 0).$$

представлены на рис. 6.12—6.13 и 6.14—6.15 соответственно.

Рис. 6.12. Результаты моделирования (слева направо и сверху вниз):
1) зависимость разности между начальным и текущими значениями кинетической энергии заряда от времени; 2) зависимость магнитного момента заряда от времени; 3) зависимость коэффициента неоднородности поля от времени; 4) разность модулей радиусов-векторов зарядов, находившихся в начальный момент времени в близких точках.

Рис. 6.13. Результаты моделирования (слева направо и сверху вниз):
1) зависимость модуля радиуса-вектора от времени; 2) зависимость скорости изменения модуля радиуса-вектора от времени; 3) спектр зависимости модуля радиуса-вектора от времени; 4) траектория движения заряда.

Рис. 6.14. Результаты моделирования (слева направо и сверху вниз):
1) зависимость модуля радиуса-вектора от времени; 2) зависимость скорости изменения модуля радиуса-вектора от времени; 3) спектр зависимости модуля радиуса-вектора от времени; 4) траектория движения заряда.

Рис. 6.15. Результаты моделирования (слева направо и сверху вниз):
1) зависимость разности между начальным и текущими значениями кинетической энергии заряда от времени; 2) зависимость магнитного момента заряда от времени; 3) зависимость коэффициента неоднородности поля от времени; 4) разность модулей радиусов-векторов зарядов, находившихся в начальный момент времени в близких точках.

Проведем анализ полученных результатов. Из рис. 6.12, 6.14 по зависимостям $R = R(t)$, полученным при различных значениях безразмерной начальной скорости и фиксированной в пространстве ориентации радиуса-вектора, и нормированным амплитудно-частотным спектрам (далее спектрам) зависимостей $R = R(t)$ видно, что функция $R = R(t)$ является некоторой периодической функцией, амплитуда которой зависит от величины безразмерной начальной скорости. Эти результаты показывают, что траектория движения заряда в неоднородном магнитном поле, создаваемом магнитным диполем, в координатном пространстве является финитным квазипериодическим движением: заряд движется по поверхности тора, радиус большой окружности тора равен

$$0,5(R_{\max} + R_{\min}),$$

радиус малой окружности тора —

$$0,5(R_{\max} - R_{\min}).$$

Увеличение значения абсолютной скорости приводит к увеличению радиусов большой и малой окружностей тора. Данный вывод справедлив также для 6-мерного фазового пространства, в котором движение заряда является квазипериодическим движением по поверхности 6-мерного тора. Этот результат понятен: существование для каждого начального условия тороидальной поверхности является следствием сохранения энергии заряда, движущегося в постоянном магнитном поле.

Из рис. 6.13 также видно, что расстояние между траекториями, близкими в начальный момент времени, оказывается случайной функцией. В сочетании с результатами, представленными на рис. 6.12 (в первую очередь, вид спектра, который стремится к непрерывному спектру), можно сделать вывод о том, что при начальной скорости движения заряда

$$\vec{v}_0 = (0, 1{,}54, 0)$$

движение электрического заряда в поле магнитного диполя является хаотическим.

***Задача* 6.12**

1. *Сравните точность сохранения адиабатического инварианта* (6.80) *в нехаотическом и хаотическом режимах.*
2. *Сравните точность выполнения условия слабого поля* (6.79) *в нехаотическом и хаотическом режимах.*

Представляет интерес провести более подробный анализ процесса перехода рассматриваемой динамической системы от детерминированного к хаотическому режиму, проведя расчеты для различных значений начальной скорости. Выбранные значения начальной скорости, а также соответствующие значения энергии частицы, максимальные значения коэффициента неоднородности магнитного поля, значение параметра, характеризующего степень сохранения адиабатического инварианта

$$A_i = 2\frac{I_{1\max,i} - I_{1\min,i}}{\left(I_{1\max,i} + I_{1\min,i}\right)}, i = \overline{1,10}$$

представлены в таблице 6.1.

Таблица 6.1. Результаты исследования особенностей движения электрического заряда в поле магнитного диполя

	$\lvert\vec{v}\rvert/c$	E, *Мэв*	κ	A,%	ΔR
1	$0,3 \cdot 10^{-2}$	$4,22 \cdot 10^{-3}$	$2,0 \cdot 10^{-3}$	1,6	0,014
2	$0,6 \cdot 10^{-2}$	$16,9 \cdot 10^{-3}$	$4,3 \cdot 10^{-3}$	3,1	0,019
3	$1,2 \cdot 10^{-2}$	$67,5 \cdot 10^{-2}$	$1,3 \cdot 10^{-2}$	6,4	0,028
4	$2,4 \cdot 10^{-2}$	$0,27$	$2,3 \cdot 10^{-2}$	12,1	0,045
5	$4,8 \cdot 10^{-2}$	1,08	$8,01 \cdot 10^{-2}$	23,5	0,078
6	$7,0 \cdot 10^{-2}$	2,30	$1,3 \cdot 10^{-1}$	33,4	0,106
7	$9,0 \cdot 10^{-2}$	3,80	$1,7 \cdot 10^{-1}$	42,1	0,129
8	$1,5 \cdot 10^{-1}$	10,55	$3,1 \cdot 10^{-1}$	68,4	0,190
9	$2,0 \cdot 10^{-1}$	18,76	$4,4 \cdot 10^{-1}$	82,9	0,228
10	$3,0 \cdot 10^{-1}$	44,21	1,13	199,7	0,406

Анализ результатов, представленных в табл. 6.1, показывает, что при увеличении безразмерной скорости движения заряда, достигаемого увеличением абсолютного значения скорости при неизменных характеристиках магнитного поля, происходит монотонное увеличение коэффициента κ при одновременном разрушении адиабатического инварианта $v_\perp^2/\left|\vec{H}\right|$. При этом оказывается, что точность сохранения адиабатического инварианта уже при значениях $\kappa \approx 9 \cdot 10^{-2}$, (то есть удовлетворяющих, по устоявшемуся мнению, условию применимости дрейфового приближения $\kappa \ll 1$) составляет не более 23,5%. Следовательно, использование модели сохраняющихся адиабатических инвариантов для описания движения в магнитном поле Земли протонов с энергией более 2 *Мэв* неправомерно. Более того, как будет показано ниже, даже для протонов, имеющих значительно меньшие кинетические энергии, данная модель не позволяет выявить ряд особенностей движения электрических зарядов в поле магнитного диполя.

Рассмотрим более подробно трансформацию зависимостей $R = R(t)$, представленных на рис. 6.16–6.17, фазовых траекторий, представленных на рис. 6.20–6.21, и спектров зависимостей $R = R(t)$, представленных на рис. 6.18–6.19, при последовательном увеличении значений безразмерной начальной скорости.

При $\vec{\tilde{v}}_0 = (0, 1.5 \cdot 10^{-2}, 0)$ ($\kappa = 2,1 \cdot 10^{-3}$, $A = 1,6\%$) движение частицы согласуется с моделью сохраняющихся адиабатических инвариантов: вращение частицы по ларморовской окружности, движения центра ларморовской окружности вдоль силовой линии, дрейфа центра ларморовской окружности в плоскости *XOY*(рис. 6.16). Отметим, не касаясь деталей, что здесь и далее на рис. 6.16 вследствие особенностей цифровых методов спектрального анализа, подробно рассмотренных в [17], при выбранных длительностях временного интервала для поиска решения системы уравнений движения (6.76) в спектре функции $R = R(t)$ не удается разрешить гармоники, соответствующие наиболее медленному из перечисленных выше движений — дрейфу

Рис. 6.16. Зависимость $R = R(t)$:
1) $\vec{v}_0 = (0, 1{,}5 \cdot 10^{-2}, 0)$;
2) $\vec{v}_0 = (0, 6 \cdot 10^{-2}, 0)$;
3) $\vec{v}_0 = (0, 1{,}2 \cdot 10^{-1}, 0)$;
4) $\vec{v}_0 = (0, 2{,}4 \cdot 10^{-1}, 0)$.

Рис. 6.17. Зависимость $R = R(t)$:
1) $\vec{v}_0 = (0, 3{,}5 \cdot 10^{-1}, 0)$;
2) $\vec{v}_0 = (0, 4{,}5 \cdot 10^{-1}, 0)$;
3) $\vec{v}_0 = (0, 7{,}5 \cdot 10^{-1}, 0)$;
4) $\vec{v}_0 = (0, 1{,}5, 0)$.

Рис. 6.18. Спектр зависимости $R = R(t)$:
1) $\vec{v}_0 = (0, 1{,}5 \cdot 10^{-2}, 0)$;
2) $\vec{v}_0 = (0, 6 \cdot 10^{-2}, 0)$;
3) $\vec{v}_0 = (0, 1{,}2 \cdot 10^{-1}, 0)$;
4) $\vec{v}_0 = (0, 2{,}4 \cdot 10^{-1}, 0)$.

Рис. 6.19. Спектр зависимости $R = R(t)$:
1) $\vec{v}_0 = (0, 3{,}5 \cdot 10^{-1}, 0)$;
2) $\vec{v}_0 = (0, 4{,}5 \cdot 10^{-1}, 0)$;
3) $\vec{v}_0 = (0, 7{,}5 \cdot 10^{-1}, 0)$;
4) $\vec{v}_0 = (0, 1{,}5, 0)$.

Рис. 6.20. Зависимость $R' = f(R)$:
1) $\vec{v}_0 = (0, 1{,}5 \cdot 10^{-2}, 0)$;
2) $\vec{v}_0 = (0, 6 \cdot 10^{-2}, 0)$;
3) $\vec{v}_0 = (0, 1{,}2 \cdot 10^{-1}, 0)$;
4) $\vec{v}_0 = (0, 2{,}4 \cdot 10^{-1}, 0)$.

Рис. 6.21. Зависимость $R' = f(R)$:
1) $\vec{v}_0 = (0, 3{,}5 \cdot 10^{-1}, 0)$;
2) $\vec{v}_0 = (0, 4{,}5 \cdot 10^{-1}, 0)$;
3) $\vec{v}_0 = (0, 7{,}5 \cdot 10^{-1}, 0)$;
4) $\vec{v}_0 = (0, 1{,}5, 0)$.

частицы в плоскости XOY, поскольку на выбранном временном интервале укладывается менее одного периода дрейфового движения. Указанный недостаток легко устранить, увеличив при неизменном шаге интегрирования временной интервал, на котором ищется решение системы уравнений движения (6.76). Однако этого не требуется, поскольку скорость дрейфового движения при выбранных начальных условиях ($v_{||} \ll v_{\perp} \cong |\vec{v}_0|$) оказывается значительно меньше скоростей ларморовского вращения и движения центра ларморовской окружности, в чем легко убедиться, записав (6.67) в безразмерном виде и учтя, что при выбранных начальных условиях

$$\tilde{u}_d = \frac{\kappa}{\tilde{v}_{\perp}} \left(\frac{\tilde{v}_{\perp}^2}{2} + \tilde{v}_{||}^2 \right) \approx k\tilde{v}, \tag{6.81}$$

где $\tilde{v}_{||} = v_{||}/\left(R_0/T_0\right)$.

Из (6.81) видно, что влиянием смещения, обусловленного дрейфом заряда, на особенности движения заряда в первом приближении можно пренебречь.

Анализ зависимости $R = R(t)$ и соответствующей фазовой траектории при $\vec{\tilde{v}}_0 = (0, 1{,}5 \cdot 10^{-2}, 0)$ позволяет сделать вывод о существовании у системы (6.76) простого или регулярного аттрактора [15]. При этом необходимо отметить особенность движения, не выявляемую при использовании модели сохраняющихся адиабатических инвариантов. В спектре функции $R = R(t)$ помимо гармоник, соответствующих движению заряда вдоль силовых линий (f_1) и вращения заряда по ларморовской окружности (f_2), присутствуют две дополнительные гармоники с частотами $f_2 \pm f_1$. Их существование обусловлено взаимодействием рассматриваемых движений: перемещение заряда вдоль силовой линии приводит к изменению скорости вращения по ларморовской окружности вследствие изменения напряженности магнитного поля.

Для $\vec{\tilde{v}}_0 = (0, 6 \cdot 10^{-2}, 0)$ ($\kappa \approx 1,3 \cdot 10^{-2}$, $A \approx 6,4\%$) при сохранении в целом регулярности аттрактора происходит усложнение движения: во-первых, появляется двоенная вторая гармоника ($2f_2$), соответствующая вращению заряда по ларморовской окружности, во-вторых, в спектре помимо описанных выше гармоник появляются дополнительные боковые гармоники с частотами $f_2 \pm 2f_1$, $2f2 \pm f1$, $2f2 \pm 2f1$ (рис. 6.20). Данный результат обусловлен увеличением «степени нелинейности» рассматриваемой динамической системы: увеличение начальной скорости заряда приводит к более быстрому смещению в область сильно неоднородного магнитного поля, и, как следствие, большему взаимовлиянию рассматриваемых движений друг на друга.

При $\vec{\tilde{v}}_0 = (0, 1.2 \cdot 10^{-1}, 0)$ ($\kappa \approx 2.3 \cdot 10^{-2}, A \approx 12,1\%$) в спектре появляется дополнительная гармоника с частотой, равной утроенной частоте вращения заряда по ларморовской окружности и боковых гармоник, имеющих частоты $3f_1 \pm f_2$. Дальнейшее увеличение начальной скорости приводит к увеличению амплитуды данных гармоник (см. рис. 6.18). При начальной скорости при $\vec{\tilde{v}}_0 = (0, 4{,}5 \cdot 10^{-1}, 0)$ ($\kappa \approx 1,25 \cdot 10^{-1}, A \approx 33,4\%$) в спектре появляются гармоники, имеющие частоты $4f_1 \pm f_2$, при дальнейшем увеличении начальной скорости повторяется описанный сценарий (см. рис. 6.18–6.19).

6.6. МОДЕЛИРОВАНИЕ ДВИЖЕНИЯ ЗАРЯДА В СЛОЖНЫХ ЭЛЕКТРОСТАТИЧЕСКИХ ПОЛЯХ

Примером еще одной детерминированной системы, в которой при определенных условиях может возникать динамический хаос, является электрический заряд, движущийся в электростатическом поле, создаваемом двумя неподвижными электрическими зарядами (рис. 6.22). Будем для определенности считать, что движущийся электрический заряд является отрицательным, а электрические заряды, создающие электростатическое поле, — положительными.

Рис. 6.22. К построению математической модели, описывающей движение электрического заряда в электростатическом поле, создаваемом двумя неподвижными электрическими зарядами

В соответствии с законом Кулона (6.5) и принципом суперпозиции уравнение движения заряда $-q$ записывается в виде:

$$m\ddot{\vec{r}} = -\frac{1}{4\pi\varepsilon_0}\frac{qQ_1}{|\vec{r}-\vec{R}_1|^3}\left(\vec{r}-\vec{R}_1\right) - \frac{1}{4\pi\varepsilon_0}\frac{qQ_2}{|\vec{r}-\vec{R}_2|^3}\left(\vec{r}-\vec{R}_2\right), \qquad (6.82)$$

где $\vec{r} = (x, y)$ — радиус-вектор заряда $-q$, $\vec{R}_1 = (X_1, Y_1)$ — радиус-вектор заряда Q_1, $\vec{R}_2 = (X_2, Y_2)$ — радиус-вектор заряда Q_2.

Переходя к координатной записи уравнения движения заряда (6.82), получаем следующую систему ДУ второго порядка:

$$\begin{aligned} m\ddot{x} &= -\frac{1}{4\pi\varepsilon_0}\frac{qQ_1}{|\vec{r}-\vec{R}_1|^3}(x-X_1) - \frac{1}{4\pi\varepsilon_0}\frac{qQ_2}{|\vec{r}-\vec{R}_2|^3}(x-X_2), \\ m\ddot{y} &= -\frac{1}{4\pi\varepsilon_0}\frac{qQ_1}{|\vec{r}-\vec{R}_1|^3}(y-Y_1) - \frac{1}{4\pi\varepsilon_0}\frac{qQ_2}{|\vec{r}-\vec{R}_2|^3}(y-Y_2). \end{aligned} \qquad (6.83)$$

Предваряя дальнейшее решение задачи, напомним, что условие движения по круговой орбите заряда $-q$ в электростатическом поле, создаваемом неподвижным электрическим зарядом, записывается в виде:

$$m\frac{v_0^2}{R_0} = \frac{1}{4\pi\varepsilon_0}\frac{qQ_1}{R_0^2}. \qquad (6.84)$$

Из (6.84) находим линейную скорость кругового движения заряда $-q$:

$$v_0 = \sqrt{\frac{1}{4\pi\varepsilon_0}\frac{qQ_1}{mR_0}}, \tag{6.85}$$

тогда период его движения T_0 равен

$$T_0 = \frac{2\pi R_0}{v_0} = 2\pi R_0^{3/2} \Big/ \sqrt{\frac{qQ_1}{4\pi\varepsilon_0 m}}. \tag{6.86}$$

Проведем обезразмеривание уравнения движения (6.82). Для этого выберем в качестве единиц измерения расстояния R_0 — радиус круговой орбиты заряда $-q$, движущегося в электростатическом поле, создаваемом электрическим зарядом Q_1, в качестве единиц измерения времени T_0 — период обращения по рассматриваемой орбите. Указанный выбор единиц измерения эквивалентен использованию следующих переменных:

$$\vec{\tilde{r}} = \frac{\vec{r}}{R_0}, \quad \vec{\tilde{R}}_1 = \frac{\vec{R}_1}{R_0}, \quad \vec{\tilde{R}}_2 = \frac{\vec{R}_2}{R_0}, \quad \tilde{t} = \frac{t}{T_0}.$$

При этом безразмерная скорость движения заряда $-q$ равняется

$$\vec{\tilde{v}} = \frac{d\vec{\tilde{r}}}{d\tilde{t}} = \frac{d\left(\vec{r}/R_0\right)}{d\left(t/T_0\right)} = \frac{T_0}{R_0}\frac{d\vec{r}}{dt},$$

а безразмерное ускорение

$$\vec{\tilde{a}} = \frac{d^2\vec{\tilde{r}}}{d\tilde{t}^2} = \frac{d^2\left(\vec{r}/R_0\right)}{d(t/T_0)^2} = \frac{T_0^2}{R_0}\frac{d\vec{\tilde{r}}}{d\tilde{t}}.$$

Отметим, что в случае движения заряда $-q$ по окружности вокруг заряда Q_1 безразмерные значения периода обращения $\tilde{T} = 1$, безразмерное значение радиуса $\tilde{R}=1$, модуль безразмерного значения скорости $|\tilde{v}|_c$:

$$|\tilde{v}|_c = \frac{T_0}{R_0}\sqrt{\frac{1}{4\pi\varepsilon_0}\frac{qQ_1}{mR_0}} = \frac{2\pi R_0^{3/2}}{\sqrt{qQ_1/4\pi\varepsilon_0 m}}\frac{1}{R_0}\sqrt{\frac{1}{4\pi\varepsilon_0}\frac{qQ_1}{mR_0}} = 2\pi.$$

Заменяя в (6.82) размерные переменные безразмерными, получаем

$$m\frac{R_0}{T_0^2}\frac{d^2\vec{\tilde{r}}}{d\tilde{t}^2} = -\frac{1}{4\pi\varepsilon_0}\frac{qQ_1R_0}{R_0^3}\frac{\left(\vec{\tilde{r}} - \vec{\tilde{R}}_1\right)}{\left|\vec{\tilde{r}} - \vec{\tilde{R}}_1\right|^3} - \frac{1}{4\pi\varepsilon_0}\frac{qQ_2R_0}{R_0^3}\frac{\left(\vec{\tilde{r}} - \vec{\tilde{R}}_2\right)}{\left|\vec{\tilde{r}} - \vec{\tilde{R}}_2\right|^3},$$

откуда

$$\frac{d^2\vec{\tilde{r}}}{d\tilde{t}^2} = \frac{4\pi^2\left(\vec{\tilde{r}} - \vec{\tilde{R}}_1\right)}{\left|\vec{\tilde{r}} - \vec{\tilde{R}}_1\right|^3} - \frac{4\pi^2 K\left(\vec{\tilde{r}} - \vec{\tilde{R}}_2\right)}{\left|\vec{\tilde{r}} - \vec{\tilde{R}}_2\right|^3}. \tag{6.87}$$

Подставляя (6.86) в (6.87), окончательно получаем

$$\frac{d^2\vec{\tilde{r}}}{d\tilde{t}^2} = \frac{4\pi^2\left(\vec{\tilde{r}} - \vec{\tilde{R}}_1\right)}{\left|\vec{\tilde{r}} - \vec{\tilde{R}}_1\right|^3} - \frac{4\pi^2 K\left(\vec{\tilde{r}} - \vec{\tilde{R}}_2\right)}{\left|\vec{\tilde{r}} - \vec{\tilde{R}}_2\right|^3}, \tag{6.88}$$

где $K = Q_2/Q_1$.

Соответственно, система ДУ (6.83) в безразмерном виде записывается как

$$\begin{aligned} \ddot{\tilde{x}} &= -\frac{4\pi^2\left(\tilde{x} - \tilde{X}_1\right)}{\left|\vec{\tilde{r}} - \vec{\tilde{R}}_1\right|^3} - \frac{4\pi^2 K\left(\tilde{x} - \tilde{X}_2\right)}{\left|\vec{\tilde{r}} - \vec{\tilde{R}}_2\right|^3}, \\ \ddot{\tilde{y}} &= -\frac{4\pi^2\left(\tilde{y} - \tilde{Y}_1\right)}{\left|\vec{\tilde{r}} - \vec{\tilde{R}}_1\right|^3} - \frac{4\pi^2 K\left(\tilde{y} - \tilde{Y}_2\right)}{\left|\vec{\tilde{r}} - \vec{\tilde{R}}_2\right|^3}, \end{aligned} \tag{6.89}$$

где $\tilde{X}_1 = X_1/R_0$, $\tilde{Y}_1 = Y_1/R_0$, $\tilde{X}_2 = X_2/R_0$,$\tilde{Y}_2 = Y_2/R_0$.

Движение заряда $-q$ в рассматриваемом электростатическом поле можно рассматривать как движение частицы в потенциале:

$$U(r) = -\frac{1}{4\pi\varepsilon_0}\frac{qQ_1}{\left|\vec{r} - \vec{R}_1\right|} - \frac{1}{4\pi\varepsilon_0}\frac{qQ_2}{\left|\vec{r} - \vec{R}_2\right|}, \tag{6.90}$$

который, используя безразмерные переменные, можно записать в виде

$$U\left(|\vec{\tilde{r}}|\right) = -U_0\left[\frac{1}{\left|\vec{\tilde{r}} - \vec{\tilde{R}}_1\right|} - \frac{K}{\left|\vec{\tilde{r}} - \vec{\tilde{R}}_2\right|}\right], \tag{6.91}$$

где

$$U_0 = \frac{qQ_1}{4\pi\varepsilon_0 R_0}. \tag{6.92}$$

Из (6.91) видно, что в качестве единиц измерения потенциала оказывается удобным использовать величину. Тогда безразмерный потенциал вводится как

$$\tilde{U}(\tilde{r}) = \frac{U(\tilde{r})}{U_0} = \left[-\frac{1}{\left|\vec{\tilde{r}} - \vec{\tilde{R}}_1\right|} - \frac{K}{\left|\vec{\tilde{r}} - \vec{\tilde{R}}_2\right|}\right]. \tag{6.93}$$

Получим выражение для безразмерной потенциальной энергии

$$\tilde{E}_k = \frac{mv^2}{2}\Big/U_0 = \frac{(R_0/T_0)^2}{U_0}\frac{m\tilde{v}^2}{2}. \tag{6.94}$$

Подставляя (6.92), (6.86) в (6.94), окончательно получаем

$$\tilde{E}_k = \frac{\tilde{v}^2}{8\pi^2}. \tag{6.95}$$

Выражения (6.93), (6.95) сразу позволяют записать выражение для полной безразмерной энергии заряда, являющейся интегралом движения:

$$E = E_k + \tilde{U} = \frac{\tilde{v}^2}{8\pi^2} - \frac{1}{\left|\vec{\tilde{r}} - \vec{\tilde{R}}_1\right|} - \frac{K}{\left|\vec{\tilde{r}} - \vec{\tilde{R}}_2\right|}. \tag{6.96}$$

Для обоснованного выбора граничных условий, предваряя нахождение численного решения системы ДУ (6.89), исследуем потенциал (6.93), в котором движется заряд $-q$. Для его визуализации необходимо создать файл `U2q.m`, листинг которого приведен ниже:

```
% листинг файла U2q.m
function z=U2q(x,y)
% функция, возвращающая значения
% потенциальной энергии
% рассматриваемой системы

global X1 Y1 X2 Y2 Q1 Q2
K=Q2/Q1;
Nx=size(x,2);
Ny=size(y,2);

z=zeros(Ny,Nx);
for i=1:Ny
    for j=1:Nx
        z(i,j)=-1/((x(j)-X1)^2+...
               (y(i)-Y1)^2)^0.5-...
               K/((x(j)-X2)^2+...
               (y(i)-Y2)^2)^0.5;
    end;
end;
```

Далее нужно выполнить следующую последовательность команд, сохраненную нами в файле `Glava6_4.m`:

```
% листинг файла Glava6_4.m
global X1 Y1 X2 Y2 Q1 Q2
% задание координат радиуса-вектора
% заряда Q1
X1=0;
Y1=0;

% задание координат радиуса-вектора
% заряда Q2
X2=2;
Y2=0;
```

```
Q1=2; % значение первого заряда
Q2=1; % значение второго заряда

% задание координатной сетки
x1=X1-1;
y1=Y1-1;
x2=X2+1;
y2=Y2+1;
Nx=56; % число узлов по оси OX
kx=1:Nx;
dx=(x2-x1)/(Nx-1);
x=x1+dx*(kx-1);

Ny=56; % число узлов по оси OY
ky=1:Ny;
dy=(y2-y1)/(Ny-1);
y=y1+dy*(ky-1);

M=U2q(x,y); % вычисление значений
            % потенциала в узлах координатной
            % сетки

% визуализация потенциальной
% энергии
[X Y]=meshgrid(x,y);
surf(X,Y,M)
view([-14 48])

% визуализация карты эквипотенциалей
% потенциальной энергии
% рассматриваемой системы
figure
contourf(X,Y,M,55);

% визуализация сечения
% потенциала плоскостью y=0
figure
plot(x,M(28,:))
```

Результаты выполнения описанной выше последовательности команд представлены на рис. 6.23–6.25.

Из рис. 6.25 видно, что в зависимости от полной энергии заряда $-q$ и его начальных координат движение заряда может быть:

1) простым финитным в окрестности заряда Q_1;

2) простым финитным в окрестности заряда Q_2;

Рис. 6.23. Поверхность потенциала, создаваемого системой двух зарядов

Рис. 6.24. Карта эквипотенциалей потенциала системы, состоящей из двух зарядов

Рис. 6.25. Сечение потенциала системы, состоящей из двух зарядов, плоскостью $x = 0$

3) сложным финитным, когда траектория движения проходит как в окрестности первого, так и второго зарядов;

4) инфинитным.

Для нахождения численного решения системы ДУ (6.89) необходимо привести данную систему к эквивалентной системе ДУ первого порядка:

$$\begin{gathered}\dot{z}_1 = z_2,\\ \dot{z}_2 = -\frac{4\pi^2\left(z_1-\tilde{X}_1\right)}{\left|\vec{\tilde{r}}-\vec{\tilde{R}}_1\right|^3} - \frac{4\pi^2 K\left(z_1-\tilde{X}_2\right)}{\left|\vec{\tilde{r}}-\vec{\tilde{R}}_2\right|^3},\\ \dot{z}_3 = z_4,\\ \dot{z}_4 = -\frac{4\pi^2\left(z_3-\tilde{Y}_1\right)}{\left|\vec{\tilde{r}}-\vec{\tilde{R}}_1\right|^3} - \frac{4\pi^2 K\left(z_3-\tilde{Y}_2\right)}{\left|\vec{\tilde{r}}-\vec{\tilde{R}}_2\right|^3}.\end{gathered} \tag{6.97}$$

Для нахождения численного решение системы ДУ (6.97) необходимо создать файл `D2q.m`, содержащий описание функции, возвращающей значения вектора-функции, стоящей в правой части (6.97). Ниже приведен листинг данной функции.

```
% листинг функции D2q.m
function dz=D2q(t,z)
% функция, возвращающая значение
% вектора-функции,
% стоящего в правой части (6.97)

global X1 Y1 X2 Y2 Q1 Q2
dz=zeros(4,1);
K=Q2/Q1;
R1=((z(1)-X1)^2+(z(3)-Y1)^2)^0.5;
R2=((z(1)-X2)^2+(z(3)-Y2)^2)^0.5;
dz(1)=z(2);
dz(2)=-4*pi^2*(z(1)-X1)/R1^3-...
       4*pi^2*K*(z(1)-X2)/R2^3;
dz(3)=z(4);
dz(4)=-4*pi^2*(z(3)-Y1)/R1^3-...
      4*pi^2*K*(z(3)-Y2)/R2^3;
```

Далее необходимо выполнить следующую последовательность команд, сохраненную нами в файле `Glava6_5.m`:

```
% листинг файла Glava6_5.m
global X1 Y1 X2 Y2 Q1 Q2

% задание координат
```

```
% радиуса-вектора заряда
X1=0;
Y1=0;

% задание координат
% радиуса-вектора заряда
X2=2;
Y2=0;

Q1=2; % значение первого заряда
Q2=1; % значение второго заряда

% задание начальных условий

% задание координат радиуса-вектора
% заряда -q в момент времени t=0
X0=-1;
Y0=0;

% задание кинетической энергии заряда -q
dW=0.5;

% вычисление начальной скорости
% движения заряда -q
V=pi*(8*dW)^0.5;

% задание координат вектора скорости
% заряда -q в момент времени t=0
Vx=0;
Vy=V;

% вычисление  значения энергии
% заряда -q в момент времени t=0
E0=U2q(X0,Y0)+dW;

% решение системы ДУ (6.97)
options = odeset('RelTol',1e-5,...
                 'AbsTol',[1e-5 1e-5 1e-5 1e-5]);
[t R]=ode45('D2q',...
            [0:1*10^-3:2],...
            [X0 Vx Y0 Vy],...
            options);

% визуализация траектории движения
% заряда -q
```

```
figure
plot(R(:,1),R(:,3),X0,Y0,'x',X1,Y1,'o',X2,Y2,'o');
axis([-2 4 -1 1.5]);

% вычисление зависимости
% полной энергии
% заряда -q от времени
r1=((R(:,1)-X1).^2+(R(:,3)-Y1).^2).^0.5;
r2=((R(:,1)-X2).^2+(R(:,3)-Y2).^2).^0.5;
K=Q2/Q1;
Ep=-1./r1-K./r2;
Ek=(R(:,2).^2+R(:,4).^2)/(8*pi^2);
E=Ep+Ek;

% визуализация зависимости
% погрешности полной энергии
% заряда от времени
figure;
plot(t,abs(E-E0)/E0*100);
```

Результаты выполнения приведенной выше последовательности команд представлены на рис. 6.26–6.27.

Рис. 6.26. Траектория движения заряда $-q$ в электростатическом поле, создаваемом двумя неподвижными зарядами (x — начальное положение заряда $-q$, ∘ — положения зарядов $Q1$, $Q2$)

Рис. 6.27. Зависимость погрешности полной энергии заряда $-q$ от времени

Из рис. 6.26 видно, что при выбранных начальных условиях траектория заряда $-q$ охватывает оба заряда. Из рис. 6.27 видно, что отклонение полной энергии заряда от точного значения не превышает $0,6 \cdot 10^{-3}$, что свидетельствует о достаточно высокой точности полученного численного решения.

***Задача* 6.13**

1. *Объясните, используя рис. 6.25, полученную выше форму траектории.*
2. *Постройте соответствующие графики и проведите анализ зависимостей координат заряда $-q$ и проекций скорости на координатные оси от времени.*
3. *Постройте зависимость радиуса-вектора от времени $R = R(t)$.*
4. *Вычислите зависимость скорости изменения радиуса-вектора заряда $-q$ от времени $\dot{R} = dR/dt$.*
5. *Постройте и проанализируйте зависимость $\dot{R} = dR/dt$.*

***Задача* 6.14**

1. *Получите численные решения системы ДУ* (6.97) *для двух точек, близких друг к другу в момент времени*
2. *Вычислите зависимость расстояния между данными траекториями от времени и проведите анализ данной зависимости.*

***Задача* 6.15**

1. *Выберете, используя зависимость, представленную на рис. 6.25, начальные условия так, чтобы движение заряда $-q$ было:*
   - *финитным и траектория охватывала только заряд Q_1;*
   - *финитным и траектория движения охватывала только заряд Q_2;*
   - *инфинитным.*
2. *Проверьте правильность выбора начальных условий, выполнив соответствующие численные расчеты.*
3. *Для каждого из рассмотренных случаев движения заряда $-q$:*
   - *постройте соответствующие графики и проведите анализ зависимостей координат заряда $-q$ и проекций скорости на координатные оси от времени;*
   - *постройте зависимость радиуса-вектора от времени $R = R(t)$;*
   - *вычислите зависимость скорости изменения радиуса-вектора заряда $-q$ от времени $\dot{R} = dR/dt$;*
   - *постройте и проанализируйте зависимость $\dot{R} = dR/dt$.*

***Задача* 6.16**

1. *Получите численные решения системы ДУ* (6.97) *для найденных в предыдущей задаче начальных условий для двух точек, близких друг к другу в момент времени $t = 0$.*

2. *Вычислите зависимости расстояния между данными траекториями от времени и проведите сравнительный анализ данных зависимостей.*

ЛИТЕРАТУРА К ГЛАВЕ 6

1. Ландау Л. Д., Лифшиц Е. М. Механика. М.: Наука, 1958.
2. Носач В. В. Решение задач аппроксимации с помощью персональных компьютеров. М.: МИКАП, 1994.
3. Ландау Л. Д., Лифшиц Е. М. Теория поля. М.: Наука, 1988.
4. Угаров В. А. Специальная теория относительности. М.: Наука, 1977.
5. Арцимович Л. А., Лукьянов С. Б. Движение заряженных частиц в электрических и магнитных полях. М.: Наука, 1978.
6. Арцимович Л. А. Замкнутые плазменные конфигурации. М.: Наука, 1969.
7. Арцимович Л. А., Сагдеев Р. З. Физика плазмы для физиков. М.: Атомиздат, 1979.
8. Альвен Х. Космическая электродинамика. М.: Изд-во иностранной литературы, 1952.
9. Пилькенер С. Б. Основы космической электродинамики. М.: Физматгиз, 1961.
10. Ландау Л. Д., Лифшиц Е. М. Теория поля. М.: Физматлит, 1975.
11. Заславский Г. М., Сагдеев Р. З. Введение в нелинейную физику: От маятника до турбулентности и хаоса. М.: Наука, 1988.
12. Родионов М. С. // Атомная энергия. 1959. Т. 6. С. 623.
13. Арнольд В. И. // Доклады АН СССР. 1962. Т. 142. С. 758.
14. Чириков Б. В. // Вопросы теории плазмы. М.: Энергоатомиздат, 1983. Вып. 13. С. 3.
15. Поршнев С. В. Динамическая неустойчивость движения заряженных частиц в постоянном неоднородном магнитном поле // Журнал радиоэлектроники. Интернет-журнал: http://jre.cplire.ru/nov00/5/text.html.
16. Мун Ф. Хаотические колебания. М.: Мир, 1989.
17. Марпл-мл. С. П. Цифровой спектральный анализ и его приложения. М.: Мир, 1990.

Глава 7

ФУРЬЕ-АНАЛИЗ НЕПРЕРЫВНЫХ И ДИСКРЕТНЫХ ФУНКЦИЙ

7.1. ВВЕДЕНИЕ

Одним из эффективных подходов решения задач из различных разделов физики (в первую очередь теории колебаний и волновой оптики) является подход, основанный на спектральном разложении функций, являющихся решением тех или иных уравнений (метод Фурье). В то же время в традиционных учебниках по физике использование данного метода ограничено достаточно узким кругом задач, имеющих аналитические решения. Выбранный нами в данной книге подход, основанный на нахождении численных решений уравнений, описывающих поведение физических систем, и существующая возможность вычисления спектров таблично заданных функций существенно расширяют возможности данного метода и, как следствие, позволяют повысить наглядность и физическую содержательность излагаемого материала. Отметим, что в данном разделе мы не претендуем на абсолютную математическую строгость изложения материала, которая, по мнению Р. Хэмминга, «скорее связанна с внутренними потребностями математики и эстетики, чем с нуждами физического мира» [1], но пытаемся в первую очередь показать полезность данных математических понятий для решения реальных физических задач и научить их применению на практике.

7.2. РАЗЛОЖЕНИЕ ПЕРИОДИЧЕСКИХ СИГНАЛОВ В РЯДЫ ФУРЬЕ

По определению периодической функцией называют функцию, отвечающую условию:

$$s(t) = s(t + nT), \quad n = 1, 2, \ldots \tag{7.1}$$

где T — период функции.

Для нахождения спектрального разложения функции $s(t)$ введем в рассмотрение следующие наборы функций:

$$\begin{gathered} 1/\sqrt{T},\ \sqrt{2/T}\sin(2\pi t/T),\ \sqrt{2/T}\sin(4\pi t/T), \ldots, \sqrt{2/T}\sin(2\pi n t/T), \ldots \\ \sqrt{2/T}\cos(2\pi t/T),\ \sqrt{2/T}\cos(2\pi 2t/T), \ldots, \sqrt{2/T}\cos(2\pi n t/T) \ldots \end{gathered} \tag{7.2}$$

Любая функция из (7.2), которую для краткости обозначим $u_m(t)$, удовлетворяет условию периодичности (7.1).

Рассмотрим три следующих интеграла:

$$
\begin{aligned}
&\frac{1}{T}\int_0^T \cos(2\pi mt/T)\cos(2\pi nt/T)dt = \begin{cases} 0, & \text{если} \quad m \neq n, \\ 1, & \text{если} \quad m = n \neq 0, \\ 2, & \text{если} \quad m = n = 0, \end{cases} \\
&\frac{1}{T}\int_0^T \cos(2\pi mt/T)\sin(2\pi nt/T)dt = 0, \\
&\frac{1}{T}\int_0^T \sin(2\pi mt/T)\sin(2\pi nt/T)dt = \begin{cases} 0, & \text{если} \quad m \neq n, \\ 1, & \text{если} \quad m = n \neq 0, \\ 0, & \text{если} \quad m = n = 0, \end{cases} \\
&\frac{1}{T}\int_0^T \cos(2\pi mt/T)\sin(2\pi nt/T)dt = 0, \\
&\frac{1}{T}\int_0^T \sin(2\pi mt/T)\sin(2\pi nt/T)dt = \begin{cases} 0, & \text{если} \quad m \neq n, \\ 1, & \text{если} \quad m = n \neq 0, \\ 0, & \text{если} \quad m = n = 0. \end{cases}
\end{aligned}
\tag{7.3}
$$

Функции, удовлетворяющие условию (7.3), называются ортогональными, а систему функций (7.2) называют ортонормированным базисом, образованным гармоническими функциями с кратными частотами.

В общем случае условие ортогональности базисных функций $u_i(t)$, $u_k(t)$ можно записать в компактной форме, используя символ Кронекера:

$$\int_0^T u_i(t)u_k(t)dt = \delta_{ik}, \tag{7.4}$$

где

$$\delta_{ik} = \begin{cases} 0, & \text{если} \quad i \neq k, \\ 1, & \text{если} \quad i = k. \end{cases}$$

Разложим произвольную периодическую функцию $s(t)$ в ряд по системе базисных функций $u_i(t)$:

$$s(t) = \sum_{i=0}^{\infty} c_i u_i(t). \tag{7.5}$$

Представление (7.5) называется обобщенным рядом Фурье функции $s(t)$ в выбранном базисе. Коэффициенты данного ряда находятся умножением (7.5) на базисную функцию $u_k(t)$ и интегрированием по периоду функции $s(t)$:

$$\int_0^T s(t)u_k(t)dt = \sum_{i=0}^{\infty} c_i \int_0^T u_i(t)u_k(t)dt. \tag{7.6}$$

Откуда, используя свойство ортоноромированности (7.4), найдем

$$c_k = \int_0^T s(t)u_k(t)dt, \tag{7.7}$$

Подставляя в (7.7) набор функций (7.2), найдем значения коэффициентов ряда

$$a_0 = \frac{2}{T}\int\limits_0^T s(t)dt, \tag{7.8}$$

$$a_n = \frac{2}{T}\int\limits_0^T s(t)\cos(2\pi n/T)dt, \tag{7.9}$$

$$b_n = \frac{2}{T}\int\limits_0^T s(t)\sin(2\pi n/T)dt. \tag{7.10}$$

Введя основную частоту $\omega = 2\pi T$ последовательности, образующей периодическую функцию $s(t)$, запишем ряд Фурье для периодического сигнала

$$s(t) = \frac{a_0}{2} + \sum_{n=1}^{\infty}(a_n\cos(n\omega t) + b_n\sin(n\omega t)). \tag{7.11}$$

Анализ (7.11) показывает, что функция $s(t)$ содержит независящую от времени постоянную составляющую $a_0/2$ и бесконечный набор гармонических колебаний, называемых гармониками с частотами $\omega_n = n\omega$ $(n = 1, 2, \ldots)$, кратными основной частоте последовательности. Можно показать, что имеет место равенство

$$\frac{1}{T}\int\limits_0^T s(t)^2dt = \frac{a_0^2}{2} + \sum_{k=1}^{\infty}[a_k^2 + b_k^2], \tag{7.12}$$

называемое равенством Парсеваля.

Можно получить эквивалентную форму ряда Фурье, если записать коэффициенты ряда Фурье в виде

$$a_n = A_n\cos\varphi_n, \quad b_n = A_n\sin\varphi_n,$$

где

$$A_n = \sqrt{a_n^2 + b_n^2},$$

$$\mathrm{tg}\varphi_n = b_n/a_n,$$

то получим:

$$s(t) = \frac{a_0}{2} + \sum_{n=1}^{\infty}A_n\cos(n\omega t - \varphi_n). \tag{7.13}$$

Спектральное разложение периодической функции $s(t)$ можно выполнить, используя систему базисных функций в виде экспонент с мнимыми показателями:

$$\{u_k\} = \left\{\frac{\exp(ik\omega t)}{\sqrt{T}}\right\}, \quad k = 0, \pm1, \pm2, \ldots, \tag{7.14}$$

$$\int_0^T u_m(t)u_k^*(t)dt = \frac{1}{T}\int_0^T e^{imt}e^{-ikt}dt = \delta_{mk}a$$

являются ортогональными.

Ряд Фурье в данном случае принимает вид

$$s(t) = \frac{1}{\sqrt{T}}\sum_{n=-\infty}^{\infty} c_n e^{in\omega t} \tag{7.15}$$

с коэффициентами

$$c_n = \frac{1}{\sqrt{T}}\int_0^T s(t)e^{-in\omega t}dt. \tag{7.16}$$

На практике принято использовать и другую форму записи ряда Фурье:

$$s(t) = \sum_{n=-\infty}^{\infty} C_n e^{in\omega t}, \tag{7.17}$$

где

$$C_n = \frac{1}{T}\int_0^T s(t)e^{-in\omega t}dt. \tag{7.18}$$

Выражения (7.15)–(7.18) представляют собой ряд Фурье в комплексной форме.

Спектр функции $s(t)$ в соответствии с формулами (7.15)–(7.18) содержит компоненты на отрицательной полуоси частот, причем $C_{-n} = C_n^*$, поэтому слагаемые с положительными и отрицательными частотами объединяются в пары, например:

$$C_n e^{in\omega t} + C_{-n}e^{-in\omega t} = |C_n|e^{i(n\omega t+\varphi_n)} + |C_n|e^{-i(n\omega t+\varphi_n)} = 2|C_n|\cos(n\omega t - \varphi_n).$$

Таким образом, отрицательная частота является не физическим, а математическим понятием, вытекающим из способа представления комплексных чисел.

***Задача* 7.1**

1. *Найдите аналитически коэффициенты ряда Фурье периодической последовательности прямоугольных видеоимпульсов (рис. 7.1) с известными параметрами.*

2. *В радиотехнике отношение $q = T/\tau$ называют скваженностью последовательности. Постройте и сравните зависимости амплитуды спектральных гармоник от частоты $n\omega$ при различной скваженности последовательности.*

Рис. 7.1. К вычислению спектра последовательности прямоугольных импульсов

7.3. ЭФФЕКТ ГИББСА

Эффект Гиббса возникает при восстановлении разрывных функций с помощью усеченного ряда Фурье. Изучим основные его проявления на примере разложения прямоугольной импульсной функции

$$s(t) = \begin{cases} -1/2, & \text{если} \quad 0 \le t \le T/2, \\ 1/2, & \text{если} \quad T/2 < t \le T, \end{cases}$$

которая имеет скачок, равный 1 в точке разрыва $T/2$.

Формальное разложение в ряд Фурье находится подстановкой функции $s(t)$ в формулы (7.8)–(7.10), вычислением интегралов, оказывающихся равными

$$a_k = 0,$$

$$b_k = \frac{1}{\pi}\frac{[1+(-1)^{k+1}]}{k} = \begin{cases} 2/\pi k, & k \text{ — четное}, \\ 0, & k \text{ — нечетное}, \end{cases}$$

и подстановкой выражений для коэффициентов в (7.11):

$$s(t) = \frac{2}{\pi}\left[\sin\left(\frac{2\pi}{T}t\right) + \frac{1}{3}\sin\left(\frac{2\pi}{T}3t\right) + \frac{1}{5}\sin\left(\frac{2\pi}{T}5t\right) + \ldots\right] =$$
$$= \frac{2}{\pi}\sum_{k=0}^{\infty}\frac{1}{2k+1}\sin\left(\frac{2\pi}{T}(2k+1)t\right).$$

Несмотря на то что полученное разложение позволяет познакомиться с основными особенностями эффекта Гиббса, мы предварим дальнейшее рассмотрением этого вопроса описанием последовательности команд, позволяющей получить в MATLAB разложение в ряд Фурье произвольных функций.

Для решения поставленной задачи необходимо создать следующие m-файлы: `FF.m` (содержит описание функции, разлагаемой в ряд Фурье), `AF.m` (описание функции, возвращающей значение заданного коэффициента разложения в ряд Фурье по косинусам в соответствии с (7.8), (7.9), `BF.m` (описание функции, возвращающей значение

заданного коэффициента разложения в ряд Фурье по синусам в соответствии с (7.10). Далее приводятся листинги данных файлов.

```
% листинг файла FF.m
function z=FF(t,T)
% описание функции,
% раскладываемой в ряд Фурье
N=length(t);
for i=1:N
  if t(i)<0
    z(i)=0;
  end;
  if (t(i)>=0)&(t(i)<=T/2)
    z(i)=1/2;
  end;
  if (t(i)>T/2)&(t(i)<=T)
    z(i)=-1/2;
  end;
  if t(i)>T
    z(i)=0;
  end;
end;

% листинг файла AF.m
function z=AF(k,T)
% описание функции, возвращающей значение
% k-го коэффициента
% разложения в ряд Фурье
% функции FF(t,T) по косинусам

dt=T/1000;
t=0:dt:T;

F=FF(t,T).*cos(2*pi*k/T*t); % вычисление значения
                            % подынтегральной функции

% вычисление интеграла
z=2/T*trapz(t,F);

% листинг файла BF.m
function z=BF(k,T)
% описание функции, возвращающей значение
% k-го коэффициента
% разложения в ряд Фурье
% функции FF(t,T) по синусам
```

```
dt=T/1000;
t=0:dt:T;

F=FF(t,T).*sin(2*pi*k/T*t); % вычисление значения
                            % подынтегральной функции

z=2/T*trapz(t,F); % вычисление интеграла
```

Далее необходимо выполнить следующую последовательность команд, сохраненную нами в файле `Glava7_1.m`:

```
% листинг файла Glava7_1.m
clear all % очистка рабочей области
Nf=9; % число гармоник
k=1:Nf;
T=1; % длительность импульса

A0=AF(0,1); % вычисление A(0)
            % в соответствии с (7.10)

for k=1:Nf  % вычисление коэффициентов
            % A(k), B(k)
  A(k)=AF(k,T);
  B(k)=BF(k,T);
end;

% вычисление значений
% усеченного ряда Фурье
% на временном интервале [0,1]
Np=1000;
t=0:T/Np:1;
for i=1:Np+1
  S=A0/2;
  for k=1:Nf
    S=S+A(k)*cos(2*pi*k/T*t(i))+...
      B(k)*sin(2*pi*k/T*t(i));
  end;
  s(i)=S;
end;
% визуализация усеченного
% ряда Фурье и исходной функции
plot(t,s,t,FF(t,T),'--')
```

Результаты выполнения описанной выше последовательности команд представлены на рис. 7.2. Сравнение графиков исходной функции и функции, являющейся результатом разложения в ряд Фурье, представленных на рис. 7.2, обнаруживает их от-

личие. Данное отличие между исходной функцией и усеченным рядом Фурье носит название эффекта Гиббса.

Рис. 7.2. Прямоугольный импульс и частичная сумма ряда Фурье (9 членов ряда)

Для выявления причины обнаруженного эффекта следует изменить в приведенном выше документе число гармоник ряда Фурье и пересчитать документ заново. Графики исходной функции и ее ряда Фурье при использовании 19 гармоник ($Nf = 19$) представлены на рис. 7.3. Анализ полученных результатов позволяет сделать каче-

Рис. 7.3. Прямоугольный импульс и частичная сумма ряда Фурье (19 членов ряда)

ственный вывод о том, что увеличение числа членов усеченного ряда Фурье приводит к уменьшению отличия между функциями $f(t)$ и $s(t)$.

***Задача* 7.2**

Для оценки количественных характеристик степени отличия между функциями $f(t)$ и $s(t)$ при различном числе членов усеченного ряда Фурье можно использовать максимальное значение функции $f(t) - s(t)$ и дисперсию данной функции. Постройте при различных значениях числа членов усеченного ряда Фурье Nf:

1. *Графики функций $f(t) - s(t)$.*

2. *Оцените, используя построенные ранее графики, периоды колебаний функции $f(t)-s(t)$ и сравните его с периодами колебаний последнего удержанного члена ряда Фурье.*

3. *Постройте зависимости максимального значения функции $f(t)-s(t)$ и ее среднеквадратичного отклонения от числа членов усеченного ряда Фурье на временном интервале* $[0,05; 0,045]$. ***Примечание:*** *для вычисления среднеквадратического отклонения используйте функцию* `std(v)`, `v` — *вектор, содержащий значения функции $f(t)-s(t)$.*

Решив задачу 7.2, вы, уважаемый читатель, определили, что период колебаний функции $f(t)-s(t)$ имеет период либо первого отброшенного члена, либо последнего удержанного члена. К. Ланцош предложил следующий способ уменьшения эффекта пульсаций, основанный на сглаживании усеченного ряда

$$s_N(t) = \frac{a_0}{2} + \sum_{k=1}^{N}\left(a_k \cos\left(\frac{2\pi k}{T}t\right) + b_k \sin\left(\frac{2\pi k}{T}t\right)\right) \tag{7.19}$$

за счет интегрирования (усреднения) по данному периоду.

При выборе в качестве интервала сглаживания периода последнего члена усеченного ряда Фурье (здесь для того, чтобы продемонстрировать исчезновение этого члена N — номер последнего удержанного члена, но на практике N может использоваться и номер первого отбрасываемого члена) сглаженное значение $h_N(t)$ получим как среднее от $s_N(t)$:

$$h_N(t) = \frac{1}{T/N}\int\limits_{t-(T/2N)}^{t+(T/2N)} s_N(\xi)d\xi. \tag{7.20}$$

Подставляя (7.19) в (7.20), получим

$$\begin{aligned} h_N(t) = {} & \frac{N}{T}\int\limits_{t-(T/2N)}^{t+(T/2N)} \frac{a_0}{2}d\xi + \\ & + \frac{N}{T}\sum_{k=1}^{N}\left[a_k \int\limits_{t-(T/2N)}^{t+(T/2N)} \cos\left(\frac{2\pi k}{T}\xi\right)d\xi + b_k \int\limits_{t-(T/2N)}^{t+(T/2N)} \sin\left(\frac{2\pi k}{T}\xi\right)d\xi\right] + \ldots = \\ & = \frac{a_0}{2} + \frac{N}{T}\sum_{k=1}^{N}\left[a_k\left\{\frac{\sin\left[2\pi k/T(t+T/2N)\right] - \sin\left[2\pi k/T(t-T/2)\right]}{2\pi k/T}\right\} - \right. \\ & \left. - b_k\left\{\frac{\cos\left[2\pi k/T(t+T/2N)\right] - \cos\left[2\pi k/T(t-T/2N)\right]}{2\pi k/T}\right\}\right]. \end{aligned} \tag{7.21}$$

Применяя к (7.21) известные тригонометрические формулы, окончательно получаем:

$$h_N(t) = \frac{a_0}{2} + \sum_{k=1}^{N} \sigma(N,k) \left[a_k \cos\left(\frac{2\pi k}{T} t\right) + b_k \sin\left(\frac{2\pi k}{T} t\right) \right], \tag{7.22}$$

где $\sigma(N,k)$ — так называемые сигма-факторы:

$$\sigma(N,k) = \frac{\sin(\pi k/N)}{\pi k/N}. \tag{7.23}$$

Таким образом, сглаженный ряд Фурье есть исходный ряд Фурье с коэффициентами, умноженными на соответствующие сигма-факторы. Отметим, что эффект сглаживания достигается за счет того, что при $k = N$ сигма-фактор N-го члена оказывается равным нулю.

С физической точки зрения формулу (7.22) можно трактовать как наблюдение исходной функции $s_N(t)$ через узкое просвечивающее прямоугольное окно шириной T/N. Наблюдаемая функция $h_N(t)$ есть средняя интенсивность света от исходной функции $s_N(t)$. Она оказывается более яркой, когда функция $s_N(t)$ больше, и менее яркой, когда функция меньше.

***Задача* 7.3**

Дополните описанный выше документ блоком, позволяющим проводить сглаживание усеченного ряда Фурье, и проведите сравнительный анализ отклонения между функциями $f(t)$ и $h_N(t)$, аналогично тому, как это было сделано в задаче 7.2.

7.4. СПЕКТРАЛЬНЫЙ АНАЛИЗ НЕПРЕРЫВНЫХ ФУНКЦИЙ

Рассмотрим обобщение метода рядов Фурье на случай непериодических сигналов, позволяющее получать их спектральные характеристики. В теории обработки радиотехнических сигналов она называется задачей анализа.

Пусть функция $s(t)$ задана на временном интервале конечной длительности $[0, T]$. Дополнив данную функцию такими же сигналами, периодически следующими через интервал времени T, получим изученную в предыдущем параграфе периодическую последовательность $s_p(t)$, которая может быть представлена в виде комплексного ряда Фурье

$$s_p(t) = \sum_{n=-\infty}^{\infty} C_n e^{jn\omega t} \tag{7.24}$$

с коэффициентами

$$C_n = \frac{1}{T} \int_0^T s(t) e^{-jn\omega t} dt. \tag{7.25}$$

Для возвращения к одиночному интервалу устремим к бесконечности период повторения T, при этом:

1. Частоты соседних гармоник $n\omega$ и $(n+1)\omega$ окажутся сколь угодно близкими, поэтому в формулах (7.24), (7.25) дискретную переменную $n\omega$ можно заменить непрерывной переменной ω.

2. Коэффициенты C_n станут неограниченно малыми из-за наличия величины T в знаменателе формулы (7.25).

Для нахождения предельного вида (7.24) при $T \to \infty$ введем понятие спектральной плотности мощности, воспользовавшись тем, что коэффициенты ряда Фурье образуют комплексно сопряженные пары:

$$C_n = A_n e^{i\varphi_n}, \quad C_{-n} = A_n e^{-i\varphi_n},$$

а каждой паре отвечает простое гармоническое колебание

$$A_n e^{i(n\omega t+\varphi_n)} + A_n e^{-i(n\omega t+\varphi_n)} = 2A_n \cos(n\omega t + \varphi_n)$$

с комплексной амплитудой $2A_n e^{i\varphi_n} = 2C_n$.

На малом интервале частот $\Delta\omega$ в окрестности некоторого выбранного значения частоты ω_0 будет содержаться $N = \Delta\omega/\omega = \Delta\omega T/2\pi$ отдельных пар спектральных составляющих, частоты которых отличаются сколь угодно мало, поэтому составляющие можно складывать, предполагая, что они имеют одну и ту же частоту и характеризуются одинаковыми комплексными амплитудами

$$2C_n = \frac{2}{T}\int\limits_{-\infty}^{\infty} s(t)e^{-i\omega_0 t}dt.$$

Таким образом, комплексная амплитуда эквивалентного гармонического сигнала, отображающая вклад всех спектральных составляющих, содержащихся внутри интервала $\Delta\omega$, равна

$$\Delta A_{\omega_0} = \frac{2N}{T}\int\limits_{-\infty}^{\infty} s(t)e^{-i\omega_0 t}dt = \frac{\Delta\omega}{\pi}\int\limits_{-\infty}^{\infty} s(t)e^{-i\omega_0 t}dt. \tag{7.26}$$

Из (7.25) видно, что можно ввести функцию

$$S(\omega) = \int\limits_{-\infty}^{\infty} s(t)e^{-i\omega t}dt, \tag{7.27}$$

называемую спектральной плотностью функции $s(t)$.

Обратной задачей к задаче анализа является задача нахождения вида функции $s(t)$ по его известной спектральной плотности, называемая в теории обработки радиотехнических сигналов задачей синтеза. Для ее решения предположим, что непериодический сигнал получается из периодической последовательности, когда ее период $T \to \infty$. Воспользовавшись формулами (7.24) и (7.25) запишем

$$s(t) = \lim_{T\to\infty}\sum_{n=-\infty}^{\infty}\frac{1}{T}S(n\omega)e^{in\omega t}. \tag{7.28}$$

Входящий в (7.28) коэффициент $1/T$ при любом целом n пропорционален разности между частотами соседних гармоник:

$$\frac{1}{T} = \frac{\omega}{2\pi} = \frac{1}{2\pi}[n(\omega+1) - n\omega]. \tag{7.29}$$

Подставляя (7.29) в (7.28), получаем:

$$s(t) = \lim_{T\to\infty} \frac{1}{2\pi} \sum_{n=-\infty}^{\infty} S(n\omega)e^{in\omega t}[\omega(n+1) - n\omega]. \tag{7.30}$$

Так как частотные интервалы между соседними гармониками неограниченно сокращаются, сумму в (7.30) можно заменить интегралом

$$s(t) = \frac{1}{2\pi} \int\limits_{-\infty}^{\infty} S(\omega)e^{i\omega t}d\omega. \tag{7.31}$$

Формула (7.31) называется обратным преобразованием Фурье функции $s(t)$.

Полученный фундаментальный результат формулируется следующим образом: функция $s(t)$ и его спектральная плотность $S(\omega)$ связаны прямым и обратным преобразованиями Фурье:

$$S(\omega) = \int\limits_{-\infty}^{\infty} s(t)e^{-i\omega t}dt, \tag{7.32}$$

$$s(t) = \frac{1}{2\pi} \int\limits_{-\infty}^{\infty} S(\omega)e^{i\omega t}d\omega. \tag{7.33}$$

В математике доказано, что соотношения (7.32), (7.33) справедливы для абсолютно интегрируемых функций, то есть функций удовлетворяющих условию

$$\int\limits_{-\infty}^{\infty} |s(t)|dt < \infty. \tag{7.34}$$

Данное условие существенно ограничивает класс допустимых функций, однако, с одной стороны, в современной математике разработан аппарат так называемых обобщенных функций, позволяющих разумным образом вычислять спектральные плотности неинтегрируемых функций, с другой — все реальные сигналы, регистрируемые в физических экспериментах, имеют конечную длительность, а потому автоматических удовлетворяют условию (7.34).

Перечислим без доказательства основные свойства преобразования Фурье:

1. Линейность преобразования Фурье: спектр $U(\omega)$ линейной комбинации функций $u(t) = \sum\limits_i a_i s_i(t)$ есть линейная комбинация спектров исходных функций с аналогичными коэффициентами:

$$U(\omega) = \sum_i a_i S_i(\omega).$$

2. Спектр вещественной функции $s(t)$ является комплексной функцией:

$$S(\omega) = \int\limits_{-\infty}^{\infty} s(t)\cos(\omega t) - i\int\limits_{-\infty}^{\infty} s(t)\sin(\omega t) = A(\omega) - iB(\omega),$$

причем

$$\int\limits_{-\infty}^{\infty} A(\omega)\sin(\omega t) = 0, \qquad \int\limits_{-\infty}^{\infty} B(\omega)\cos(\omega t) = 0. \tag{7.35}$$

3. Спектральная плотность функции, смещенной во времени, $s(t-t_0)$ связана со спектральной плотностью исходной функции $s(t)$, имеющей спектр $S(\omega)$, соотношением

$$s(t-t_0) \to S(\omega)e^{-i\omega t_0}.$$

4. Спектральная плотность произведения функций $s(t)$, $v(t)$, имеющих спектры $S(\omega)$ и $V(\omega)$, соответственно, $S_{s\cdot v}$ есть свертка спектров исходных функций:

$$S_{s\cdot v} = \int\limits_{-\infty}^{\infty} S(\xi)V(\omega-\xi)d\xi.$$

5. Функция, спектр которой представим в виде произведения

$$S(\omega) = S_1(\omega)S_2(\omega),$$

причем $S_1(\omega)$ — спектр функции $s_1(t)$, $S_2(\omega)$ — спектр функции $s_2(t)$, является сверткой сигналов во временной области:

$$s(t) = \int\limits_{-\infty}^{\infty} s_1(t-\xi)s_2(\xi)d\xi.$$

6. Скалярное произведение двух функций $(s(t), v(t))$:

$$(s(t), v(t)) = \int\limits_{-\infty}^{\infty} s(t)v^*(t)dt$$

с точностью до коэффициента пропорционально скалярному произведению их спектральных плотностей

$$(s, v) = \int\limits_{-\infty}^{\infty} s(t)v^*(t)dt = \frac{1}{2\pi}\int\limits_{-\infty}^{\infty} S(\omega)V^*(\omega)d\omega.$$

Отметим, что последнее свойство позволяет вычислить спектральную плотность для бесконечно протяженных неинтегрируемых функций:

1) функции постоянной величины $s(t) = A = \text{const}$:

$$S(\omega) = 2\pi A\delta(\omega); \tag{7.36}$$

2) монохроматической комплексной экспоненциальной функции $s(t) = A\exp(i\omega_0 t)$:

$$2\pi A\delta(\omega - \omega_0); \tag{7.37}$$

3) спектральную плотность гармонических колебаний $s(t) = A\cos(\omega_0 t)$, $s(t) = A\sin(\omega_0 t)$, соответственно:

$$S(\omega) = \pi A[\delta(\omega - \omega_0) + \delta(\omega + \omega_0)], \tag{7.38}$$

$$S(\omega) = -i\pi A[\delta(\omega - \omega_0) + \delta(\omega + \omega_0)]. \tag{7.39}$$

Здесь $\delta(x - a)$ — дельта-функция Дирака, обладающая следующими свойствами:

$$\delta(x - a) = \begin{cases} 0, & \text{если} \quad x \neq a, \\ \infty, & \text{если} \quad x = a, \end{cases} \tag{7.40}$$

$$\int\limits_{-\infty}^{\infty} \delta(x - a)dx = 1, \tag{7.41}$$

$$\int\limits_{-\infty}^{\infty} f(x)\delta(x - a)dx = f(a). \tag{7.42}$$

7.5. СПЕКТРАЛЬНЫЙ АНАЛИЗ ДИСКРЕТНЫХ ФУНКЦИЙ

На практике наиболее часто приходится находить спектры функций, заданных в дискретные моменты времени, к таковым относятся как реальные сигналы, регистрируемые с помощью аналого-цифрового преобразователя (АЦП), так и любые зависимости, задаваемые таблично. Например, решения дифференциальных уравнений и систем ДУ, вычисляемые в дискретные моменты времени. Далее мы будем рассматривать только задачу спектрального анализа функций, заданных на равномерной временной сетке, то есть через равный промежуток времени Δ, называемый интервалом дискретизации:

$$\Delta = t_m - t_{m-1} = t_{m-1} - t_{m-2} = \ldots, \tag{7.43}$$

так как любую дискретную функцию можно, используя процедуру интерполяции, привести к равномерной временной сетке.

Для нахождения спектральной плотности дискретной функции $S_d(\omega)$ заметим, что переход от непрерывной функции $s(t)$ к дискретной функции $s_d(t)$ можно описать, введя в рассмотрение обобщенную функцию

$$\eta(t) = \sum_{k=-\infty}^{\infty} \delta(t - k\Delta), \tag{7.44}$$

называемую дискретизирующей последовательностью.

Учитывая фильтрующие свойства дельта-функции (7.33), дискретная функция есть скалярное произведение непрерывной функции $s(t)$ и функции $\eta(t)$:

$$\begin{aligned} s_d(t) = (s(t), \eta(t)) = \int\limits_{-\infty}^{\infty} s(t) \sum_{k=-\infty}^{\infty} \delta(t - k\Delta) dt = \\ = (\ldots, x(-2\Delta), x(-\Delta), x(0), x(\Delta), x(2\Delta), \ldots). \end{aligned} \tag{7.45}$$

Так как функция $\eta(t)$ определена лишь в точках $t_k = k\Delta$, $(k = 0, \pm 1, \pm 2, \ldots)$ интегрирование в (7.45) можно заменить суммированием, при этом роль дифференциала будет играть интервал дискретизации Δ:

$$s_d(t) = \Delta \sum_{k=-\infty}^{\infty} s_k \delta(t - k\Delta), \tag{7.46}$$

то есть с точностью до коэффициента пропорциональности Δ интеграл (7.46) равен произведению функции $s(t)$ и дискретизирующей последовательности $\eta(t)$. Будем считать, что для функций $s(t)$ и $\eta(t)$ известны их спектральные плотности $S(\omega)$ и $S_\eta(\omega)$ соответственно. В соответствие со свойством 4 (см. раздел 7.3) спектр произведения двух функций равен свертке спектров этих функций, поэтому

$$S_d(\omega) = \frac{\Delta}{2\pi} \int\limits_{-\infty}^{\infty} S_\eta(\xi) S(\omega - \xi) d\xi. \tag{7.47}$$

Для нахождения спектральной плотности функции $\eta(t)$ разложим данную функцию в ряд Фурье. В соответствии с (7.18)

$$C_n = \frac{1}{\Delta} \int\limits_{0}^{\Delta} \delta(t) e^{i2\pi t/\Delta} dt = \frac{1}{\Delta}. \tag{7.48}$$

Подставляя (7.48) в (7.17), получаем

$$\eta(t) = \frac{1}{\Delta} \sum_{n=-\infty}^{\infty} e^{i2\pi nt/\Delta} dt. \tag{7.49}$$

Подставив (7.49) в (7.27) и учтя (7.37), найдем

$$S_\eta(\omega) = \frac{1}{\Delta} \int\limits_{-\infty}^{\infty} \sum_{n=-\infty}^{\infty} e^{i2\pi nt/\Delta} e^{-i\omega t} dt = \frac{2\pi}{\Delta} \sum_{n=-\infty}^{\infty} \delta(\omega - 2\pi n/\Delta). \tag{7.50}$$

Из (7.50) видно, что спектр дискретизирующей последовательности состоит в частотной области из бесконечной совокупности Δ-импульсов, при этом функция $S_\eta(\omega)$ является периодической с периодом $2\pi/\Delta\ c^{-1}$.

Подставляя (7.50) в (7.47) и меняя порядок суммирования и интегрирования, находим

$$S_d(\omega) = \sum_{n=-\infty}^{\infty} S(\omega - 2\pi n/\Delta). \tag{7.51}$$

Из (7.51) видно, что спектр дискретной функции представляет собой сумму бесконечного числа «копий» спектра исходного непрерывной функции, расположенной на оси частот через одинаковые интервалы (рис. 7.4).

Рис. 7.4. Спектральная плотность модулированной импульсной последовательности при различных значениях верхней граничной частоты (верхний рисунок — верхняя граничная частота велика; нижний рисунок — верхняя граничная частота мала)

Из рис. 7.4 видно, что для действительных функций $s(t)$, имеющих спектр, симметричный относительно точки $\omega = 0$ и ограниченный верхней граничной частотой $\omega_u = 2\pi f_u$, условие отсутствия наложения отдельных копий спектра записывается в виде

$$\omega_u \le \frac{\pi}{\Delta}, \quad f_u \le \frac{1}{2\Delta}. \tag{7.52}$$

Выражение для нахождения спектральной плотности функции $s_d(t)$ по ее дискретным значениям можно получить, подставив (7.46) в (7.32):

$$\begin{aligned} S_d(\omega) &= \int_{-\infty}^{\infty} s_d(t) e^{-i\omega t} dt = \\ &= \Delta \int_{-\infty}^{\infty} \sum_{k=-\infty}^{\infty} s_k \delta(t - k\Delta) e^{-i\omega t} dt = \Delta \sum_{k=-\infty}^{\infty} s_k e^{-i\omega k\Delta}. \end{aligned} \tag{7.53}$$

С другой стороны спектральная плотность была найдена выше другим способом (см. выражение (7.50)), поэтому справедливо соотношение

$$\Delta \sum_{k=-\infty}^{\infty} s_k e^{-i\omega k\Delta} = \sum_{n=-\infty}^{\infty} S_d(\omega - 2\pi n/\Delta), \tag{7.54}$$

называемое формулой суммирования Пуассона.

Из (7.54) видно, что однозначно найти функцию $S_d(\omega)$ по значениям функции $s_d(t)$, вообще говоря, невозможно из-за эффекта наложения копий спектра. Единственное исключение составляет случай, когда исходная функция $s(t)$ имеет спектр, удовлетворяющий условию (7.52), тогда очевидно, что спектр дискретной функции $S_d(\omega)$ совпадает со спектром непрерывной функции $S(\omega)$:

$$S(\omega) = \begin{cases} 0, & \omega < -\pi/\Delta, \\ \Delta \sum\limits_{k=-\infty}^{\infty} x_k e^{-i\omega k\Delta}, & -\pi/\Delta \le \omega \le \pi/\Delta, \\ 0, & \omega > \pi/\omega. \end{cases} \tag{7.55}$$

Полученный результат является следствием ***теоремы Котельникова***, имеющей следующую формулировку:

Произвольный сигнал, спектр которого не содержит частот выше f_u ***Гц, может быть полностью восстановлен, если известны отсчетные значения этого сигнала, взятые через равные промежутки времени*** $1/2f_u$.

Здесь под восстановлением имеется в виду вычисление значений функции в любой момент времени, не совпадающий с моментами дискретизации.

7.6. СПЕКТРАЛЬНЫЙ АНАЛИЗ ДИСКРЕТНЫХ ФУНКЦИЙ КОНЕЧНОЙ ДЛИТЕЛЬНОСТИ

В предыдущем параграфе при решении задачи о нахождении спектральной плотности дискретной функции мы предполагали, что значения этой функции заданы в неограниченном числе точек на временной оси. На практике получить столь обширные сведения о поведении функции оказывается невозможным. Это обстоятельство приводит к тому, что всегда спектральный анализ проводится на временном интервале конечной длительности.

Рассмотрим особенности спектрального представления дискретной функции, заданной на временном интервале конечной длительности $[0, T]$ N отсчетами s_0, s_1, $s_2, \ldots, s_{N-1}$, взятыми соответственно в моменты времени $0, \Delta, 2\Delta, \ldots, (N-1)\Delta$. Полное число отсчетов $N = T/\Delta$.

Используя подход, развитый в предыдущем параграфе, поставим в соответствие исходной непрерывной функции $s(t)$ ее дискретное представление

$$s_d(t) = \Delta \sum_{k=0}^{N-1} s_k \delta(t - k\Delta). \tag{7.56}$$

Дискретная последовательность $s_d(t)$ представима рядом Фурье:

$$s_d(t) = \Delta \sum_{n=-\infty}^{\infty} C_n e^{i2\pi nt/T}, \tag{7.57}$$

где

$$C_n = \frac{1}{T\Delta} \int_0^T s_d(t) e^{-i2\pi nt/T} dt. \tag{7.58}$$

Подставляя (7.56) в (7.58) и вводя безразмерную переменную $\xi = t/\Delta$, получим:

$$C_n = \frac{1}{N\Delta} \int_0^{N\Delta} \sum_{k=0}^{N-1} s_k \delta(t - k\Delta) e^{-i2\pi t/T} dt =$$

$$= \frac{1}{N} \int_0^N \sum_{k=0}^{N-1} s_k \delta(\xi - k) e^{-i2\pi n\xi/N} d\xi = \frac{1}{N} \sum_{k=0}^{N-1} s_k \int_0^N \delta(\xi - k) e^{-i2\pi n\xi/N} d\xi.$$

Используя фильтрующее свойство дельта-функции, получаем

$$C_n = \frac{1}{N} \sum_{k=0}^{N-1} s_k e^{-i2\pi nk/N}. \tag{7.59}$$

Формула (7.59) определяет последовательность коэффициентов, образующих дискретное преобразование Фурье (ДПФ), которое имеет следующие свойства:

1. ДПФ есть линейное преобразование.

2. Число коэффициентов C_0, C_1, C_2, ..., C_{N-1}, вычисляемых в соответствии с формулой (7.59), равно числу отсчетов дискретной последовательности.

3. Коэффициент C_0 (постоянная составляющая), есть среднее значение дискретной последовательности.

4. Если N — четное число, то

$$C_{N/2} = \frac{1}{N} \sum_{k=0}^{N-1} (-1)^k s_k.$$

5. Для вещественной дискретной последовательности коэффициенты ДПФ, номера которых расположены симметрично относительно $N/2$, образуют сопряженные пары:

$$C_{N-n} = \frac{1}{N} \sum_{k=0}^{N-1} s_k e^{-i2\pi(N-n)k/N} = \frac{1}{N} \sum_{k=0}^{N-1} s_k e^{i2\pi nk/N} = C_n^*,$$

поэтому можно считать, что коэффициенты $C_{N/2+1}, \ldots, C_{N-1}$ отвечают отрицательным частотам.

***Задача* 7.4**

Вычислите, используя MATLAB, ДПФ дискретной последовательности, заданной восьмью равноотстоящими отсчетами

$$\{s_k\} = (1, 1, 1, 1, 0, 0, 0, 0),$$

и изобразите графически абсолютные значения коэффициентов ДПФ, называемых спектром мощности.

1. *Сколько различных гармоник имеет ДПФ данной последовательности?*

2. *Считая длительность интервала дискретизации равной 1 сек, вычислите частоты этих гармоник и величины $\Delta f_i = f_{i+1} - f_i$ (расстояние между гармониками)? Как изменится результат, если интервал дискретизации будет равен 0,1 с, 0,5 с, 2 с?*

3. *Какова верхняя граничная частота спектра дискретной последовательности при указанных длительностях интервала дискретизации?*

***Задача* 7.5**

Вычислите, используя MATLAB, ДПФ дискретной последовательности, заданной восьмью равноотстоящими отсчетами

$$\{s_k\} = (1, 1, 1, 1, 0, 0, 0, 0, 0, 0, 0, 0),$$

и изобразите графически абсолютные значения коэффициентов ДПФ.

1. *Сколько различных гармоник имеет ДПФ данной последовательности?*

2. *Считая длительность интервала дискретизации, равной 1 с, вычислите частоты этих гармоник величины $\Delta f_i = f_{i+1} - f_i$ (расстояние между гармониками). Как изменится результат, если интервал дискретизации будет равен 0,1 с, 0,5 с, 2 с?*

3. *Какова верхняя граничная частота спектра дискретной последовательности при указанных длительностях интервала дискретизации?*

4. *Сравните полученные результаты с результатами задачи 7.3. К чему приводит дополнение дискретной последовательности нулями? Почему граничные частоты спектров дискретных последовательностей, имеющих различное количество членов, но одинаковый интервал дискретизации, оказываются одинаковыми?*

Если для дискретной последовательности

$$s_0, s_1, s_2, \ldots, s_{N-1}$$

найдены коэффициенты ДПФ

$$C_0, C_1, C_2, \ldots, C_{N-1},$$

то восстановление исходной дискретной последовательности может быть осуществлено в соответствии с (7.57). При этом, как очевидно, бесконечный ряд Фурье становится рядом с конечным числом членов:

$$\begin{aligned} x(t) =& C_0 + 2|C_1|\cos(2\pi t/T + \varphi_1) + \\ &+ 2|C_2|\cos(4\pi t/T + \varphi_2) + \ldots + 2|C_{N/2}|\cos\left(N\pi t/T + \varphi_{N/2}\right), \end{aligned} \tag{7.60}$$

где $\varphi_i = \arg(C_i)$, $i = 1, 2, \ldots, N/2$ — фазовый угол коэффициента ДПФ.

Для нахождения формулы восстановления дискретной последовательности

$$s_0, s_1, s_2, \ldots, s_{N-1}$$

по известным коэффициентам ДПФ

$$C_0, C_1, C_2, \ldots, C_{N-1}$$

умножаем (7.58) на базисную функцию $\exp(i2\pi nk/N)$ и проводим суммирование по индексу n:

$$\begin{aligned} \sum_{n=0}^{N-1} C_n e^{i2\pi nk/N} &= \frac{1}{N}\sum_{n=0}^{N-1} e^{i2\pi nk/N} \sum_{m=0}^{N-1} s_m e^{-i2\pi nm/N} = \\ &= \frac{1}{N}\sum_{m=0}^{N-1} s_m \sum_{n=0}^{N-1} e^{i2\pi(k-m)n/N}. \end{aligned} \tag{7.61}$$

Так как базисные функции являются ортогональными

$$\sum_{n=0}^{N-1} e^{i2\pi(k-m)n/N} = N\delta_{km}, \tag{7.62}$$

где δ_{ik} — символ Кронекера:

$$\delta_{km} = \begin{cases} 1, & \text{если} \quad k = m, \\ 0, & \text{если} \quad k \neq m. \end{cases}$$

Подставляя (7.62) в (7.61), получаем

$$\frac{1}{N}\sum_{m=0}^{N-1} s_m \sum_{n=0}^{N-1} e^{i2\pi(k-m)n/N} = \frac{1}{N}\sum_{m=0}^{N-1} s_m N\delta_{km} = s_k,$$

откуда

$$x_k = \sum_{n=0}^{N-1} C_n e^{i2\pi nk/N}. \tag{7.63}$$

***Задача* 7.6**

Используя MATLAB и коэффициенты ДПФ, полученные в задачах 7.3, 7.4, проверьте правильность формул (7.56) *и* (7.59).

7.7. БЫСТРОЕ ПРЕОБРАЗОВАНИЕ ФУРЬЕ

Из формул (7.59) и (7.63) видно, что для вычисления ДПФ или обратного ДПФ последовательности из N элементов, требуется выполнить N^2 операций с комплексными числами. Если число элементов обрабатываемых массивов составляет порядок тысячи и более, то время, необходимое на выполнение этих преобразований, резко возрастает и теряется возможность обработки сигналов в реальном масштабе времени.

В 60-е годы Кули и Тьюки был предложен метод вычисления коэффициентов Фурье, позволивший снизить объем вычислений до операций. Он получил название алгоритма быстрого преобразования Фурье (БПФ). В настоящее время процедуры, реализующие алгоритм БПФ, входят во все математические библиотеки, используемые при написании программ на языках программирования высокого уровня, и специализированные пакеты для математических вычислений. В связи с тем, что подробное рассмотрение алгоритма выходит за рамки нашей книги, мы рассмотрим только основную идею БПФ для случая, когда число отсчетов $N = 2^p$, где p — целое число.

Разобьем входную последовательность s_k, где $k = 0, 1, 2, \ldots, N/2-1$, на две части с четными и нечетными номерами:

$$\{s_k\}_E = \{s_{2k}\}, \quad \{s_k\}_O = \{s_{2k+1}\}, \tag{7.64}$$

Это позволяет представить n-й коэффициент ДПФ в виде

$$\begin{aligned} C_n =& \frac{1}{N} \sum_{k=0}^{N/2-1} \left(s_{2k} e^{-i\frac{2\pi n}{N}2k} + s_{2k+1} e^{-i\frac{2\pi n}{N}(2k+1)} \right) = \\ =& \frac{1}{N} \left(\sum_{k=0}^{N/2-1} s_{2kE} e^{-i\frac{2\pi n}{N/2}k} + e^{-i\frac{2\pi n}{N}} \sum_{k=0}^{N/2-1} s_{(2k+1)O} e^{-i\frac{2\pi n}{N/2}k} \right). \end{aligned} \tag{7.65}$$

Из (7.65) видно, что первая половина коэффициентов ДПФ исходного сигнала с номерами от 0 до $N/2-1$ выражается через коэффициенты ДПФ двух частных последовательностей:

$$C_n = C_{nE} + e^{-i\frac{2\pi n}{N}} C_{nO}, \tag{7.66}$$

где $n = 0, 1, 2, \ldots, N/2-1$.

Так как последовательности коэффициентов массивов $\{s_k\}_E$ и $\{s_k\}_O$ являются периодическими с периодом $N/2$, то

$$C_{nE} = C_{n+N/2E}, \quad C_{nO} = C_{n+N/2O}. \tag{7.67}$$

Подставляя (7.67) в (7.66) и учитывая, что $e^{-i\frac{2\pi(n+N/2)}{N}} = e^{-i\pi} e^{-i\frac{2\pi n}{N}} = -e^{-i\frac{2\pi n}{N}}$, получаем выражение для второй половины множества коэффициентов ДПФ:

$$C_{n+N/2} = C_{nE} - e^{-i\frac{2\pi n}{N}} C_{nO}, \tag{7.68}$$

где $n = 0, 1, 2, \ldots, N/2-1$.

Формулы (7.66), (7.68) лежат в основе алгоритма БПФ: последовательности отсчетов с четными и нечетными номерами вновь разбиваются на две части. Процесс продолжается до тех пор, пока не получается последовательность, состоящая из одного элемента. ДПФ данной последовательности совпадет с самим элементом. Затем последовательно находятся коэффициенты ДПФ предыдущих последовательностей.

В MATLAB быстрое одномерное преобразование Фурье реализовано парой функций, выполняющих прямое и обратное БПФ: `fft/ifft`. Данные функции используются как для действительных, так и для комплексных последовательностей, при этом длина последовательностей может быть произвольной.

Обращение к функциям:

`fft(v)` — возвращает дискретное преобразование Фурье 2^m-мерного вектора, аргумент которого есть результат дискретизации через равные промежутки времени некоторой функции. Результат работы программы — комплексный вектор размерности 2^{m+1}. Элементы вектора, возвращаемого функцией `fft`, вычисляются по формуле

$$c_n = \sum_{k=0}^{N-1} v_k e^{i2\pi nk/N},$$

где N — число элементов вектора v.

`ifft(v)` — обратное дискретное преобразование Фурье комплексного вектора, содержащего значения ДПФ. Вектор v должен иметь 2^{m+1} элементов. Результат работы программы — действительный вектор размерности 2^{m+1}. Элементы вектора, возвращаемого функцией `ifft`, вычисляются по формуле

$$c_n = \frac{1}{N} \sum_{k=0}^{N-1} v_k e^{-i2\pi nk/N},$$

где N — число элементов вектора v. Отметим, что для всех векторов справедливо соотношение `ifft(fft(v))=v`.

`fft(v,n)` — возвращает дискретное преобразование Фурье 2^n-мерного вектора, аргумент которого есть результат дискретизации через равные промежутки времени некоторой функции. Результат работы программы есть комплексный вектор размерности 2^{n+1}. Если $n > length(v)$, последовательность, хранящаяся в векторе v, дополняется нулями.

`ifft(v,n)` — обратное дискретное преобразование Фурье, комплексного вектора, содержащего значения ДПФ. Результат работы программы — комплексный вектор размерности 2^{n+1}. Если $n > length(v)$, последовательность, хранящаяся в векторе v, дополняется нулями.

В качестве примера использования функций вычисления БПФ MATLAB рассмотрим задачу о вычислении спектра функции, являющейся суммой двух периодических функций

$$s(t) = A_1 \sin(2\pi f_1 t) + A_2 \sin(2\pi f_2 t),$$

где $A_1 = 0,8$, $f_1 = 40$ *Гц*, $A_2 = 0,4$, $f_2 = 90$ *Гц*, заданной набором дискретных значений в $N = 8192$ точках на интервале $[0, 2]$ *с*. Для решения задачи необходимо выполнить следующую последовательность команд, сохраненную нами в файле `Glava7_2.m`:

```
% листинг файла Glava7_2.m
>> N=8192; % число точек, используемых
           % для вычисления функции
>> i=1:N;
>> Tmax=2; % длительность временного
           % интервала в секундах
>> t(i)=Tmax/(N-1)*(i-1); % задание временной сетки

>> f1=40 % частота первой составляющей в Гц
>> f2=90 % частота второй составляющей в Гц

% вычисление мгновенных значений функции
>> f(i)=0.8*sin(2*pi*f1*t(i))+...
        0.4*sin(2*pi*f2*t(i));
>> figure(1);plot(t,f);

% вычисление спектра
>> c=fft(f);

% вычисление нормированного
% амплитудно-частотного спектра
>> j=2:N/2;
>> Cm(j-1)=abs(c(j-1))/(N/2);
>> Freq(j-1)=(j-1)/Tmax; % вычисление вектора частот в Гц
>> figure(2);plot(Freq,Cm);
>> axis([0 100 0 1]);
```

Результат выполнения описанной выше последовательности команд представлен на рис. 7.5, 7.6.

Рис. 7.5. Зависимость мгновенных значений функции $f = f(t)$ от времени

Рис. 7.6. Спектр функции $f = f(t)$

***Задача* 7.7**

Вычислите спектр функции

$$s\left(t\right) = A\sin\left(2\pi f_s t + \pi/5\right),$$

где $A = 3$, $f_s = 10$ *Гц по* $N = 2^{10}$ *известным значениям на интервале* $[0, 2]$ *с.*

1. *Изменился ли вид спектра по сравнению с ранее рассмотренным случаем?*
2. *Сравните фазы гармоник, имеющих частоту* $f = 10$ *Гц в первом и во втором случаях.*

***Задача* 7.8**

Вычислите спектр функции

$$s(t) = A_1\sin(2\pi f_1 t + \varphi_1) + A_2\sin(2\pi f_2 t + \varphi_2),$$

где $A_1 = 3$, $f_s = 10$ Гц, $\varphi_1 = \pi/8$, $A_2 = 0,5$, $f_2 = 15$ Гц, $\varphi_2 = \pi/3$ *по* $N = 2^{10}$ *известным значениям на интервале* $[0, 2]$ *с.*

1. *Изменился ли вид спектра по сравнению с ранее рассмотренным случаем?*
2. *Сравните фазы гармоник, имеющих частоту 10 Гц в первом и во втором случаях.*

***Задача* 7.9**

Для разделения функций, входящих в функцию $s(t)$*, заданную в предыдущей задаче, можно использовать полосовую фильтрацию, которая заключается в умножении спектра функции* $S(\omega)$ *на весовую функцию* $w(\omega)$*:*

$$w(\omega) = \begin{cases} 1, & \text{если} \quad \omega \in [\omega_1, \omega_2], \\ 0, & \text{если} \quad \omega \notin [\omega_1, \omega_2], \end{cases}$$

и выполнении обратного преобразования Фурье.

1. *Используя полосовую фильтрацию, проведите разделение функций, входящих в функцию* $s(t)$*.*
2. *Сравните отклонение функций, значения которых определены в результате полосовой фильтрации функции* $s(t)$*, от исходных функций.*

***Задача* 7.10**

1. *Исследуйте численно спектральный состав функции вида*

$$s(t) = U_0\left(1 + \sum_m a_m\cos(\Omega_m t + \Phi_m)\right)\cos(\omega_0 t + \varphi_0),$$

которую в радиотехнике используют для описания сигналов с амплитудной модуляцией при различных значениях параметров.

2. *Исследуйте численно спектральный состав функции вида*

$$s(t) = U_0 \cos\left(\omega_0 t + \sum_m a_m \sin(\Omega_m t + \Phi_m)\right),$$

которую в радиотехнике используют для описания сигналов с нелинейной частотной модуляцией при различных значениях параметров.

3. *Исследуйте численно спектральный состав функции вида*

$$s(t) = U_m \cos(\omega_0 t + \mu t^2/2),$$

которую в радиотехнике используют для описания сигналов с линейной частотной модуляцией при различных значениях параметров.

***Задача* 7.11**

В первой главе нами была подробно рассмотрена задача построения траектории движения Марса в системе координат, связанной с Землей, в которой была вычислена зависимость расстояния $s(t)$ между Землей и Марсом от времени. Проведите спектральный анализ функции $s(t)$ и определите значения: постоянной составляющей функции $s(t)$, амплитуду несущей и ее частоту, глубину амплитудной модуляции и частоту модулирующей функции.

7.8. ВЛИЯНИЕ ДЛИНЫ ИНТЕРВАЛА АНАЛИЗА НА ТОЧНОСТЬ ИЗМЕРЕНИЯ ЧАСТОТЫ МОНОХРОМАТИЧЕСКОГО СИГНАЛА

Рассмотрим одну из проблем, возникающих при спектральном анализе как непрерывных, так и дискретных функций конечной длительности. Напомним, что для первого типа функции преобразование Фурье вычисляется в соответствии с формулами (7.24), (7.25), второго типа функции — (7.57), (7.58). Анализ (7.25), (7.58) показывает, что значения коэффициентов ряда Фурье оказываются зависящими от длительности интервала, на котором задана исходная функция или дискретная последовательность, что обусловлено:

1. Наличием множителя $1/T$, T — длительность интервала анализа, в выражениях для вычисления спектральной плотности мощности и коэффициентов ряда Фурье (7.25), (7.58) соответственно.

2. Зависимостью частоты основной гармоники, а следовательно, и частот всех остальных базисных функций от длительности интервала анализа.

Предваряя аналитическое рассмотрение данной проблемы, рассмотрим результаты проведенных численных экспериментов, в которых были вычислены коэффициенты разложения в ряд Фурье гармонического монохроматического сигнала, частота которого равняется 10 Гц, при различной длительности интервалов анализа (рис. 7.7–7.10).

Рис. 7.7. Спектр гармонического монохроматического сигнала (длительности интервалов анализа (сверху вниз): 1 *с*, 1,1 *с*, 1,2 *с*, 1,3 *с*)

Рис. 7.8. Спектр гармонического монохроматического сигнала (длительности интервалов анализа (сверху вниз): 1,4 *с*, 1,5 *с*, 1,6 *с*, 1,7 *с*)

Рис. 7.9. Спектр гармонического монохроматического сигнала (длительности интервалов анализа (сверху вниз): 1,8 *с*, 1,9 *с*, 2,0 *с*, 2,1 *с*)

Рис. 7.10. Спектр гармонического монохроматического сигнала (длительности интервалов анализа (сверху вниз): 2,2 *с*, 2,3 *с*, 2,4 *с*, 2,5 *с*)

Из рис. 7.7—7.10 видно, что при изменении интервала анализа происходит изменение спектра сигнала, называемое в литературе «расплыванием спектра». Данное изменение проявляется в том, что:

1) происходит смещение частоты гармоники с максимальной амплитудой относительно его истинного значения;

2) отличными от нуля оказываются амплитуды гармоник, расположенных вблизи гармоники с максимальной амплитудой;

3) наибольшая амплитуда гармоники оказывается меньше амплитуды гармонического монохроматического сигнала.

Для более наглядной иллюстрации описанных выше изменений спектра удобно использовать зависимости частоты гармоники с максимальной амплитудой и амплитуды данной гармоники от длительности интервала анализа (рис. 7.11, 7.12).

Рис. 7.11, 7.12 являются более наглядной иллюстрацией сделанного выше вывода, основанного на анализе изменения формы спектра при изменении длительности интервала анализа. Для построения зависимостей, представленных на рис. 7.7—7.12, необходимо выполнить следующую последовательность команд, сохраненную нами в файле `Glava7_3.m`:

```
% Листинг файла Glava7_3.m
L=2^14; % задание числа узлов
        % временной сетки
k=1:L;
f=10; % задание частоты сигнала

% вычисление спектров,
% частот спектральных гармоник,
% имеющих максимальную амплитуду,
```

Рис. 7.11. Зависимость частоты спектральной гармоники гармонического монохроматического сигнала, имеющей максимальную амплитуду, от длительности интервала анализа (начало отсчета соответствует длительности временного интервала, на котором укладывается 10 периодов анализируемого сигнала)

Рис. 7.12. Зависимость максимальной амплитуды спектральных гармоник гармонического монохроматического сигнала от длительности интервала анализа (начало отсчета соответствует длительности временного интервала, на котором укладывается 10 периодов анализируемого сигнала)

```
% и зависимостей частоты спектральной гармоники
% с максимальной амплитудой и ее амплитуды
% для интервалов анализа длительностью
% 1.0, 1.1, 1.2, 1.3 с
for i=1:4
   j=2:L/2-1;
  Tmax=1/f*(10+0.1*(i-1)); % вычисление длительности
                           % интервала анализа
  Freq(j-1)=(j-1)/Tmax; % вычисление частот
                        % базисных функций
                        % преобразования Фурье
  dt=Tmax/(L-1); % вычисление шага
                 % временной сетки
  t=dt*(k-1); % вычисление координат
              % узлов временной сетки
  u=sin(2*pi*f*t); % вычисление мгновенных
                   % значений дискретного сигнала
  S=fft(u); % вычисление спектра

  % визуализация спектра
  S1(j-1)=abs(S(j))/(L/2);
  [q1 q2]=max(S1);
  Fm(i,1)=Tmax*f-10;
  Fm(i,2)=Freq(q2);
  Fm(i,3)=q1;
  subplot(4,1,i);
```

```
  stem(Freq,S1);
  axis([1 20 0 1])
end;

% вычисление спектров,
% частот спектральных гармоник,
% имеющих максимальную амплитуду,
% и зависимостей частоты спектральной гармоники
% с максимальной амплитудой и ее амплитуды
% для интервалов анализа длительностью
% 1.4, 1.5, 1.6, 1.7 с
figure
for i=5:8
  Tmax=1/f*(10+0.1*(i-1)); % вычисление длительности
                           % интервала анализа
  Freq(j-1)=(j-1)/Tmax; % вычисление частот
                        % базисных функций
                        % преобразования Фурье
  dt=Tmax/(L-1); % вычисление шага
                 % временной сетки
  t=dt*(k-1);
  u=sin(2*pi*f*t); % вычисление мгновенных значений
                   % дискретного сигнала
  S=fft(u); % вычисление спектра

  % визуализация спектра
  S1(j-1)=abs(S(j))/(L/2);
  [q1 q2]=max(S1);
  Fm(i,1)=Tmax*f-10;
  Fm(i,2)=Freq(q2);
  Fm(i,3)=q1;
  subplot(4,1,i-4);
  stem(Freq,S1);
  axis([1 20 0 1])
end;

% вычисление спектров,
% частот спектральных гармоник,
% имеющих максимальную амплитуду,
% и зависимостей частоты спектральной
% гармоники с максимальной амплитудой
% и ее амплитуды
% для интервалов анализа длительностью
% 1.8, 1.9, 2.0, 2.1 с
```

```
figure
for i=9:12
Tmax=1/f*(10+0.1*(i-1)); % вычисление длительности
                         % интервала анализа
  Freq(j-1)=(j-1)/Tmax; % вычисление частот
                        % базисных функций
                        % преобразования Фурье
  dt=Tmax/(L-1); % вычисление шага
                 % временной сетки
  t=dt*(k-1);
  u=sin(2*pi*f*t); % вычисление мгновенных значений
                   % дискретного сигнала
  S=fft(u); % вычисление спектра
  % визуализация спектра
  S1(j-1)=abs(S(j))/(L/2);
  [q1 q2]=max(S1);
  Fm(i,1)=Tmax*f-10;
  Fm(i,2)=Freq(q2);
  Fm(i,3)=q1;
  subplot(4,1,i-8);
  stem(Freq,S1);
  axis([1 20 0 1])
end;

% вычисление спектров,
% частот спектральных гармоник,
% имеющих максимальную амплитуду,
% и зависимостей частоты спектральной
% гармоники с максимальной амплитудой
% и ее амплитуды
% для интервалов анализа длительностью
% 2.2, 2.3, 2.4, 2.5 с
figure
for i=13:16
      Tmax=1/f*(10+0.1*(i-1)); % вычисление длительности
                               % интервала анализа
      Freq(j-1)=(j-1)/Tmax; % вычисление частот базисных функций
                            % преобразования Фурье
  dt=Tmax/(L-1); % вычисление шага
                 % временной сетки
  t=dt*(k-1);
  u=sin(2*pi*f*t); % вычисление мгновенных значений
                   % дискретного сигнала
  S=fft(u); % вычисление спектра
```

```
  % визуализация спектра
  S1(j-1)=abs(S(j))/(L/2);
  [q1 q2]=max(S1);
  Fm(i,1)=Tmax*f-10;
  Fm(i,2)=Freq(q2);
  Fm(i,3)=q1;
  subplot(4,1,i-12);
  stem(Freq,S1);
  axis([1 20 0 1])
end;

% визуализация зависимостей
% частоты спектральной гармоники,
% имеющей максимальную амплитуду,
% и амплитуды
% от длительности интервала анализа
figure;
plot(Fm(:,1),Fm(:,2),'Marker','o','MarkerSize',2)

figure;
plot(Fm(:,1),Fm(:,3),'Marker','o','MarkerSize',2)
```

Таким образом, проведенный нами анализ позволяет сделать следующий вывод: необходимым условием совпадения значений частоты и амплитуды гармонического монохроматического сигнала, оцениваемых по спектру сигнала, с их истинными значениями может быть записано в следуюшем виде:

$$f_c T_c = f\,N\Delta t = M, \tag{7.69}$$

где f_c — частота сигнала, T_c — длительность интервала анализа, N — число узлов временной сетки, Δt — период дискретизации сигнала, M — произвольное целое число.

Для аналитического обоснования полученного результата вспомним о связи между ДПФ

$$S_d(r\Delta\omega) = \sum_{n=0}^{N-1} s_d(n\Delta t)\exp(-j\frac{2\pi}{N}rn) \tag{7.70}$$

и дискретно-непрерывным преобразованием Фурье

$$S(\omega) = \sum_{n=0}^{N-1} s(n\Delta t)\exp(-j\omega\, n\Delta t). \tag{7.71}$$

Подставляя (7.70) в (7.71), получим

$$\begin{aligned} S(\omega) = \frac{A}{2}\left\{\sum_{n=0}^{N-1}\exp[-j(\omega-\omega_0)n\Delta t] + \sum_{n=0}^{N-1}\exp[-j(\omega+\omega_0)n\Delta t]\right\} = \\ = \frac{1}{2}\left(\Sigma_1 + \Sigma_2\right), \end{aligned} \tag{7.72}$$

где ω_0 — частота анализируемого гармонического монохроматического сигнала, A — его амплитуда.

Здесь выражение

$$\Sigma_1 = 1 + \exp[-j(\omega - \omega_0)\Delta t] + \exp[-j(\omega - \omega_0)2\Delta t] + \\ + \exp[-j(\omega - \omega_0)(N-1)\Delta t] \tag{7.73}$$

представляет собой сумму членов геометрической прогрессии

$$a_i = a_0 g^i, \quad i = 0, 1, \ldots,$$

(a_0 — начальный член, g — знаменатель прогрессии), вычисляемую по известной формуле:

$$s_m = \sum_{i=0}^{m} a_i = \frac{a_0 - a_m g}{1 - g}. \tag{7.74}$$

Подставляя в (7.74)

$$\begin{aligned} &a_0 = 1, \quad m = N - 1, \\ &g = \exp[-j(\omega - \omega_0)\Delta t], \\ &a_m = \exp[-j(\omega - \omega_0)(N-1)\Delta t], \end{aligned}$$

получим

$$\begin{aligned} \Sigma_1 &= \frac{1 - \exp[-j(\omega - \omega_0)N\Delta t]}{1 - \exp[-j(\omega - \omega_0)\Delta t]} = \\ &= \frac{\exp[-j(\omega - \omega_0)\frac{N}{2}\Delta t]}{\exp[-j(\omega - \omega_0)\frac{\Delta t}{2}]} \cdot \frac{\exp[j(\omega - \omega_0)\frac{N}{2}\Delta t] - \exp[-j(\omega - \omega_0)\frac{N}{2}\Delta t]}{\exp[j(\omega - \omega_0)\frac{\Delta t}{2}] - \exp[-j(\omega - \omega_0)\frac{\Delta t}{2}]} = \\ &= \exp[-j(\omega - \omega_0)\frac{(N-1)}{2}\Delta t] \cdot \frac{\sin[(\omega - \omega_0)\frac{N}{2}\Delta t]}{\sin[(\omega - \omega_0)\frac{\Delta t}{2}]}. \end{aligned} \tag{7.75}$$

Аналогично,

$$\Sigma_2 = \exp[-j(\omega + \omega_0)\frac{(N-1)}{2}\Delta t] \cdot \frac{\sin[(\omega + \omega_0)\frac{N}{2}\Delta t]}{\sin[(\omega + \omega_0)\frac{\Delta t}{2}]}. \tag{7.76}$$

Из (7.76) с учетом (7.75), (7.76) получаем выражение для модуля преобразования Фурье дискретизированного гармонического сигнала:

$$|S_d(\omega)| = \frac{A}{2}\left|\frac{\sin[(\omega - \omega_0)\frac{N}{2}\Delta t]}{\sin[(\omega - \omega_0)\frac{\Delta t}{2}]} + \exp[-j\omega_0(N-1)\Delta t] \cdot \frac{\sin[(\omega + \omega_0)\frac{N}{2}\Delta t]}{\sin[(\omega + \omega_0)\frac{\Delta t}{2}]}\right|. \tag{7.77}$$

Из (7.77) видно, что основная причина «растекания спектра» состоит в том, что частоты базисных функций преобразования Фурье, в том случае, когда на интервале анализа укладывается нецелое число периодов анализируемого сигнала, таковы, что

анализируемый сигнал оказывается неортогональным базисным функциям. Как следствие, отличными от нуля оказываются амплитуды не одной, но нескольких гармоник.

Вводя безразмерные переменные

$$r = f/\Delta f = fT = fN\Delta t, \quad r_0 = f_0/\Delta f = f_0 T = f_0 N\Delta t, \tag{7.78}$$

получаем выражение, удобное для построения графиков:

$$|X_d(r)| = \frac{A}{2}\left|\frac{\sin[\pi(r-r_0)]}{\sin\left[\pi\frac{(r-r_0)}{N}\right]} + \exp\left[-j2\pi r_0\left(1-\frac{1}{N}\right)\right]\cdot\frac{\sin[\pi(r+r_0)]}{\sin\left[\pi\frac{(r+r_0)}{N}\right]}\right|. \tag{7.79}$$

***Задача* 7.12**

Используя соотношения (7.78), (7.79), *постройте спектр преобразования Фурье рассмотренного в данном разделе гармонического монохроматического сигнала.*

***Задача* 7.13**

Создайте функцию, возвращающую мгновенные значения дискретного сигнала, который описывается функцией вида

$$f(t) = A_1 \sin(\omega_1 t + \varphi_1) + A_2 \sin(\omega_2 t + \varphi_2), \tag{7.80}$$

где $\omega_2 = \omega_1 + \Delta\omega$.

Для функции (7.80) определите:

1. *Длительность интервала анализа, при которой оказывается возможным разрешить спектральные составляющие данного сигнала.*
2. *Оцените точность оценки частоты, амплитуды и фазы каждой составляющей на выбранном измерительном интервале.*

ЛИТЕРАТУРА К ГЛАВЕ 7

1. *Хемминг Р. В.* Цифровые фильтры. М.: Советское радио, 1980.
2. *Баскаков С. И.* Радиотехнические цепи и сигналы. М.: Высшая школа, 1988.
3. *Марпл С. Л.* Цифровой спектральный анализ и его приложения. М.: Мир, 1990.
4. *Оппенгейм А. В., Шафер Р. В.* Цифровая обработка сигналов. М.: Радио и связь, 1979.
5. *Рабинер Л., Голд Б.* Теория и применение цифровой обработки сигналов. М.: Мир, 1978.
6. *Бендат Дж., Пирсол А.* Прикладной анализ случайных данных. М.: Мир, 1989.
7. *Джекинс Г., Ваттс Д.* Спектральный анализ и его приложения. М.: Мир, вып. 1, 1971, вып. 2, 1972.

Глава 8

МОДЕЛИРОВАНИЕ КОЛЕБАТЕЛЬНЫХ ПРОЦЕССОВ

8.1. ЛИНЕЙНЫЙ ГАРМОНИЧЕСКИЙ ОСЦИЛЛЯТОР

Один из наиболее распространенных типов движения механических систем представляет собой малые колебания, которые система совершает вблизи положения устойчивого равновесия. Наиболее простым с математической точки зрения оказывается описание движения систем, имеющих одну степень свободы. Примером такой системы является тело массой m, расположенное на абсолютно гладкой горизонтальной поверхности и прикрепленное к свободному концу пружины жесткостью k (рис. 8.1).

Рис. 8.1. К построению модели линейного гармонического осциллятора

Будем описывать положение тела координатой x и примем за начало отсчета ($x = 0$) точку равновесия груза на пружине. При смещении тела из положения равновесия $x = 0$ на небольшое расстояние x, со стороны пружины на тело будет действовать возвращающая сила, пропорциональная величине смещения и направленная в сторону положения равновесия:

$$F = -kx. \tag{8.1}$$

Знак минус в (8.1) указывает на то, что сила стремится вернуть тело в положение равновесия.

На основании второго закона Ньютона

$$m\ddot{x} = -kx. \tag{8.2}$$

Разделив обе части уравнения на массу m и введя обозначение

$$\omega_0 = \sqrt{\frac{k}{m}}, \tag{8.3}$$

запишем уравнения движения линейного гармонического осциллятора в виде

$$\ddot{x} + \omega_0^2 x = 0, \tag{8.4}$$

где ω_0 называется циклической частотой.

Уравнение (8.4) является линейным дифференциальным уравнением второго порядка. Аналитическое решение данного уравнения хорошо известно:

$$x\left(t\right) = A\cos\left(\omega_0 t + \delta\right), \tag{8.5}$$

где A, δ — постоянные, называемые амплитудой и начальной фазой, определяются из начальных условий для координаты и скорости:

$$x_0 = x(0),$$
$$v_0 = v(0).$$

По определению период периодической функции есть наименьшее время, через которое движение повторяется, то есть

$$x(t+T) = x(t). \tag{8.6}$$

Так как период функции косинус равен 2π, то величина $\omega_0 T$ соответствует одному периоду колебания:

$$\omega_0 T = 2\pi,$$

откуда находим связь между ω и T:

$$T = \frac{2\pi}{\omega_0} = \frac{2\pi}{\sqrt{k/m}} = 2\pi\sqrt{\frac{m}{k}}. \tag{8.7}$$

Частота колебаний ν представляет собой число периодов в одну секунду и определяется выражением

$$\nu = \frac{1}{T}. \tag{8.8}$$

Необходимо отметить, что в соответствии с (8.7) период колебаний зависит от отношения k/m, но не зависит от A и δ. Это означает, что период колебаний линейного гармонического осциллятора не зависит от амплитуды колебаний.

Легко показать, что полная энергия линейного гармонического осциллятора E

$$E = \frac{1}{2}mv^2 + \frac{1}{2}kx^2, \tag{8.9}$$

где первое слагаемое — это кинетическая энергия, второе слагаемое — потенциальная энергия.

Предваряя рассмотрение более сложных случаев периодического движения физических систем, уравнения движения большинства из которых не допускают аналитического решения, продемонстрируем общий подход к нахождению и анализу численных решений уравнения движения на примере линейного гармонического осциллятора (8.4). Вычисленные зависимости $x = x(t)$, $v = v(t)$ полностью описывают движение одномерного линейного гармонического осциллятора. С другой стороны, пару значений функций $\{x(t_n), y(t_n)\}$ в выбранный момент времени можно рассматривать как координаты некоторой точки в двумерном пространстве, называемом фазовой плоскостью. Кривая, образованная на фазовой плоскости, при нанесении точек $\{x(t_i), y(t_i)\}$ в последовательные моменты времени называется фазовой траекторий.

Для нахождения численного решения уравнения движения линейного гармонического осциллятора (8.4) и их визуализации следует привести ДУ второго порядка к системе ДУ первого порядка, затем создать m-файл, содержащий определение функции, стоящей в правой части системы ДУ. Ниже приводится листинг соответствующего файла `Oscillator.m`.

```
% Листинг файла Oscillator.m
function dy=Oscillator(t,z);
% функция, стоящая в правой части
системы ДУ первого порядка,
% эквивалентной ДУ (8.4)
global omega;
dy=zeros(2,1); % задание вектора-столбца
               % размерности 2x1
dy(1)=z(2);
dy(2)=-omega^2*z(1);
```

Далее необходимо выполнить следующую последовательность команд, сохраненную нами в файле `Glava8_1.m`:

```
% листинг файла Glava8_1.m
global omega;
k=9; % коэффициент жесткости
m=1; % масса осциллятора
T=2*pi*(k/m)^0.5; % период колебаний
omega=2*pi/T; % циклическая частота
R0=[0.5 1]; % начальные условия
N=10^4; % число узлов
        % временной сетки

t Z]=ode45('Oscillator',[0:5*T/N:5*T],R0); % решение системы ДУ

figure(1);
plot(t,Z(:,1)); % визуализация
                % зависимости x=x(t)
figure(2);
plot(t,Z(:,2)); % визуализация
                % зависимости v=v(t)
figure(3); plot(Z(:,1),Z(:,2)); % визуализация фазовой плоскости

figure(4);comet(Z(:,1),Z(:,2)); % динамическая визуализация
                                % движения на фазовой плоскости
```

Результаты выполнения приведенной последовательности команд представлены на рис. 8.2—8.4.

Рис. 8.2. Зависимость координаты гармонического осциллятора от времени

Рис. 8.3. Зависимость скорости гармонического осциллятора от времени

Рис. 8.4. Фазовая траектория гармонического осциллятора

Задача 8.1

1. *Вычислите полную энергию гармонического осциллятора для тех моментов времени, в которые известны значения координаты и скорости. Постройте график временной зависимости величины* $\Delta_n = (E_n - E_0)/E_0$. *Как меняется данная величина во времени?*

2. *Постройте график зависимости отклонения между численными значениями координаты и скорости и аналитическими. Как меняются данные величины во времени? Оцените точность полученного численного решения по координате и по скорости.*

3. *Дополните описанный выше документ блоком, позволяющим проводить спектральный анализ численных решений ДУ (8.4): вычислите спектр зависимостей* $x = x(t)$, $v = v(t)$, *постройте графики спектральных плотностей* $S_x(f), S_y(f)$ *и определите, используя данные графики, значения циклической частоты* $\omega_0 = 2\pi f_0$ *и периода колебаний* T.

4. *Для различных значений ω_0 вычислите, соответствующее им значение периодов T. Предположив, что величина T пропорциональна $(k/m)^\alpha$, оцените показатель степени, построив график зависимости $T = T(k/m)$ в дважды логарифмическом масштабе.*

5. *Вычислите значения амплитуды A и полной энергии E для начальных условий $x(0) = 4$, $v(0) = 0$ и $x(0) = 0$, $v(0) = 4$, выбрав в обоих случаях значения k, m, при которых $\omega_0 = 2$. Какая величина определяет значение амплитуды A?*

6. *Постройте зависимости потенциальной и кинетической энергий от времени. В какие моменты времени эти зависимости достигают своего наибольшего и наименьшего значений? В каких точках находится в данный момент осциллятор?*

7. *Вычислите и сравните средние значения кинетической и потенциальной энергий за один период колебания.*

8. *Вычислите зависимости $x(t)$ для различных значений амплитуды A и покажите, что форма данной функции и ее период не зависят от A при фиксированном значении отношения k/m. В каких единицах необходимо измерять время для того, чтобы величина $x(t)/A$ не зависела от A и k/m? Как будет выглядеть уравнение движения линейного гармонического осциллятора (8.4) в данном случае?*

9. *Как видно из рис. 8.4, фазовая траектория линейного гармонического осциллятора является эллипсом. Дополните документ блоком, позволяющим вычислять размеры его большой и малой полуосей. Будут ли отличаться фазовые траектории для различных начальных условий? Какие физические величины определяют размеры большой и малой полуосей эллипса? Запишите аналитическое уравнение эллипса, проведите его анализ и сравните свои выводы с результатами, полученными при численных расчетах.*

***Задача* 8.2**

Уравнения, описывающие поведение линейного гармонического осциллятора, получены в предположении малости амплитуды колебаний, что позволяет записать силу, возвращающую осциллятор в положение равновесия в виде (8.1). Для описания движения с конечными амплитудами выражение для возвращающей силы будет содержать члены, пропорциональные второй и более высоким степеням x, а уравнение движения примет следующий вид:

$$\ddot{x} + \omega_0^2 x = -\alpha\, x^2. \tag{8.10}$$

Проведите сравнительный анализ решений уравнения (8.10) при различных значениях параметра a и решений уравнения (8.4), используя зависимости $x = x(t)$, $v = v(t)$, фазовую траекторию и график спектральной плотности зависимости $x = x(t)$.

8.2. МАТЕМАТИЧЕСКИЙ МАЯТНИК

Рассмотрим движение груза массой m, прикрепленного к одному из концов жесткого стержня длиной L, другой конец которого закреплен в точке подвеса (рис. 8.5).

Такая система, как известно из опыта, будучи выведенной из положения равновесия будет совершать колебания. Для описания движения данной системы в первом приближении используют модель, в которой пренебрегают трением в точке подвеса, массой стержня по сравнению с массой груза и считают, что вся масса груза приложена в одной точке.

Данная модель называется математическим маятником. Так как движение груза происходит по дуге окружности радиуса L с центром в точке O, то положение груза характеризуется углом отклонения стержня о вертикали θ. При движении по окружности линейная скорость и ускорение груза равны

$$v = L\frac{d\theta}{dt}, \tag{8.11}$$

$$a = L\frac{d^2\theta}{dt^2}. \tag{8.12}$$

Рис. 8.5. Математический маятник

В используемой модели на математический маятник действуют две силы: сила тяжести $m\vec{g}$, направленная вертикально вниз, и сила реакции стержня (рис. 8.5). Равнодействующая этих сил, равная, как видно из рис. 8.5, $-mg\sin\theta$, направлена в сторону уменьшения угла θ.

Следовательно, уравнение движения математического маятника записывается в виде:

$$mL\frac{d^2\theta}{dt^2} = -mg\sin\theta, \tag{8.13}$$

или

$$\frac{d^2\theta}{dt^2} = -\frac{g}{L}\sin\theta. \tag{8.14}$$

В общем случае уравнение (8.14) оказывается нелинейным. Его решение, как и решения большинства нелинейных уравнений, не выражается через элементарные функции. Отмеченная особенность (8.14) определяет необходимость использования для его решения численных методов.

Однако при достаточно малых углах, при которых $\sin\theta \approx \theta$, уравнение (8.14) становится линейным

$$\frac{d^2\theta}{dt^2} = -\frac{g}{L}\theta. \tag{8.15}$$

Вводя обозначение

$$\omega_0 = \sqrt{\frac{g}{L}} \tag{8.16}$$

и сравнивая уравнения (8.4) и (8.13), видим, что они оказываются одинаковыми. Следовательно, результаты, полученные в разделе 8.1 для линейного гармонического осциллятора, могут быть использованы при описании математического маятника. В частности, период малых колебаний математического маятника T

$$T = \frac{2\pi}{\omega_0} = 2\pi\sqrt{\frac{L}{g}} \tag{8.17}$$

не зависит от амплитуды колебаний.

Получим выражение для полной энергии математического маятника, являющейся интегралом движения. Как видно из рис. 8.5, потенциальная энергия математического маятника U, отсчитываемая от точки равновесия маятника, равна

$$U = mgh = mgL\left(1 - \cos\theta\right). \tag{8.18}$$

Кинетическая энергия маятника равна

$$\frac{1}{2}mv^2 = \frac{1}{2}mL^2\left(\frac{d\theta}{dt}\right)^2,$$

поэтому полная энергия маятника задается следующим выражением:

$$E = \frac{1}{2}mL^2\left(\frac{d\theta}{dt}\right)^2 + mgL\left(1 - \cos\theta\right). \tag{8.19}$$

Уравнение (8.19) позволяет получить формулу, связывающую период колебания математического маятника и угол начального отклонения. Для этого разрешим уравнение (8.19) относительно $d\theta/dt$:

$$\frac{d\theta}{dt} = \pm\sqrt{\frac{2}{mL^2}\left(E - mgL\left(1 - \cos\theta\right)\right)}. \tag{8.20}$$

Из (8.20) видно, что переменные разделяются:

$$dt = \pm\frac{d\theta}{\sqrt{\frac{2}{mL^2}\left(E - mgL\left(1 - \cos\theta\right)\right)}}. \tag{8.21}$$

Интегрируя (8.21), находим выражение для периода колебаний математического маятника:

$$T = 2\int\limits_{-\theta_0}^{\theta_0}\frac{d\theta}{\sqrt{\frac{2}{mL^2}\left(E - mgL\left(1 - \cos\theta\right)\right)}}, \tag{8.22}$$

где θ_0 — начальный угол отклонения маятника.

При движении маятника из начального положения с нулевой начальной скоростью $E = mgL\left(1 - \cos\theta_0\right)$, поэтому

$$T = \sqrt{\frac{2L}{g}}\int\limits_{-\theta_0}^{\theta_0}\frac{d\theta}{\sqrt{\cos\theta - \cos\theta_0}}. \tag{8.23}$$

***Задача* 8.3**

Создайте в MATLAB файл, позволяющий находить численное решение уравнения движения математического маятника (8.14). Получите численное решение уравнения движения при различных начальных углах отклонения и проведите их сравнительный анализ, используя зависимости $x = x(t)$, $v = v(t)$, фазовую траекторию и график спектральной плотности зависимости $x = x(t)$.

1. *Сравните спектральные плотности при малых и конечных углах отклонения маятника от вертикали. В чем проявляется отличие линейных и нелинейных колебаний математического маятника?*

2. *Используя графики спектральных плотностей $S_x(f)$ зависимости $x = x(t)$, постройте зависимость периода колебаний математического маятника от начального угла отклонения? Качественно объясните получаемые результаты.*

3. *Используя формулу (8.23), постройте зависимость периода колебаний математического маятника от начального угла отклонения θ_0. Сравните полученный результат с аналогичными результатами, полученными при анализе спектров $S_x(f)$.*

4. *Как выглядят фазовые траектории математического маятника при различных значениях начального угла отклонения?*

Возможности MATLAB позволяют не только получить количественные характеристики движения математического маятника, являющиеся в известной мере статическими, но и получить динамическую картинку, демонстрирующую его движение.

Для создания анимации движения математического маятника необходимо в качестве первого шага вычислить соответствующие декартовы координаты по известной зависимости угла отклонения маятника от времени, хранящейся в матрице Z, возвращенной функцией `ode45`. Как видно из рис. 8.6, при нахождении маятника в первом, втором, третьем и четвертом квадрантах координаты x, y вычисляются по следующим формулам, соответственно:

Рис. 8.6. К вычислению декартовых координат математического маятника

1. $x(t) = L\sin(\pi - \theta(t)), y(t) = L\cos(\pi - \theta(t),$

2. $x\,(t) = L\sin\,(\pi - \theta\,(t)), y\,(t) = L\cos\,(\pi - \theta\,(t)),$

3. $x(t) = -L\sin\theta(t), y(t) = -L\cos\theta(t)$,

4. $x(t) = L\sin\theta(t), y(t) = -L\cos\theta(t)$.

Для создания анимационного клипа необходимо выполнить следующую последовательность команд, сохраненную нами в файле `Glava8_2.m`:

```
% листинг файла Glava8_2.m
clear all  % очистка рабочей области

global omega
g=9.8; % ускорение свободного падения
L=1; % длина маятника
T=2*pi*(g/L)^0.5; % период колебаний
omega=2*pi/T; % циклическая частота
phi0=pi*0.995; % начальный угол
R0=[phi0 0]; % начальные условия

N=5000; % число узлов временной сетки
[t Z]=ode45('Oscillator',[0:3*T/N:3*T],R0); % решение уравнения
                                            % движения
% вычисление декартовых координат маятника
 for i=1:N+1
         if Z(i,1)>pi/2
            S1(i,1)=L*cos(Z(i,1)-pi/2);
            S1(i,2)=L*sin(Z(i,1)-pi/2);
        end;
        if Z(i,1)<-pi/2
           S1(i,1)=-L*cos(abs(Z(i,1))-pi/2);
           S1(i,2)=L*sin(abs(Z(i,1))-pi/2);
        end;
        if (-pi/2<=Z(i,1))&(Z(i,1)<=pi/2)
           S1(i,1)=L*sin(Z(i,1));
           S1(i,2)=-L*cos(Z(i,1));
        end;
 end;

% визуализация мгновенных значений
% координат маятника (рис. 8.7)
figure(1);
plot(t,S1(:,1),t,S1(:,2));
% координаты подвеса
Sa(1,1)=0;
Sa(1,2)=0;
% координаты маятника
Sa(2,1)=S1(1,1);
Sa(2,2)=S1(1,2);
```

```
% отображение маятника
% в момент времени t=0 (рис. 8.8)
figure(2);
plot(Sa(:,1),Sa(:,2),Sa(2,1),Sa(2,2),'o');
axis([-1.2 1.2 -1.2 1.2]);
set(gca,'nextplot','replacechildren'); % включение режима
                                       % перерисовки графиков
                                       % в одном и том же окне
k=1;

% создание анимационного клипа
for i=1:100:length(S1)
        Sa(2,1)=S1(i,1);
        Sa(2,2)=S1(i,2);
        C=plot(Sa(:,1),Sa(:,2),Sa(2,1),Sa(2,2),'o');
        F(k)=getframe;
        k=k+1;
    end;
movie(F,1) % однократное воспроизведение клипа
```

Результат выполнения приведенной выше последовательности команд представлен на рис. 8.7, 8.8.

Рис. 8.7. Зависимости $x = x(t),\ y = y(t)$

Рис. 8.8. Зависимости $x = x(t),\ y = y(t)$

***Задача* 8.4**

Проанализируйте зависимости $x = x(t)$, $y = y(t)$ и убедитесь в правильности работы программы, вычисляющей данные зависимости.

***Задача* 8.5** *Дополните созданный ранее файл блоком, позволяющим одновременно анимировать движения линейного и нелинейного математических маятников.*

8.3. ЗАТУХАЮЩИЕ КОЛЕБАНИЯ

Из опыта известно, что в большинстве реальных колебательных систем присутствует трение. Это приводит к тому, что в отсутствие подкачки энергии в колебательную систему, амплитуда колебаний уменьшается до ее полной остановки. Такие колебания называются затухающими. В качестве примера колебательной системы, совершающей гармонические колебания, рассмотрим движение груза на пружине (рис. 8.1). При движении с малыми скоростями, как было подробно обсуждено в главе 3, можно использовать модель, в которой сила трения пропорциональна первой степени скорости движения. В этом случае уравнения движения можно записать в виде:

$$\frac{d^2x}{dt^2} = -\omega_0^2 x - \gamma \frac{dx}{dt}, \tag{8.24}$$

где γ — коэффициент затухания.

***Задача* 8.6**

Создайте в MATLAB m-файл, содержащий описание функции, стоящей в правой части системы дифференциальных уравнений, эквивалентной уравнению (8.24).

1. *Проведите расчеты и постройте зависимости для колебательной системы с параметрами* $\omega_0 = 3, x(0) = 1, v(0) = 0, \gamma = 0,5$.

2. *Используя спектр функции* $x = x(t)$, *определите частоту и период колебаний осциллятора. Сравните спектры и частоту колебаний со случаем отсутствия затухания. Проведите расчеты для нескольких значений* $\gamma \in [0{,}5; 4]$ *и постройте график зависимости частоты колебаний от коэффициента затухания. Попробуйте определить аналитическое выражения данной функции.*

3. *Можно определить амплитуду как максимальное значение функции* $x = x(t)$ *на одном периоде колебаний. Вычислите время релаксации* τ, *определяемое как время, за которое амплитуда колебаний уменьшается в* $e \approx 2,72$ *раза относительно максимального значения. Покажите, что значение* τ *неизменно в течение всего времени колебаний. Проведите расчеты для нескольких значений* $\gamma \in [0{,}5; 4]$ *и постройте график зависимости времени релаксации от коэффициента затухания. Качественно объясните полученную зависимость.*

4. *Постройте график зависимости полной энергии от времени для различных значений коэффициента затухания. Если энергия маятника уменьшается не монотонно, объясните причину это явления.*

5. *Вычислите средние значения кинетической, потенциальной и полной энергий по полному периоду. Постройте графики зависимости этих средних от номера периода. Проведите сравнительный анализ временных зависимостей этих средних от* τ *при различных значениях коэффициента затухания* γ.

6. *Проведите вычисления и постройте зависимости $x = x(t)$, $y = y(t)$ для $\gamma = 4, 5, 6, 7$ и 8. Будет ли осциллятор совершать колебательные движения для всех значений g? Определите положение равновесия как точку, в которой амплитуда колебаний становится меньше некоторого заданного значения, например 10^{-3}. Дополните использованную вами последовательность команд командами, позволяющими вычислять время достижения равновесия. Постройте зависимость времени достижения равновесия от коэффициента затухания.*

7. *Осциллятор называется критически затухающим при данном значении ω_0, если γ равно наименьшему значению, при котором переход к положению равновесия происходит монотонно. Для каких значений γ будет иметь место критическое затухание при $\omega_0 = 2$, $\omega_0 = 3$?*

8. *Постройте фазовые диаграммы для случаев $\omega_0 = 2$, $\gamma = 0{,}5, 2, 4, 6, 8$. Проанализируйте особенности фазовых диаграмм при различных величинах коэффициента затухания и объясните причину их качественных отличий.*

8.4. ВЫНУЖДЕННЫЕ КОЛЕБАНИЯ ЛИНЕЙНОГО ГАРМОНИЧЕСКОГО ОСЦИЛЛЯТОРА

При воздействии на линейный гармонический осциллятор внешней переменной силы, равной $F(t)$, система будет совершать движения, которые в отличие от рассмотренных в предыдущих параграфах называются вынужденными колебаниями. Соответствующее уравнение движения имеет вид

$$\ddot{x} = -\omega_0^2 x - \gamma \frac{dx}{dt} + \frac{F(t)}{m}. \tag{8.25}$$

***Задача* 8.7**

Создайте в MATLAB файл, позволяющий находить численное решение уравнения (8.25) для силы вида

$$\frac{F(t)}{m} = A_0 \cos(\omega t). \tag{8.26}$$

1. *Проведите расчеты и постройте зависимости $x = x(t)$, $y = y(t)$ для колебательной системы с параметрами $\omega_0 = 3$, $x(0) = 1$, $v(0) = 0$, $\gamma = 0{,}5$, $A_0 = 1$, $\omega = 2$. Сравните полученные результаты с результатами предыдущей задачи. В чем состоит качественное отличие поведения функции от случая невозмущенного движения? Чему равны период и частота функции после нескольких колебаний?*

2. *Проведите расчеты и постройте зависимости $x = x(t)$, $v = v(t)$ для начальных условий $x(0) = 0$, $v(0) = 1$. Чему равны период и частота функции после нескольких колебаний? Существует ли у функции предельная*

форма, которая не зависит от начальных условий? Зависит ли поведение функции от начальных условий на коротких временах? Выделите переходную часть функции, которая зависит от начальных условий и затухает со временем, и установившуюся часть, определяющую поведение функции на больших временах и независящую от начальных условий.

3. *Вычислите и сравните спектры исходной функции $x = x(t)$, части функции $x = x(t)$, соответствующей переходному периоду, части функции $x = x(t)$, соответствующей установившемуся движению.*

4. *Вычислите значения функции $x = x(t)$ и ее спектр $S_x = S_x(f)$ для $\omega_0 = 1$ и $\omega_0 = 4$. Чему равны в каждом случае период и частота установившихся колебаний функции $x = x(t)$? Какие параметры влияют на частоту установившихся колебаний?*

5. *Убедитесь в том, что установившееся движение описывается формулой*

$$x\,(t) = A\,(\omega)\cos\,(\omega\,t + \delta)\,,$$

где δ — разность фаз между приложенной силой и установившимися колебаниями. Вычислите δ для $\omega_0 = 3$, $\gamma = 0{,}5$, и значений $\omega = 0$; 1,0; 2,0; 2,2; 2,4; 2,6; 2,8; 3,0; 3,2; 3,4. Повторите расчет для $\gamma = 1,5$ и постройте график зависимости $\delta(\omega)$ для обоих значений γ. Дайте качественное объяснение полученной зависимости. (Указание: для оценки δ используйте спектр стационарной части функции $x = x(t)$.)

6. *Постройте и проанализируйте фазовую диаграмму для случаев $\omega_0 = 3$, $\gamma = 0{,}5; 2; 4; 6; 8$. Как выглядят фазовые диаграммы при различных γ? Объясните обнаруженные отличия.*

***Задача* 8.8**

Результаты предыдущей задачи показывают, что в установившемся режиме характер колебаний линейного гармонического осциллятора с трением зависит от частоты приложенной к нему силы. Одной из характеристик этих колебаний является их амплитуда, являющаяся некоторой функцией частоты внешней силы.

1. *Для начальных условий $x(0) = 0, v(0) = 0$ вычислите значения функции $A(\omega)$, называемой амплитудно-частотной характеристикой, для $\omega = 0$, 1,0, 2,0, 2,2, 2,4, 2,6, 2,8, 3,0, 3,2, 3,4, 3,6, 3,8, 4,0, 5,0, 6,0 при $\omega_0 = 5, \gamma = 0,5$. Постройте график функции $A(\omega)$ и качественно объясните поведение этой функции. При наличии у функции $A(\omega)$ максимума определите соответствующее данной точке значение частоты ω_m (резонансная частота). Сравните значение собственной частоты гармонического осциллятора ω_0 и резонансной частоты ω_m.*

2. *Вычислите значение функции $A(\omega)$ в точке $\omega = \omega_m$ и отношение $\Delta\omega/\omega_m$, где $\Delta\omega$ является «шириной» резонанса. (Величина $\Delta\omega$ определяется как*

ширина интервала частот между точками амплитудно-частотной характеристики, равными $A_m/\sqrt{2}$.)

Проведите вычисления при $\omega_0 = 0{,}1, 0{,}5, 1{,}0, 2{,}0$ и оцените вид зависимостей $A_m = A_m(\gamma)$ и $\Delta\omega/\omega_m = \Delta\omega/\omega_m(\gamma)$.

3. *Вычислите среднюю за период колебаний кинетическую энергию осциллятора в установившемся режиме при указанных выше параметрах колебательной системы и внешней силы. Сравните резонансную частоту кинетической энергии с резонансной частотой амплитуды.*

***Задача* 8.9**

Исследуйте и опишите особенности движения линейного гармонического осциллятора под влиянием силы $F(t)$:

1) $F(t) = \mathrm{const} = F_0$;

2) $F(t) = A\sin(\beta t)$;

3) $F(t) = at$;

4) $F(t) = F_0 e^{-\alpha t}$;

5) $F(t) = F_0 e^{-\alpha t}\cos(\beta t)$,

считая, что в начальный момент времени система находится в положении равновесия ($x(0) = 0$, $y(0) = 0$). Рассмотрите случаи, когда $a, \beta \ll \omega_0$, $a, \beta \cong \omega_0$, $\beta = \omega_0$, $a, \beta \gg \omega_0$.

***Задача* 8.10**

Демонстрация принципа суперпозиции в динамике линейных систем.

1. *Рассмотрите движение линейного гармонического осциллятора с трением под действием периодической силы (8.25) при $\omega_0 = 3$, $x(0) = 0$, $v(0) = 0$, $\gamma = 0,5$, $\omega = 2$ для двух случаев $A_0 = 1$, $A_0 = 2$. Как соотносятся зависимости $x = x(t)$ в установившихся режимах? Какой результат будет получен при $A_0 = 4$? Удовлетворяют ли переходные режимы для различных внешних сил тем же соотношениям, что и установившиеся режимы?*

2. *Вычислите отклики на внешние силы F_1, F_2, F_3 и $F_1 + F_2 + F_3$ при $\omega = 2$, $A_0 = 0,5$, $\omega = 4$, $A_0 = 0,25$, $\omega = 6$, $A_0 = 0,125$. Сравните спектры установившихся режимов функций $x = x(t)$ и сформулируйте принцип суперпозиции для решений линейных уравнений.*

3. *Проведите расчеты, аналогичные расчетам в п. 2, для математического маятника, движение которого описывается уравнением (8.14). Применим ли принцип суперпозиции к нелинейным системам?*

8.5. ДВОЙНОЙ ПЛОСКИЙ МАЯТНИК

Рассмотрим особенности движения системы с двумя степенями свободы — двойной плоский маятник с точечными массами m_1 и m_2, изображенный на рис. 8.9, мгновенное положение которого описывается двумя углами φ_1 и φ_2, образуемыми соответственно нитями l_1, l_2 с вертикалью.

Рис. 8.9. Двойной плоский маятник

Тогда кинетическая T_1 и потенциальная энергии точки m_1 вычисляются по следующим формулам:

$$T_1 = \frac{1}{2} m_1 l_1^2 \dot{\varphi}_1^2, \tag{8.27}$$

$$U_1 = -m_1 g l_1 \cos\varphi_1, \tag{8.28}$$

где g — ускорение свободного падения.

Для нахождения кинетической энергии второй точки выразим ее декартовы координаты x_2, y_2 (выбираем начало координат в точке подвеса, ось OY направляем по вертикали вниз) через углы φ_1 и φ_2:

$$x_2 = l_1 \sin\varphi_1 + l_2 \sin\varphi_2, \tag{8.29}$$

$$y_2 = l_1 \cos\varphi_1 + l_2 \cos\varphi_2. \tag{8.30}$$

Продифференцировав (8.29), (8.30) и подставив в известное выражение для кинетической энергии

$$T_2 = \frac{m_2}{2}\left(\dot{x}_2^2 + \dot{y}_2^2\right),$$

получим

$$T_2 = \frac{m_2}{2}\left[l_1^2\dot{\varphi}_1^2 + l_2^2\dot{\varphi}_2^2 + 2 l_1 l_2 \cos\left(\varphi_1 - \varphi_2\right)\dot{\varphi}_1\dot{\varphi}_2\right]. \tag{8.31}$$

Потенциальная энергия второго груза равна:

$$U_2 = -m_2 g y_2 = -m_2 g\left(l_1 \cos\varphi_1 + l_2 \cos\varphi_2\right). \tag{8.32}$$

Учитывая (8.27)–(8.32), получаем окончательное выражение для функции Лагранжа:

$$L = \frac{m_1 + m_2}{2} l_1^2 \dot{\varphi}_1^2 + \frac{m_2}{2} l_2^2 \dot{\varphi}_2^2 + m_2 l_1 l_2 \cos\left(\varphi_1 - \varphi_2\right)\dot{\varphi}_1\dot{\varphi}_2 + \ldots \\ \left(m_1 + m_2\right) g l_1 \cos\varphi_1 + m_2 g l_2 \cos\varphi_2. \tag{8.33}$$

Далее, подставив (8.33) в уравнения Лагранжа

$$\frac{d}{dt}\frac{\partial L}{\partial \dot{q}_i} = \frac{\partial L}{\partial q_i}, \quad i = 1, 2,$$

получим

$$\frac{\partial L}{\partial \dot{\varphi}_1} = (m_1 + m_2)\, l_1^2 \dot{\varphi}_1 + m_2 l_1 l_2 \cos(\varphi_1 - \varphi_2)\, \dot{\varphi}_2,$$

$$\frac{\partial L}{\partial \dot{\varphi}_2} = m_2 l_2^2 \dot{\varphi}_2 + m_2 l_1 l_2 \cos(\varphi_1 - \varphi_2)\, \dot{\varphi}_1,$$

$$\frac{d}{dt}\frac{\partial L}{\partial \dot{\varphi}_1} = (m_1 + m_2)\, l_1^2 \ddot{\varphi}_1 + m_2 l_1 l_2 \cos(\varphi_1 - \varphi_2)\, \ddot{\varphi}_2 - m_2 l_1 l_2 \sin(\varphi_1 - \varphi_2)\, (\dot{\varphi}_1 - \dot{\varphi}_2)\, \dot{\varphi}_2, \tag{8.34}$$

$$\frac{d}{dt}\frac{\partial L}{\partial \dot{\varphi}_2} = m_2 l_2^2 \ddot{\varphi}_2 + m_2 l_1 l_2 \cos(\varphi_1 - \varphi_2)\, \ddot{\varphi}_1 - m_2 l_1 l_2 \sin(\varphi_1 - \varphi_2)\, (\dot{\varphi}_1 - \dot{\varphi}_2)\, \dot{\varphi}_1, \tag{8.35}$$

$$\frac{\partial L}{\partial \varphi_1} = -m_2 l_1 l_2 \sin(\varphi_1 - \varphi_2)\, \dot{\varphi}_1 \dot{\varphi}_2 - (m_1 + m_2)\, g l_1 \sin \varphi_1, \tag{8.36}$$

$$\frac{\partial L}{\partial \varphi_1} = -m_2 l_1 l_2 \sin(\varphi_1 - \varphi_2)\, \dot{\varphi}_1 \dot{\varphi}_2 - (m_1 + m_2)\, g l_1 \sin \varphi_1, \tag{8.37}$$

Приравняв, соответственно, выражения (8.34), (8.36) и (8.35), (8.37), получаем систему ДУ, описывающих движение двойного маятника:

$$\begin{aligned}
&(m_1 + m_2) l_1^2 \ddot{\varphi}_1 + m_2 l_1 l_2 \cos(\varphi_1 - \varphi_2) \ddot{\varphi}_2 - \\
&- m_2 l_1 l_2 \sin(\varphi_1 - \varphi_2)(\dot{\varphi}_1 - \dot{\varphi}_2)\dot{\varphi}_2 = \\
&- m_2 l_1 l_2 \sin(\varphi_1 - \varphi_2)\dot{\varphi}_1 \dot{\varphi}_2 - (m_1 + m_2) g l_1 \sin \varphi_1, \\
&m_2 l_2^2 \ddot{\varphi}_2 + m_2 l_1 l_2 \cos(\varphi_1 - \varphi_2)\ddot{\varphi}_1 - \\
&- m_2 l_1 l_2 \sin(\varphi_1 - \varphi_2)(\dot{\varphi}_1 - \dot{\varphi}_2)\dot{\varphi}_1 = \\
&= m_2 l_1 l_2 \sin(\varphi_1 - \varphi_2)\dot{\varphi}_1 \dot{\varphi}_2 - m_2 g l_2 \sin \varphi_2.
\end{aligned} \tag{8.38}$$

Приведя в (8.38) подобные члены, получаем

$$\begin{aligned}
&(m_1 + m_2) l_1^2 \ddot{\varphi}_1 + m_2 l_1 l_2 \cos(\varphi_1 - \varphi_2) \ddot{\varphi}_2 + \\
&+ m_2 l_1 l_2 \sin(\varphi_1 - \varphi_2)\dot{\varphi}_2^2 = -(m_1 + m_2) g l_1 \sin \varphi_1, \\
&m_2 l_2^2 \ddot{\varphi}_2 + m_2 l_1 l_2 \cos(\varphi_1 - \varphi_2)\ddot{\varphi}_1 - \\
&- m_2 l_1 l_2 \sin(\varphi_1 - \varphi_2)\dot{\varphi}_1^2 = -m_2 g l_2 \sin \varphi_2.
\end{aligned} \tag{8.39}$$

Для удобства нахождения в дальнейшем численного решения системы ДУ (8.39) преобразуем ее, оставив в левой части только вторые производные $\ddot{\varphi}_1, \ddot{\varphi}_2$, к следующему виду:

$$\begin{aligned}
&\begin{bmatrix} (m_1 + m_2)\, l_1 & m_2\, l_2 \cos(\varphi_1 - \varphi_2) \\ m_2 l_1 \cos(\varphi_1 - \varphi_2) & m_2\, l_2 \end{bmatrix} \cdot \begin{bmatrix} \ddot{\varphi}_1 \\ \ddot{\varphi}_2 \end{bmatrix} = \\
&\begin{bmatrix} -m_2\, l_2 \sin(\varphi_1 - \varphi_2)\, \dot{\varphi}_2^2 - (m_1 + m_2)\, g \sin \varphi_1 \\ m_2\, l_1 \sin(\varphi_1 - \varphi_2)\, \dot{\varphi}_1^2 - m_2 g \sin \varphi_2 \end{bmatrix},
\end{aligned} \tag{8.40}$$

далее решим систему (8.40) относительно производных $\ddot{\varphi}_1, \ddot{\varphi}_2$:

$$\begin{aligned}
\ddot{\varphi}_1 &= -\frac{[m_2 l_2 \sin(\varphi_1-\varphi_2)\dot{\varphi}_2^2 + (m_1+m_2) g \sin\varphi_1]}{l_1(m_1+m_2\sin^2(\varphi_1-\varphi_2))} - \\
&- \frac{[m_2 l_1 \sin(\varphi_1-\varphi_2)\dot{\varphi}_1^2 - m_2 g \sin\varphi_2]\cos(\varphi_1-\varphi_2)}{l_1(m_1+m_2\sin^2(\varphi_1-\varphi_2))}, \\
\ddot{\varphi}_2 &= \frac{(l_1 \sin(\varphi_1-\varphi_2)\dot{\varphi}_1^2 - g\sin\varphi_2)(m_1+m_2)}{l_2(m_1+m_2\sin^2(\varphi_1-\varphi_2))} + \\
&+ \frac{(m_2 l_2 \sin(\varphi_1-\varphi_2)\dot{\varphi}_2^2 + (m_1+m_2) g l_1 \sin\varphi_1)\cos(\varphi_1-\varphi_2)}{l_2(m_1+m_2\sin^2(\varphi_1-\varphi_2))},
\end{aligned} \tag{8.41}$$

и наконец, приведем (8.41) к эквивалентной системе ДУ первого порядка:

$$\begin{aligned}
\dot{\varphi}_1 &= z_1, \\
\dot{z}_1 &= -\frac{[m_2 l_2 \sin(\varphi_1-\varphi_2) z_2^2 + (m_1+m_2) g \sin\varphi_1]}{l_1(m_1+m_2\sin^2(\varphi_1-\varphi_2))} - \\
&- \frac{[m_2 l_1 \sin(\varphi_1-\varphi_2) z_1^2 - m_2 g \sin\varphi_2]\cos(\varphi_1-\varphi_2)}{l_1(m_1+m_2\sin^2(\varphi_1-\varphi_2))}, \\
\dot{\varphi}_2 &= z_2, \\
\dot{z}_2 &= \frac{(l_1 \sin(\varphi_1-\varphi_2) z_1^2 - g\sin\varphi_2)(m_1+m_2)}{l_2(m_1+m_2\sin^2(\varphi_1-\varphi_2))} + \\
&+ \frac{(m_2 l_2 \sin(\varphi_1-\varphi_2) z_2^2 + (m_1+m_2) g l_1 \sin\varphi_1)\cos(\varphi_1-\varphi_2)}{l_2(m_1+m_2\sin^2(\varphi_1-\varphi_2))}.
\end{aligned} \tag{8.42}$$

Запись системы ДУ в виде (8.42) позволяет создать m-файл `Pendulum.m`, содержащий описание вектора-функции, стоящей в правой части (8.42), листинг которого приведен ниже.

```
% листинг файла Pendulum.m
function dy=Pendulum(t,z);
% функция, возвращающая
% значение вектора-функции,
% стоящей в правой части (8.42)
% задание глобальных переменных
global m1 m2 L1 L2 g

dy=zeros(4,1);
D=m1+m2*sin(z(1)-z(3))^2;
dy(1)=z(2);
dy(2)=-(m2*L2*sin(z(1)-z(3))*z(4)^2+...
        (m1+m2)*g*sin(z(1)))-...
        (m2*L1*sin(z(1)-z(3))*z(2)^2-...
         m2*g*sin(z(3)))*cos(z(1)-z(3));
 dy(2)=dy(2)/(L1*D);
```

```
 dy(3)=z(4);
 dy(4)=(L1*sin(z(1)-z(3))*z(2)^2-...
        g*sin(z(3)))*(m1+m2)+...
        (m2*L2*sin(z(1)-z(3))*z(4)^2+...
        (m1+m2)*g*L1*sin(z(1)))*cos(z(1)-z(3));
  dy(4)=dy(4)/(L2*D);
```

Далее необходимо выполнить следующую последовательность команд, сохраненную нами в файле `Glava8_3.m`:

```
% листинг файла Glava8_3.m
global m1 m2 L1 L2 g
% задание параметров маятника
m1=100;
m2=1;
L1=1;
L2=2;
g=9.8;

[t Z]=ode45('Pendulum',...
            [0:10^-4:10],...
            [pi/4 0 2*pi/3 0]); % решение системы (8.42)

plot(t,Z(:,1),t,Z(:,3)) % визуализация зависимости
                        % угла отклонения
                        % первого и второго маятников
                        % от времени

% вычисление полной энергии двойного маятника
Ek=(m1+m2)/2*L1^2*Z(:,2).^2+...
    m2/2*L2^2*Z(:,4).^2+...
    m2*L1*L2*cos(Z(:,1)-Z(:,3)).*Z(:,2).*Z(:,4);
Ep=-(m1+m2)*g*L1*cos(Z(:,1))-m2*g*L2*cos(Z(:,3));
E=Ek+Ep;
figure;
plot(t,E); % визуализация зависимости
           % полной энергии системы от времени

% вычисление декартовых координат массы m1
x1=L1*sin(Z(:,1));
y1=L1*cos(Z(:,1));

% визуализация траектории движения массы m1
figure;
plot(x1,y1);
h = gca;
```

```
set(h,'YDir','reverse'); % инвертирование направления оси OY
axis([-(L1+L2)*1.1...
     (L1+L2)*1.1...
     -(L1+L2)*1.1...
     (L1+L2)*1.1]);

% вычисление декартовых координат массы m2
x2=x1+L2*sin(Z(:,3));
y2=y1+L2*cos(Z(:,3));

% визуализация траектории движения массы m2
figure;
plot(x2,y2);
h = gca;
set(h,'YDir','reverse'); % инвертирование
                         % направления оси OY

% задание координат точки подвеса
Sa(1,1)=0;
Sa(1,2)=0;
% задание начальных координат массы m1
Sa(2,1)=x1(1);
Sa(2,2)=y1(1);
% задание начальных координат массы m2
Sa(3,1)=x2(1);
Sa(3,2)=y2(1);

% визуализация начального положения маятника
figure;
plot(Sa(1:2,1),Sa(1:2,2),...
         Sa(2,1),Sa(2,2),'o',...
         Sa(2:3,1),Sa(2:3,2),...
         Sa(3,1),Sa(3,2),'o');
axis([-(L1+L2)*1.1...
      (L1+L2)*1.1...
       -(L1+L2)*1.1...
      (L1+L2)*1.1]);
h = gca;
set(h,'YDir','reverse'); % инвертирование направления оси OY

set(h,'nextplot','replacechildren'); % включение режима
                                     % перерисовки графиков
                                     % в одном и том же окне
k=1;
N=length(t);
```

```
Na=1000; % шаг выбора решений системы ДУ (8.42)

% создание анимационного клипа
figure;
for i=1:Na:N
        Sa(2,1)=x1(i);
        Sa(2,2)=y1(i);
        Sa(3,1)=x2(i);
        Sa(3,2)=y2(i);
        plot(Sa(1:2,1),Sa(1:2,2),...
                Sa(2,1),Sa(2,2),'o',...
                Sa(2:3,1),Sa(2:3,2),...
                Sa(3,1),Sa(3,2),'o');
        axis([-(L1+L2)*1.1...
             (L1+L2)*1.1...
              -(L1+L2)*1.1...
              (L1+L2)*1.1]);
        set(h,'YDir','reverse'); % инвертирование
                                 % направления оси OY
        F(k)=getframe;
        k=k+1;
end;
movie(F,1) % воспроизведение анимационного клипа
```

Результаты выполнения указанной выше последовательности команд представлены на рис. 8.10–8.18.

Рис. 8.10. Зависимость координат масс m_1, m_2 от времени

Рис. 8.11. Зависимость полной энергии системы от времени

Проверить правильность (провести верификацию) компьютерной реализации описанной выше математической модели двойного маятника можно, рассмотрев решения системы дифференциальных уравнений (8.42), например, для следующих случаев:

Рис. 8.12. Траектория движения массы m_1

Рис. 8.13. Траектория движения массы m_2

Рис. 8.14. Начальное положение двойного маятника

Рис. 8.15. Фазовая траектория массы m_1

Рис. 8.16. Фазовая траектория массы m_2

1. $m_1 \gg m_2$, $\varphi_1(0) \neq 0$, $\dot{\varphi}_1(0) = 0$, $\varphi_2(0) = 0$, $\dot{\varphi}_2(0) = 0$. (В этом случае, как очевидно, движение массы m_1 при небольших углах отклонения будет близко к гармоническим колебаниям с периодом $T_1 = 2\pi\sqrt{l_1/g}$.)

2. $m_1 \gg m_2$, $\varphi_1(0) = 0$, $\dot{\varphi}_1(0) = 0$, $\varphi_2(0) \neq 0$, $\dot{\varphi}_2(0) = 0$. (В этом случае, как очевидно, движение массы m_2 при небольших углах отклонения будет близко к гармоническим колебаниям с периодом $T_2 = 2\pi\sqrt{l_2/g}$, а масса будет оставаться практически неподвижной.)

3. $m_1 \gg m_2$, $\varphi_1(0) \neq 0$, $\dot{\varphi}_1(0) = 0$, $\varphi_2(0) \neq 0$, $\dot{\varphi}_2(0) = 0$.

Рассмотрим последний случай более подробно. При малых углах отклонения функция Лагранжа (8.33) принимает следующий вид:

$$L = \frac{m_1 + m_2}{2} l_1^2 \dot{\varphi}_1^2 + \frac{m_2}{2} l_2^2 \dot{\varphi}_2^2 + m_2 l_1 l_2 \dot{\varphi}_1 \dot{\varphi}_2 - \frac{m_1 + m_2}{2} g l_1 \dot{\varphi}_1^2 - \frac{m_2}{2} g l_2 \dot{\varphi}_2^2. \quad (8.43)$$

Далее, подставив (8.43) в уравнения Лагранжа

$$\frac{d}{dt}\frac{\partial L}{\partial \dot{q}_i} = \frac{\partial L}{\partial q_i}, \quad i = 1, 2,$$

получим

$$\begin{aligned}
\frac{\partial L}{\partial \dot{\varphi}_1} &= (m_1 + m_2)\, l_1^2 \dot{\varphi}_1 + m_2 l_1 l_2 \dot{\varphi}_2, \\
\frac{\partial L}{\partial \dot{\varphi}_2} &= m_2 l_2^2 \dot{\varphi}_2 + m_2 l_1 l_2 \dot{\varphi}_1, \\
\frac{d}{dt}\frac{\partial L}{\partial \dot{\varphi}_1} &= (m_1 + m_2)\, l_1^2 \ddot{\varphi}_1 + m_2 l_1 l_2 \ddot{\varphi}_2, && (8.44) \\
\frac{d}{dt}\frac{\partial L}{\partial \dot{\varphi}_2} &= m_2 l_2^2 \ddot{\varphi}_2 + m_2 l_1 l_2 \ddot{\varphi}_1, && (8.45) \\
\frac{\partial L}{\partial \varphi_1} &= -(m_1 + m_2)\, g l_1 \varphi_1, && (8.46) \\
\frac{\partial L}{\partial \varphi_2} &= -m_2 g l_2 \varphi_2. && (8.47)
\end{aligned}$$

Приравнивая, соответственно, выражения (8.44), (8.46) и (8.45), (8.47), получим следующую систему уравнений движения:

$$\begin{aligned}
&(m_1 + m_2) l_1 \ddot{\varphi}_1 + m_2 l_2 \ddot{\varphi}_2 + (m_1 + m_2) g \varphi_1 = 0, \\
&l_1 \ddot{\varphi}_1 + l_2 \ddot{\varphi}_2 + g \varphi_2 = 0.
\end{aligned} \quad (8.48)$$

По общим правилам решения подобных систем уравнений ищем их решение в виде

$$x_k = A_k e^{i\omega t}, \quad k = 1, 2, \quad (8.49)$$

где i — комплексная единица, A_k — некоторые неопределенные пока постоянные.

Подставляя (8.49) в (8.49), получим:

$$\begin{aligned} &A_1\left(m_1+m_2\right)\left(g-l_1\omega^2\right)-A_2m_2l_2\omega^2=0,\\ &-A_1l_1\omega^2+A_2\left(g-l_2\omega^2\right)=0. \end{aligned} \tag{8.50}$$

Для того чтобы система (8.50) имела нетривиальное решение, ее определитель должен равняться нулю:

$$\begin{vmatrix} \left(m_1+m_2\right)\left(g-l_1\omega^2\right) & -m_2l_2\omega^2 \\ -l_1\omega^2 & \left(g-l_2\omega^2\right) \end{vmatrix}=0. \tag{8.51}$$

Из (8.51) получаем характеристическое уравнение

$$\left(m_1+m_2\right)\left(g-l_1\omega^2\right)\left(g-l_2\omega^2\right)-m_2l_2l_1\omega^4=0, \tag{8.52}$$

решив которое относительно ω^2, найдем его корни:

$$\omega_{1,2}^2=\frac{g}{2m_1l_1l_2}\left[(m_1+m_2)(l_1+l_2)\pm\sqrt{(m_1+m_2)[(m_1+m_2)(l_1+l_2)-4m_1l_1l_2]}\right]. \tag{8.53}$$

Корни характеристического уравнения $\omega_{1,2}^2$ определяют частоты нормальных колебаний рассматриваемой системы. При $m_1\to\infty$, как видно из (8.53), частоты нормальных колебаний стремятся к пределам $\sqrt{g/l_1}$ и $\sqrt{g/l_2}$, соответствующим независимым колебаниям двух маятников.

В этом случае, как очевидно, движение массы m_1 при небольших углах отклонения будет близко к гармоническим колебаниям с периодом $T_2=\sqrt{(l_2/g)}$, а масса m_2 будет оставаться практически неподвижной.

***Задача* 8.11**

Проведите верификацию описанной выше реализации компьютерной модели двойного маятника, рассмотрев каждый из трех описанных выше случаев.

Одной из особенностей рассматриваемой системы является возникновение хаотических колебаний, возникающих при увеличении полной энергии системы. Для выявления перехода рассматриваемой системы в хаотический режим можно использовать следующие количественные характеристики:

1. Зависимость расстояния между траекториями, близкими при $t=0$, от времени.
2. Спектры зависимостей $\varphi_1=\varphi(t)$, $\varphi_2=\varphi(t)$.
3. Фазовые траектории.

***Задача* 8.12**

Проведите вычислительные эксперименты и определите значение полной энергии системы, при которой двойной маятник совершает хаотические колебания.

Сравните перечисленные выше количественные характеристики в нехаотическом и хаотических режимах колебаний.

8.6. МАЯТНИКИ С ОСЦИЛЛИРУЮЩЕЙ ТОЧКОЙ ПОДВЕСА

В настоящем разделе рассматриваются математические модели математических маятников, у которых точки подвеса совершают колебательные движения, включая:

1. Маятник с точкой подвеса, совершающей колебания вдоль горизонтальной оси.
2. Маятник с точкой подвеса, совершающей колебания вдоль вертикальной оси.
3. Маятник с точкой подвеса, движущейся по с постоянной скоростью по окружности.

8.6.1. КОМПЬЮТЕРНОЕ МОДЕЛИРОВАНИЕ МАТЕМАТИЧЕСКОГО МАЯТНИКА С ТОЧКОЙ ПОДВЕСА, СОВЕРШАЮЩЕЙ ГОРИЗОНТАЛЬНЫЕ ГАРМОНИЧЕСКИЕ КОЛЕБАНИЯ

Математический маятник, точка подвеса которого совершает горизонтальные колебания, представлен на рис. 8.17.

Построим математическую модель рассматриваемой системы. Из рис. 8.17 видно, что связь декартовых координат маятника (x, y) с углом его отклонения от вертикальной оси и смещением точки подвеса описывается следующими выражениями:

$$x = a\cos\gamma t + l\sin\varphi, \tag{8.54}$$

$$y = l\cos\varphi, \tag{8.55}$$

где l — длина маятника, φ — частота колебаний точки подвеса.

Рис. 8.17. Математический маятник с точкой подвеса, совершающей колебания вдоль горизонтальной оси

Дифференцируя (8.54), (8.55) по времени, получим:

$$\dot{x} = -a\gamma\sin\gamma t + l\cos\varphi\,\dot{\varphi}, \tag{8.56}$$

$$\dot{y} = -l\sin\varphi\,\dot{\varphi}. \tag{8.57}$$

Далее, подставляя (8.56), (8.57) в выражение для кинетической энергии, получим:

$$T = \frac{m}{2}\left(\dot{x}^2 + \dot{y}^2\right) = \frac{m}{2}l^2\dot{\varphi}^2 - mla\gamma\sin\gamma t\cos\varphi\,\dot{\varphi} + m(a\gamma\sin\gamma t)^2. \tag{8.58}$$

Потенциальная энергия маятника равна

$$U = -mgl\cos\varphi. \tag{8.59}$$

Следовательно, функция Лагранжа рассматриваемой динамической системы записывается в виде

$$L = T - U = \frac{m}{2} l^2 \dot{\varphi}^2 - mla\gamma \sin\gamma t \cos\varphi\, \dot{\varphi} + (a\gamma \sin\gamma t)^2 + mg\cos\varphi. \tag{8.60}$$

Выделим в (8.60) полную производную по времени. Для этого заметим, что

$$\frac{d}{dt}(\sin\gamma t \sin\varphi) = \gamma \cos\gamma t \sin\varphi + \sin\gamma t \cos\varphi\, \dot{\varphi}. \tag{8.61}$$

Из (8.61) находим

$$\sin\gamma t \cos\varphi\, \dot{\varphi} = -\gamma \cos\gamma t \sin\varphi + \frac{d}{dt}(\sin\gamma t \sin\varphi). \tag{8.62}$$

Подставляя (8.62) в (8.60), получаем

$$\begin{gathered} L = \frac{m}{2} l^2 \dot{\varphi}^2 - mla\gamma \left(-\gamma \cos\gamma t \sin\varphi + \frac{d}{dt}(\sin\gamma t \sin\varphi) \right) + \\ + m(a\gamma \sin\gamma t)^2 + mg\cos\varphi. \end{gathered} \tag{8.63}$$

И наконец, опуская в (8.63) член $(a\gamma \sin\gamma t)^2$, зависящий только от времени, и полную производную по времени, получаем искомое выражение для функции $\frac{d}{dt}(\sin\gamma t \sin\varphi)$ Лагранжа:

$$L = \frac{m}{2} l^2 \dot{\varphi}^2 + mla\gamma^2 \cos\gamma t \sin\varphi + mgl\cos\varphi. \tag{8.64}$$

Далее, подставив (8.64) в уравнения Лагранжа

$$\frac{d}{dt}\frac{\partial L}{\partial \dot{q}} = \frac{\partial L}{\partial q},$$

получим

$$\begin{gathered} \frac{\partial L}{\partial \dot{\varphi}} = ml^2 \dot{\varphi}, \\ \frac{d}{dt}\frac{\partial L}{\partial \dot{\varphi}} = ml^2 \ddot{\varphi}, \end{gathered} \tag{8.65}$$

$$\frac{\partial L}{\partial \varphi} = mla\gamma^2 \cos\varphi \cos\gamma t - mgl\sin\varphi. \tag{8.66}$$

Приравнивая выражения (8.65) и (8.66), получаем уравнение движения маятника

$$ml^2 \ddot{\varphi} = mla\gamma^2 \cos\varphi \cos\gamma t - mgl\sin\varphi, \tag{8.67}$$

Для нахождения численного решения запишем (8.67) в виде эквивалентной системы дифференциальных уравнений первого порядка:

$$\begin{gathered} \dot{\varphi} = z_1, \\ \dot{z}_1 = \frac{a}{l}\gamma^2 \cos\varphi \cos\gamma t - \frac{g}{l}\sin\varphi \end{gathered} \tag{8.68}$$

Используя запись ДУ (8.67) в виде системы (8.68), целесообразно создать файл `Pendilum2.m`, содержащий функцию, стоящую в правой части (8.68), листинг которого приведен ниже.

```
% листинг файла Pendilum2.m
function dy=Pendulum2(t,z);
% функция, возвращающая значения
% вектора-функции,
% стоящей в правой части (8.68)

global a L Gg g

dy=zeros(2,1);
dy(1)=z(2);
dy(2)=a/L*Gg^2*cos(Gg*t)*cos(z(1))-g/L*sin(z(1));
```

Далее необходимо выполнить следующую последовательность команд, сохраненную нами в файле `Glava8_4.m`:

```
% листинг файла Glava8_4.m

global a L Gg g

 % задание параметров маятника
L=0.5;
a=0.1;
Gg=5; % частота маятника
g=9.8;

% решение системы ДУ (8.68)
[t Z]=ode45('Pendulum2',...
             [0:10^-4:10],...
             [0.1*pi 0]);

% вычисление декартовых координат маятника
x=a*cos(Gg*t)+L*sin(Z(:,1));
y=L*cos(Z(:,1));

% визуализация траектории маятника
plot(x,y);
axis([-(a+L) (a+L) -(a+L) (a+L)]);
set(gca,'YDir','reverse');
```

Результаты выполнения описанной выше последовательности команд для различных параметров маятника представлены на рис. 8.18, 8.19.

Сравнивая рис. 8.18 и 8.19, видим, что в первом случае колебания маятника происходят симметрично относительно точки $\varphi_0 = 0$, во втором — относительно точки

Рис. 8.18. Траектория движения маятника с колеблющейся точкой подвеса в горизонтальной плоскости ($l = 0,5; a = 0,1; g = 5$)

Рис. 8.19. Траектория движения маятника с колеблющейся точкой подвеса в горизонтальной плоскости ($l = 0,5; a = 0,1; g = 50$)

$\varphi \approx 70°$. Таким образом, положение точки равновесия маятника оказывается зависящим от частоты колебаний подвеса.

Для объяснения данной зависимости проведем анализ уравнения движения (8.67), из которого видно, что в его правой части присутствуют два члена, первый:

$$f(\varphi, t) = -mla\gamma^2 \cos\varphi \cos\gamma t, \tag{8.69}$$

второй:

$$F = -\frac{dU}{d\varphi} = -mgl \sin\varphi. \tag{8.70}$$

Из характера траекторий движения маятника, представленных на рис. 8.18, 8.19, ясно, что в аналитическом виде искать решение уравнения (8.67) следует в виде:

$$\varphi(t) = \Phi(t) + \xi(t), \tag{8.71}$$

где $\Phi(t)$ — функция, описывающая медленные перемещения маятника, $\xi(t)$ — функция, описывающая малые осцилляции. При этом среднее значение функции вычисляемое на временном интервале $2\pi/\gamma$, а значение функции $\overline{\xi(t)} = 0$, напротив, меняется мало.

Подставляя (8.71) в (8.67) и разлагая по степеням $\xi(t)$ с точностью до членов первого порядка, получим

$$ml^2\ddot{\Phi} + ml^2\ddot{\xi} = -\frac{dU}{dx} + f(\varphi, t) = -\frac{dU}{d\Phi} - \xi(t)\frac{d^2U}{d\Phi^2} + f(\Phi, t) + \xi(t)\frac{\partial f}{\partial \Phi}. \tag{8.72}$$

В (8.72) присутствуют члены, имеющие различную быстроту изменения во времени: «медленные» и «плавные». Они должны взаимно сокращаться в каждой из этих групп в отдельности. Для осциллирующих членов достаточно написать:

$$ml^2\ddot{\xi} = f(\Phi, t). \tag{8.73}$$

Остальные «быстрые» члены уравнения (8.72) содержат множитель $\xi(t)$ и потому малы, по сравнению с оставленным членом. Интегрируя (8.73) с функцией $f(\Phi, t)$, описываемой выражением (8.69) (при этом величина $\Phi(t)$ рассматривается как постоянная), получим

$$\xi\,(t) = -\frac{f}{ml^2\gamma^2}. \tag{8.74}$$

Далее проведем усреднение (8.72) по времени, приняв во внимание, что $\bar{f} = 0$, $\bar{\xi}(t) = 0$:

$$ml^2\ddot{\Phi} = -\frac{dU}{d\Phi} + \overline{\xi\,(t)\,\frac{\partial f}{\partial \Phi}} = -\frac{dU}{d\Phi} - \frac{1}{ml^2\gamma^2}\overline{f\frac{\partial f}{\partial \Phi}}, \tag{8.75}$$

содержащее только функцию $\Phi(t)$.

Окончательно запишем (8.75) в виде

$$ml^2\ddot{\Phi} = -\frac{dU_{eff}}{d\Phi}, \tag{8.76}$$

где U_{eff} — «эффективная потенциальная энергия»:

$$U_{eff} = U + \frac{1}{2ml^2\gamma^2}\overline{f^2}. \tag{8.77}$$

Сравнивая (8.77) и (8.74), видим, что дополнительный (по отношению к полю U) член представляет собой не что иное, как среднюю кинетическую энергию колебательного движения:

$$U_{eff} = U + \frac{ml^2}{2}\overline{\dot{\xi}^2}. \tag{8.78}$$

Таким образом, усредненное по осцилляциям движение частицы происходит так, как будто на него помимо постоянного во времени поля U действует дополнительное поле, квадратично зависящее от амплитуды переменного поля.

Принимая во внимание, что $\overline{\cos^2\gamma t} = 1/2$, из (8.69), получим

$$\overline{f^2} = \frac{1}{2}\left(mal\gamma^2\right)^2\cos^2\varphi. \tag{8.79}$$

Подставляя (8.79) в (8.77), окончательно получаем

$$U_{eff} = mgl\left[-\cos\varphi + \frac{a^2\gamma^2}{4gl}\cos^2\varphi\right]. \tag{8.80}$$

Положение точек устойчивого равновесия находим из условия:

$$\frac{dU_{eff}}{d\varphi} = -mgl\left[1 + \frac{a^2\gamma^2}{2gl}\cos\varphi\right]\sin\varphi = 0. \tag{8.81}$$

Из (8.81) видно, что при $a^2\gamma^2/2gl < 1$ устойчивое положение маятника — $\varphi = 0$, при $a^2\gamma^2/2gl > 1$ устойчивое положение маятника — $\cos\varphi = 2gl/a^2l^2$. Данный результат подтверждают графики функций $U_{eff} = U_{eff}(\varphi)$, построенные для рассмотренных выше наборов параметров маятника (рис. 8.20, 8.21).

Рис. 8.20. Зависимость $U_{eff} = U_{eff}(\varphi)$ $(a = 0,1; l = 0,5; \gamma = 5)$

Рис. 8.21. Зависимость $U_{eff} = U_{eff}(\varphi)$ $(a = 0,1; l = 0,5; \gamma = 50)$

***Задача* 8.13**

1. *Оцените, используя численные решение уравнения (8.68), координаты углов, соответствующие положениям равновесия.*

2. *Сравните оценки, полученные с помощью численного решения, с оценками, полученными в соответствие с приведенными выше формулами.*

***Задача* 8.14**

Вычислите и постройте зависимость мгновенных значений энергии маятника с точкой подвеса, совершающей горизонтальные колебания. Объясните полученный результат.

8.6.2. МАТЕМАТИЧЕСКАЯ МОДЕЛЬ МАЯТНИКА С ТОЧКОЙ ПОДВЕСА, СОВЕРШАЮЩЕЙ ВЕРТИКАЛЬНЫЕ ГАРМОНИЧЕСКИЕ КОЛЕБАНИЯ

Построим математическую модель маятника, точка подвеса которого совершает горизонтальные колебания (рис. 8.22).

Из рис. 8.22 видно, что связь декартовых координат маятника (x, y) с углом его отклонения от вертикальной оси и смещением точки подвеса описывается следующими выражениями:

$$x = l \sin \varphi, \tag{8.82}$$

$$y = a \cos \gamma t + l \cos \varphi, \tag{8.83}$$

где l — длина маятника, γ — частота колебаний точки подвеса.

Дифференцируя (8.82), (8.83) по времени, получим

$$\dot{x} = l \cos \varphi \, \dot{\varphi}, \tag{8.84}$$

$$\dot{y} = -a\gamma \sin \gamma t - l \sin \varphi \, \dot{\varphi}. \tag{8.85}$$

Рис. 8.22. Маятник с точкой подвеса, совершающей колебания вдоль вертикальной оси

Далее, подставляя (8.84), (8.85) в выражение для кинетической энергии, получим:

$$T = \frac{m}{2}\left(\dot{x}^2 + \dot{y}^2\right) = \frac{m}{2} l^2 \dot{\varphi}^2 + 2mla\gamma \sin\gamma t \sin\varphi\, \dot{\varphi} + (a\gamma \sin\gamma\, t)^2. \tag{8.86}$$

Потенциальная энергия маятника равна

$$U = -mgl\cos\varphi + mga\cos\gamma t. \tag{8.87}$$

Следовательно, функция Лагранжа

$$L = T - U$$

рассматриваемой динамической системы записывается в виде

$$L = \frac{m}{2} l^2 \dot{\varphi}^2 - mla\gamma \sin\gamma t \sin\varphi\, \dot{\varphi} + m(a\gamma \sin\gamma\, t)^2 + mgl\cos\varphi + mga\cos\gamma t. \tag{8.88}$$

Выделим в (8.88) полную производную по времени. Для этого заметим, что

$$\frac{d}{dt}\left(\sin\gamma t \cos\varphi\right) = \gamma\cos\gamma t\cos\varphi - \sin\gamma t \sin\varphi\, \dot{\varphi}. \tag{8.89}$$

Из (8.89) находим

$$\sin\gamma t \sin\varphi\, \dot{\varphi} = -\gamma\cos\gamma t\cos\varphi + \frac{d}{dt}\left(\sin\gamma t\cos\varphi\right). \tag{8.90}$$

Подставляя (8.90) в (8.88) и опуская полную производную по времени:

$$\frac{d}{dt}\left(\sin\gamma t\cos\varphi\right),$$

а также члены, зависящие только от времени:

$$mga\cos\gamma\, t, m(a\gamma\sin\gamma\, t)^2,$$

получаем

$$L = \frac{m}{2} l^2 \dot{\varphi}^2 + mla\gamma^2 \cos\gamma t \cos\varphi + mgl\cos\varphi. \tag{8.91}$$

Далее, подставив (8.91) в уравнения Лагранжа

$$\frac{d}{dt}\frac{\partial L}{\partial \dot{q}} = \frac{\partial L}{\partial q},$$

получим

$$\frac{\partial L}{\partial \dot{\varphi}} = ml^2\dot{\varphi},$$

$$\frac{d}{dt}\frac{\partial L}{\partial \dot{\varphi}} = ml^2\ddot{\varphi}, \tag{8.92}$$

$$\frac{\partial L}{\partial \varphi} = -mla\gamma^2 \sin\varphi \cos\gamma t - mgl\sin\varphi. \tag{8.93}$$

Приравнивая выражения (8.92) и (8.93), получаем окончательное уравнение движения маятника

$$ml^2\ddot{\varphi} = -mla\gamma^2 \sin\varphi \cos\gamma t - mgl\sin\varphi. \tag{8.94}$$

Для нахождения численного решения запишем (8.94) в виде эквивалентной системы дифференциальных уравнений первого порядка:

$$\begin{gathered} \dot{\varphi} = z_1, \\ \dot{z}_1 = \frac{a}{l}\gamma^2 \sin\varphi\ \cos\gamma t - \frac{g}{l}\sin\varphi. \end{gathered} \tag{8.95}$$

Запись ДУ (8.94) в виде системы (8.95) позволяет создать файл, содержащий функцию, стоящую в правой части (8.95) (файл `Pendilum3.m`). Ниже приведен листинг данного файла.

```
% листинг файла Pendilum3.m
function dy=Pendulum3(t,z);
% функция, возвращающая значения
$ вектора-функции,
% стоящей в правой части (8.95)
global a L Gg g

dy=zeros(2,1);
dy(1)=z(2);
dy(2)=a/L*Gg^2*cos(Gg*t)*sin(z(1))-...
      g/L*sin(z(1));
```

Далее необходимо выполнить следующую последовательность команд, сохраненную нами в файле `Glava8_5.m`:

```
% листинг файла Glava8_5.m
global a L Gg g

% задание параметров маятника
L=0.5;
a=0.1;
Gg=5; % частота маятника
g=9.8;

% решение системы ДУ (8.95)
[t Z]=ode45('Pendulum3',...
            [0:10^-4:10],...
            [0.1*pi 0]);

% вычисление декартовых координат маятника
x= L*sin(Z(:,1));
y=a*cos(Gg*t)+L*cos(Z(:,1));

% визуализация траектории маятника
plot(x,y);
axis([-(a+L)...
     (a+L)...
     -(a+L)...
     (a+L)]);
set(gca,'YDir','reverse'); % инвертирование оси OY
```

Результаты выполнения описанной выше последовательности команд для различных параметров маятника представлены на рис. 8.23–8.25.

Рис. 8.23. Траектория движения маятника с колеблющейся точкой подвеса в горизонтальной плоскости ($l = 0,5; a = 0,1; \varphi(0) = 0,1\pi; \gamma = 50$)

Рис. 8.24. Конфигурация системы в момент времени $t = 10$

Рис. 8.25. Траектория движения маятника с колеблющейся точкой подвеса в горизонтальной плоскости ($l = 0,5; a = 0,1; \varphi(0) = 0,9\pi; \gamma = 50$)

Из рисунков 8.23–8.25 видно, что при частоте колебаний $\gamma = 5$ маятник колеблется вокруг единственной неподвижной точки $\varphi = 0$, при $\gamma = 50$ у рассматриваемого маятника существуют две точки равновесия $\varphi = 0$, $\varphi = \pi$.

Для объяснения причины появления второй точки равновесия, как и в предыдущем случае, следует провести анализ зависимости эффективной энергии от угла. Из (8.94) видно, что в рассматриваемом случае дополнительная сила, обусловленная вертикальными колебаниями точки подвеса, описывается выражением:

$$f = -mla\gamma^2 \sin\varphi \cos\gamma t. \qquad (8.96)$$

Подставляя (8.96) в (8.77), получаем:

$$U = mgl\left[-\cos\varphi + \frac{a^2\gamma^2}{4gl}\sin^2\varphi\right]. \qquad (8.97)$$

Положение точек устойчивого равновесия находим из условия

$$\frac{dU_{eff}}{d\varphi} = mgl\left[1 + \frac{a^2\gamma^2}{2gl}\cos\varphi\right]\sin\varphi = 0. \qquad (8.98)$$

При $a^2\gamma^2/2gl < 1$ уравнение (8.98) формально имеет два корня $\varphi_1 = 0$, $\varphi_2 = 0$. Точка $\varphi_1 = 0$ всегда является минимумом функции, а точка $\varphi_2 = 0$ оказывается точкой максимума, в чем легко убедиться, вычислив знаки производной слева и справа от данной точки. При $a^2\gamma^2/2gl > 1$ точка $\varphi_2 = \pi$, как легко убедиться, оказывается точкой минимума и, следовательно, в зависимости от начальных условий будет колебаться возле одной из найденных точек $\varphi_1 = 0$, $\varphi_2 = \pi$. Формально при $a^2\gamma^2/2gl > 1$ уравнение (8.98) имеет еще один корень, являющийся решением уравнения

$$1 + \frac{a^2\gamma^2}{2gl}\cos\varphi = 0. \qquad (8.99)$$

Однако корень данного уравнения соответствует точке перегиба, но не экстремуму функции. Справедливость данного вывода иллюстрируют графики, представленные рис. 8.26, 8.27.

***Задача* 8.15**

Проведите аналитические выкладки и убедитесь в том, что корни уравнения (8.99) являются точками перегиба функции $U_{eff} = U_{eff}(\varphi)$.

***Задача* 8.16**

Вычислите и постройте зависимость мгновенных значений энергии маятника с точкой подвеса, совершающей вертикальные колебания. Объясните полученный результат.

Рис. 8.26. Траектория движения маятника с колеблющейся точкой подвеса в горизонтальной плоскости ($l = 0,5; a = 0,1; \varphi(0) = 0,1\pi; \gamma = 50$)

Рис. 8.27. Конфигурация системы в момент времени $t = 10$

8.6.3. МАТЕМАТИЧЕСКАЯ МОДЕЛЬ МАЯТНИКА С ТОЧКОЙ ПОДВЕСА, СОВЕРШАЮЩЕЙ ДВИЖЕНИЕ ПО ОКРУЖНОСТИ

Построим математическую модель плоского маятника, точка подвеса которого движется по вертикальной окружности с постоянной частотой γ (рис. 8.28).

Из рис. 8.28 видно, что в выбранной системе координат координаты точечной массы m описываются следующими выражениями:

$$x = a\cos\gamma t + l\sin\varphi, \qquad (8.100)$$

$$y = -a\sin\gamma t + l\cos\varphi. \qquad (8.101)$$

Продифференцировав (8.100) и (8.101) и подставив в известное выражение для кинетической энергии

$$T = \frac{m_2}{2}\left(\dot{x}^2 + \dot{y}^2\right),$$

получим

Рис. 8.28. Маятник с точкой подвеса, движущейся по окружности с равномерной скоростью

$$\begin{aligned} T &= \frac{m}{2}(-a\gamma\sin\gamma t + l\dot{\varphi}\cos\varphi)^2 + \frac{m}{2}(-a\gamma\cos\gamma t - l\dot{\varphi}\sin\varphi)^2 = \\ &= \frac{m}{2}\left(a^2\gamma^2 + l^2\dot{\varphi}^2 + 2a\gamma l\dot{\varphi}\sin\varphi\cos\gamma t - 2a\gamma l\dot{\varphi}\cos\varphi\sin\gamma t\right) = \\ &= \frac{m}{2}\left(a^2\gamma^2 + l^2\dot{\varphi}^2 + 2a\gamma l\dot{\varphi}\ \sin(\varphi - \gamma t)\right). \end{aligned} \qquad (8.102)$$

Выделим в (8.102) полную производную от времени, заметив для этого, что

$$\frac{d}{dt}\cos(\varphi-\gamma t) = -\left(\dot{\varphi}\sin(\varphi-\gamma t) - \gamma\sin(\varphi-\gamma t)\right),$$

откуда

$$\dot{\varphi}\sin(\varphi-\gamma t) = -\left(\frac{d}{dt}\cos(\varphi-\gamma t) - \gamma\sin(\varphi-\gamma t)\right). \tag{8.103}$$

Подставив (8.103) в (8.102), получим

$$T = \frac{m}{2}\left(a^2\gamma^2 + l^2\dot{\varphi}^2 + 2al\gamma^2\sin(\varphi-\gamma t) - \frac{d}{dt}\cos(\varphi-\gamma t)\right). \tag{8.104}$$

Потенциальная энергия маятника равняется

$$U = -mg\left(-a\sin\gamma t + l\cos\varphi\right). \tag{8.105}$$

Учитывая соотношения (8.104), (8.105), получаем следующее выражение для функции Лагранжа:

$$\begin{aligned} L =& \frac{m}{2}a^2\gamma^2 + \frac{m}{2}\dot{\varphi}^2 + mla\gamma^2\sin(\varphi-\gamma t) - \\ &- mla\gamma\frac{d}{dt}\cos(\varphi-\gamma t) + mg\left(-a\sin\gamma t + l\cos\varphi\right), \end{aligned}$$

в котором можно опустить постоянный член $\frac{m}{2}a^2\gamma^2$, полную производную по времени от функции $\cos(\varphi-\gamma t)$ и функцию $a\sin\gamma t$, зависящую только от времени:

$$L = \frac{m}{2}l^2\dot{\varphi}^2 + mla\gamma^2\sin(\varphi-\gamma t) + mgl\cos\varphi. \tag{8.106}$$

Далее находим

$$\begin{aligned} \frac{\partial L}{\partial\dot{\varphi}} &= ml^2\dot{\varphi}, \\ \frac{d}{dt}\frac{\partial L}{\partial\dot{\varphi}} &= ml^2\ddot{\varphi}, \end{aligned} \tag{8.107}$$

$$\frac{\partial L}{\partial\varphi} = mla\gamma^2\cos(\varphi-\gamma t) - mgl\sin\varphi. \tag{8.108}$$

Приравняв (8.107) и (8.108), получим искомое уравнение движения:

$$ml^2\ddot{\varphi} = mla\gamma^2\cos(\varphi-\gamma t) - mgl\sin\varphi. \tag{8.109}$$

Для удобства нахождения в дальнейшем численного решения ДУ (8.109) преобразуем ее к эквивалентной системе ДУ первого порядка:

$$\begin{aligned} \dot{\varphi} &= z_1, \\ \dot{z}_1 &= \frac{a}{l}\gamma^2\cos(\varphi-\gamma t) - \frac{g}{l}\sin\varphi\,. \end{aligned} \tag{8.110}$$

Запись системы ДУ в виде (8.110) позволяет создать m-файл `Pendilume4.m`, содержащий описание вектора-функции, стоящей в правой части (8.110), листинг которого приведен ниже.

```
% листинг файла Pendilume4.m
function dy=Pendulum4(t,z);
% функция, возвращающая
% значения вектора-функции,
% стоящей в правой части (8.110)
global a L Gg g

dy=zeros(2,1);
dy(1)=z(2);
dy(2)=a/L*Gg^2*cos(z(1)-Gg*t)-...
      g/L*sin(z(1));
```

Далее необходимо выполнить следующую последовательность команд, сохраненную нами в файле `Glava8_6.m`:

```
% листинг файла Glava8_6.m
global a L Gg g

% задание параметров маятника
L=0.5;
a=0.1;
Gg=50; % частота маятника
g=9.8;

% решение системы ДУ (8.110)
[t Z]=ode45('Pendulum4',...
             [0:10^-4:10],...
             [0.9*pi 0]);

% вычисление декартовых координат маятника
x=a*cos(Gg*t)+...
  L*sin(Z(:,1));
y=-a*sin(Gg*t)+...
  L*cos(Z(:,1));

% визуализация траектории маятника
plot(x,y);
axis([-(L+a)...
      (L+a)...
      -(L+a)...
      (L+a)]);
set(gca,'YDir','reverse');
```

Результаты выполнения описанной выше последовательности команд для различных параметров маятника представлены на рис. 8.29–8.30.

Рис. 8.29. Траектория движения маятника с колеблющейся точкой подвеса в горизонтальной плоскости ($l = 0$; $a = 0,1$; $\varphi(0) = 0,9\pi$; $\gamma = 5$)

Рис. 8.30. Траектория движения маятника с колеблющейся точкой подвеса в горизонтальной плоскости ($l = 0$; $a = 0,1$; $\varphi(0) = 0,9\pi$; $\gamma = 50$)

Из рис. 8.29, 8.30 видно, что при $\gamma = 5$ маятник совершает колебания вокруг точки $\varphi = 0$, при $\gamma = 50$ маятник вращается вокруг точки подвеса. Таким образом, в отличие от рассмотренных ранее случаев при круговом движении новых точек равновесия не возникает.

Для объяснения причины появления второй точки равновесия, как и в предыдущем случае, следует провести анализ зависимости эффективной энергии от угла. Из (8.109) видно, что в рассматриваемом случае дополнительная сила, обусловленная вертикальными колебаниями точки подвеса, описывается выражением

$$f = mla\gamma^2 \cos(\varphi - \gamma t)\,. \tag{8.111}$$

Подставляя (8.111) в (8.77), получаем

$$U_{eff} = mgl\left[-\cos\varphi + \frac{a^2\gamma^2}{4gl}\right]. \tag{8.112}$$

Положение точек устойчивого равновесия находим из условия

$$\frac{dU_{eff}}{d\varphi} = mgl\sin\varphi = 0. \tag{8.113}$$

Из (8.113) видно, что вне зависимости от радиуса окружности и частоты вращения точки подвеса в рассматриваемом случае существует единственная точка равновесия $\varphi = 0$.

***Задача* 8.17**

Вычислите и постройте зависимость мгновенных значений энергии маятника с точкой подвеса, совершающей движение по окружности. Объясните полученный результат.

8.7. МОДЕЛИРОВАНИЕ КОЛЕБАТЕЛЬНОЙ ХИМИЧЕСКОЙ РЕАКЦИИ

Колебательные процессы могут возникать не только в рассмотренных выше механических системах с достаточно простой структурой, но и в значительно более сложных системах химических, биологических, социальных, экономических и др. В данном разделе мы рассматриваем математическую модель одной из наиболее известных химических реакций — реакции Белоусова–Жаботинского, являющейся одним из первых экспериментальных доказательств существования колебательных химических реакций.

Уравнения химических реакций, в которых вещества реагенты A и B превращаются в продукты реакции D и E через промежуточные продукты X и Y, имеют следующий вид [7]:

$$\begin{aligned} &A \to X;\\ &B + X \to Y + D;\\ &2X + Y \to 3X;\\ &X \to E. \end{aligned} \tag{8.114}$$

При построении математической модели химической реакции (8.114), как правило, делают следующие допущения [5,7]:

1) концентрация веществ A и B постоянна во времени и однородна по пространству;

2) продукты D и E либо химически инертны, либо немедленно удаляются из реагирующего объема;

3) реакции являются экзотермическими, что позволяет пренебречь влиянием обратных процессов, можно записать следующую систему уравнений, описывающих изменение во времени концентрации промежуточных продуктов X и Y:

$$\begin{aligned} \frac{\partial X}{\partial t} &= k_a A - k_b BX + k_c X^2 Y - k_d X + D_X \nabla^2 X,\\ \frac{\partial Y}{\partial t} &= k_b BX - k_c X^2 Y + D_Y \nabla^2 Y, \end{aligned} \tag{8.115}$$

где k_a, k_b, k_c, k_d — константы скоростей реакций, $D_{X,Y}$ — коэффициенты диффузии продуктов X и Y.

Для дальнейшего численного решения системы ДУ (8.115), относящуюся к классу систем ДУ параболического типа, оказывается удобным записать уравнения рассматриваемой системы в безразмерном виде. При измерении:

- времени в единицах k_d^{-1};
- длины в единицах l (l — характерный размер реагирующего объема);
- концентрации X и Y в единицах $(k_d/k_c)^{1/2}$;
- A в единицах $\left(k_d^3/k_a^2 k_c\right)^{1/2}$;

- B в единицах $(k_d/k_b)^{1/2}$;
- $D_{X,Y}$ в единицах $k_d l^2$;

система ДУ (8.115) примет следующий вид:

$$\begin{aligned} \frac{\partial X}{\partial t} &= A - (B+1)\,X + X^2 Y + D_X \nabla^2 X, \\ \frac{\partial Y}{\partial t} &= BX - X^2 Y + D_Y \nabla^2 Y. \end{aligned} \tag{8.116}$$

***Задача* 8.18**

Получите самостоятельно систему ДУ (8.116), проведя обезразмеривание системы ДУ (8.115).

Найдем единственное тривиальное решение (8.116), соответствующее однородному стационарному состоянию, в котором все производные равны нулю. Приравняв в (8.116) к нулю все временные и пространственные производные, получаем систему линейных уравнений относительно неизвестных X, Y:

$$\begin{aligned} &A - (B+1)\,X + X^2 Y = 0, \\ &BX - X^2 Y = 0. \end{aligned} \tag{8.117}$$

Решив (8.117), найдем равновесные значения концентраций реагентов

$$X = X_0 = A, \tag{8.118}$$

$$Y = Y_0 = B/A. \tag{8.119}$$

Граничные условия выбираем из условия «отсутствия потоков» на границе, которое выполняется при равенстве на границе поверхности реагирующего объема нулю нормальных производных от X и Y. С физической точки зрения данное условие соответствует протеканию химической реакции в замкнутом сосуде.

Предваряя численное решение системы ДУ (8.116), исследуем поведение системы при малых отклонениях от равновесного значения, положив для этого

$$X = X_0 + \delta X\,(r,t)\,, \tag{8.120}$$

$$Y = Y_0 + \delta Y\,(r,t)\,, \tag{8.121}$$

где r — пространственные переменные, а $|\delta X\,(r,t)| \ll X_0,\ |\delta Y\,(r,t)| \ll Y_0$.

Подставив (8.120), (8.121) в (8.117) и линеаризовав систему относительно малых величин, получим

$$\begin{aligned} \frac{\partial}{\partial t}\delta X &= \left(2X_0Y_0 - B - 1 + D_X\nabla^2\right)\delta X + X_0^2\delta Y, \\ \frac{\partial}{\partial t}\delta Y &= \left(B - 2X_0Y_0\right)\delta X + \left(D_y\nabla^2 - X_0^2\right)\delta Y. \end{aligned} \tag{8.122}$$

Рассмотрим одномерную задачу в отсутствие потоков на границе, решение которой будем искать в виде рядов Фурье:

$$\delta X\left(r,t\right)=\sum_{m=0}^{\infty}\delta X_m e^{i\omega_m t}\cos\left(\pi m r\right), \tag{8.123}$$

$$\delta Y\left(r,t\right)=\sum_{m=0}^{\infty}\delta Y_m e^{i\omega_m t}\cos\left(\pi m r\right), \tag{8.124}$$

где $0\leq r\leq 1$.

Из (8.123), (8.124) видно, что значение ω_m определяет устойчивость m-й нормальной моды колебаний, в частности, если $\operatorname{Im}\omega_m<0$, то мода оказывается неустойчивой (ее амплитуда растет во времени), и наоборот при $\operatorname{Im}\omega_m>0$.

Подставляя (8.123), (8.124) в (8.122), получим:

$$\begin{gathered}\sum_{m=0}^{\infty}i\omega_m\delta X_m e^{i\omega_m t}\cos\left(\pi m r\right)=\left(2X_0Y_0-B-1\right)\sum_{m=0}^{\infty}\delta X_m e^{i\omega_m t}\cos\left(\pi m r\right)-\\ -D_X\sum_{m=0}^{\infty}m^2\pi^2\delta X_m e^{i\omega_m t}\cos\left(\pi m r\right)+\sum_{m=0}^{\infty}\delta Y_m e^{i\omega_m t}\cos\left(\pi m r\right),\end{gathered} \tag{8.125}$$

$$\begin{gathered}\sum_{m=0}^{\infty}i\omega_m\delta Y_m e^{i\omega_m t}\cos\left(\pi m r\right)=\left(B-2X_0Y_0\right)\sum_{m=0}^{\infty}\delta X_m e^{i\omega_m t}\cos\left(\pi m r\right)-\\ -\left(D_Y m^2\pi^2+X_0^2\right)\sum_{m=0}^{\infty}\delta Y_m e^{i\omega_m t}\cos\left(\pi m r\right).\end{gathered} \tag{8.126}$$

Приравняв коэффициенты при каждой пространственной моде, получаем однородную систему линейных уравнений для нахождения собственных частот колебаний:

$$\begin{aligned}i\omega_m\delta X_m&=\left(2X_0Y_0-B-1-D_X m^2\pi^2\right)\delta X_m+X_0^2\delta Y_m,\\ i\omega_m\delta Y_m&=\left(B-2X_0Y_0\right)\delta X_m-\left(D_Y m^2\pi^2+X_0^2\right)\delta Y_m.\end{aligned} \tag{8.127}$$

Система (8.127) является совместной, если ее определитель

$$\begin{vmatrix} i\omega_m-\alpha_m & X_0^2\\ \left(B-2X_0Y_0\right) & i\omega_m+\beta_m\end{vmatrix}=0, \tag{8.128}$$

где

$$\alpha_m=2X_0Y_0-B-1-D_X m^2\pi^2, \tag{8.129}$$

$$\beta_m=D_Y m^2\pi^2+X_0^2. \tag{8.130}$$

Из (8.128) получаем следующее характеристическое уравнение на собственные числа:

$$-\omega_m^2-i\left(\alpha_m-\beta_m\right)\omega_m-\alpha_m\beta_m-X_0^2\left(B-2X_0Y_0\right)=0. \tag{8.131}$$

Подставив выражения (8.118), (8.119) в (8.129), (8.130), (8.131), получаем:

$$\alpha_m=B-1-D_X m^2\pi^2, \tag{8.132}$$

$$\beta_m = D_Y m^2 \pi^2 + A^2, \tag{8.133}$$

$$\omega_m^2 + i\left(\beta_m - \alpha_m\right)\omega_m + \alpha_m \beta_m - A^2 B = 0. \tag{8.134}$$

Корни квадратного уравнения (8.134) равны

$$\begin{aligned}\omega_m^{\pm} &= \frac{1}{2}\left[i\left(\beta_m - \alpha_m\right) \pm \sqrt{-\left(\beta_m - \alpha_m\right)^2 - 4\left(\alpha_m \beta_m + A^2 B\right)}\right] = \\ &= \frac{i}{2}\left[\left(\beta_m - \alpha_m\right) \pm \sqrt{\left(\beta_m + \alpha_m\right)^2 - 4A^2 B}\right].\end{aligned} \tag{8.135}$$

Подставив (8.135) в (8.123), (8.124), получаем

$$\delta X\left(r,t\right) = \sum_{m=0}^{\infty} \delta X_m e^{\frac{1}{2}\left[\left(\alpha_m - \beta_m\right) \mp \sqrt{\left(\beta_m + \alpha_m\right)^2 - 4A^2 B}\right] t} \cos\left(\pi m r\right), \tag{8.136}$$

$$\delta Y\left(r,t\right) = \sum_{m=0}^{\infty} \delta Y_m e^{\frac{1}{2}\left[\left(\alpha_m - \beta_m\right) \mp \sqrt{\left(\beta_m + \alpha_m\right)^2 - 4A^2 B}\right] t} \cos\left(\pi m r\right). \tag{8.137}$$

Из (8.136), (8.137) видно, что изменение решений системы ДУ (8.122) во времени определяется значением выражения

$$q_m^{\pm} = \left(\alpha_m - \beta_m\right) \mp \sqrt{\left(\alpha_m + \beta_m\right)^2 - 4A^2 B}. \tag{8.138}$$

Анализ уравнений (8.132), (8.133), (8.135) позволяет выявить некоторые интересные с точки зрения анализа устойчивости случаи.

1. При $m \to \infty$

$$\alpha_m \approx -D_X m^2 \pi^2,$$
$$\beta_m \approx D_Y m^2 \pi^2,$$

поэтому

$$q_m^{+} \approx -2\beta_m \approx -2D_Y m^2 \pi^2, \tag{8.139}$$

$$q_m^{-} \approx 2\alpha_m \approx -2D_X m^2 \pi^2. \tag{8.140}$$

Подставляя (8.139), (8.140) в выражения (8.136), (8.137), записываем последние в виде

$$\begin{bmatrix} \delta X^{+}\left(r,t\right) \\ \delta Y^{+}\left(r,t\right) \end{bmatrix}^{+} = \sum_{m=0}^{\infty} \begin{bmatrix} \delta X_m\left(r,t\right) \\ \delta Y_m\left(r,t\right) \end{bmatrix} e^{-D_Y m^2 \pi^2 t} \cos\left(\pi m r\right), \tag{8.141}$$

$$\begin{bmatrix} \delta X^{-}\left(r,t\right) \\ \delta Y^{-}\left(r,t\right) \end{bmatrix}^{-} = \sum_{m=0}^{\infty} \begin{bmatrix} \delta X_m\left(r,t\right) \\ \delta Y_m\left(r,t\right) \end{bmatrix} e^{-D_X m^2 \pi^2 t} \cos\left(\pi m r\right). \tag{8.142}$$

Из (8.141), (8.142) видно, что при $t \to \infty$

$$\delta X^{\pm}\left(r,t\right) \to 0,$$
$$\delta Y^{\pm}\left(r,t\right) \to 0,$$

то есть составляющие с большими номерами, что эквивалентно большим значениям волновых чисел (коротким длинам волн), являются устойчивыми.

2. Выражение (8.135) будет иметь комплексные корни при выполнении условия

$$(\alpha_m + \beta_m)^2 - 4A^2B < 0. \tag{8.143}$$

Подставляя (8.132), (8.133) в (8.143), получаем искомое условие в виде

$$\left(A - \sqrt{\Delta_m}\right)^2 < B < \left(A + \sqrt{\Delta_m}\right)^2, \tag{8.144}$$

где $\Delta_m = 1 + m^2\pi^2 (D_X - D_Y)$.

Из (8.144) видно, что должно выполняться неравенство $\Delta_m > 0$, следовательно, колебательные моды будут наблюдаться только при выполнении условия

$$D_X - D_Y < \frac{1}{m^2\pi^2}.$$

3. При выполнении условия (8.144) действительная часть (8.138) будет положительной, если $\alpha_m > \beta_m$:

$$B - 1 - m^2\pi^2 D_X > A^2 + m^2\pi^2 D_Y,$$

откуда находим

$$B > \tilde{B} = 1 + A^2 + m^2\pi^2 (D_X + D_Y). \tag{8.145}$$

Подчеркнем, что неустойчивыми оказываются только те колебательные моды, для которых одновременно выполняются условия (8.144) и (8.145).

4. Значения выражения (8.138) оказываются положительными при выполнении условия

$$\alpha_m \beta_m - A^2 B > 0,$$

или

$$B > B_m = 1 + A^2 \left[\frac{D_X}{D_Y} + \frac{1}{m^2\pi^2 D_Y}\right] + m^2\pi^2 D_X. \tag{8.146}$$

Если рассматривать выражение (8.146) как функцию непрерывной переменной m, становится очевидным, что данная функция имеет минимум в точке

$$m = \mu = \sqrt{\frac{A}{\pi^2\sqrt{D_X D_Y}}},$$

равный

$$B_\mu = \sqrt{1 + A\sqrt{\frac{D_X}{D_Y}}}.$$

В связи с тем, что переменная m является целочисленной первой неустойчивой неколебательной модой, будет мода, ближайшая к найденной выше точке минимума (μ, B_μ).

Рассмотрим, следуя [7], подходы, используемые для нахождения численного решения параболических уравнений, на примере одномерного уравнения диффузии

$$\frac{\partial \varphi}{\partial t} = \frac{\partial^2 \varphi}{\partial x^2} + S\left(x, t\right). \tag{8.147}$$

Как правило, для аппроксимации производных используются конечные разности на равномерной сетке с числом узлов $N+1$ и шагом по пространству $h = 1/M$ при этом производная по временному шагу Δt аппроксимируется простейшей разностной формулой первого порядка точности.

Будем считать, что независимая переменная $x \in [0, 1]$ и заданы граничные условия Дирихле, определяющие значении функции на границах интервала. Используя для шага по времени обозначение

$$\varphi^n \equiv \varphi\left(t_n\right), \; t_n = n\Delta t,$$

запишем конечноразностную аппроксимацию в следующем виде:

$$\frac{\varphi_i^{n+1} - \varphi_i^n}{\Delta t} = \frac{1}{h^2}\left(\delta^2 \varphi^n\right)_i + S_i^n, \tag{8.148}$$

где h — шаг пространственной сетки, $\left(\delta^2 \varphi^n\right)_i$ — оператор второй производной в i-м узле координатной сетки на n-м временном шаге, для вычисления которой можно, например, использовать трехточечную разностную формулу:

$$\delta^2 \varphi_i^n = \frac{\varphi_{i+1}^n - 2\varphi_i^n + \varphi_{i-1}^n}{2h^2}.$$

Разностная схема (8.148) называется явной, поскольку по заданным значениям функции φ в начальный момент времени можно вычислить ее значения в последующие моменты времени. Выражение (8.148) можно записать в матричной форме:

$$\varphi^{n+1} = \left(1 - H\Delta t\right)\varphi^n + S^n \Delta t, \tag{8.149}$$

где $\left(H\varphi\right)_i = -\frac{1}{h^2}\left(\delta^2 \varphi\right)_i$.

Проиллюстрируем использование явной схемы на примере уравнения (8.147) с $S \equiv 0$ с граничными условиями $\varphi(0) = \varphi(1) = 0$, предположив, что начальные условия задаются функцией

$$\varphi\left(x, t = 0\right) = e^{-20\left(x - \frac{1}{2}\right)^2} - e^{-20\left(x - \frac{2}{2}\right)^2} - e^{-20\left(x + \frac{1}{2}\right)^2}, \tag{8.150}$$

график которой представлен на рис. 8.31.

Аналитическое решение рассматриваемого уравнения с начальным условием (8.150) имеет вид:

$$\varphi\left(x, t\right) = \tau^{-\frac{1}{2}}\left[e^{-20\left(x - \frac{1}{2}\right)^2/\tau} - e^{-20\left(x - \frac{3}{2}\right)^2/\tau} - e^{-20\left(x + \frac{1}{2}\right)^2/\tau}\right], \tag{8.151}$$

где $\tau = 1 + 80t$.

Для нахождения численного решения создадим m-функции `FNGAUSS.m`, возвращающую значения базовой функции, используемой для конструирования функции, возвращающей точное решение, и `FNEXACT.m` — функция, возвращающая точное решение. Ниже представлены листинги данных файлов.

Рис. 8.31. График модельной функции (8.150)

```
% листинг файла FNGAUSS.m
function z=FNGAUSS(t,x)
% функция, возвращающая значение
% базовой функции, используемой для
% конструирования точного решения
% уравнения (8.147)
z=exp((-20*(x-0.5)^2)/(1+80*t))/(1+80*t)^0.5;

% листинг файла FNEXACT.m
function  z=FNEXACT(t,x)
% функция, возвращающая точное решение
% уравнения (8.147) с S=0
z=FNGAUSS(t,x)-FNGAUSS(t,x-1)-FNGAUSS(t,x+1);
```

Далее необходимо выполнить следующую последовательность команд, сохраненную нами в файле `Glava8_7.m`:

```
% Листинг файла Glava8_7.m
N=25; % число узлов координатной сетки
h=1/(N-1); % шаг координатной сетки
dt=10^-3; % шаг по времени
dTH=dt/h^2; % отношение шага по времени
            % к квадрату шага по координате
NT=50; % число шагов по времени

% задание начальных условий
t=0;
Phi(1)=0;
Phi(N)=0;
for i=2:N-1
    Phi(i)=FNEXACT(t,(i-1)*h);
```

```
end;
S=Phi';
S1=Phi';

% вычисление численного решения уравнения (8.147)
for i=1:NT  % цикл по времени
    Pt=0;
    for m=2:N-1 % цикл по координате
        Ps=Phi(m)+dTH*(Pt+Phi(m+1)-2*Phi(m));
        Pt=Phi(m);
        Phi(m)=Ps;
    end;
    S=[S,Phi']; % сохранение решения
                % для каждого момента времени
end;
S=S';

% вычисление точного решения
% уравнения (8.147)
for i=2:NT+1
     for m=2:N-1
         Phi(m)=FNEXACT((i-1)*dt,(m-1)*h);
     end;
    S1=[S1,Phi']; % сохранение точного решения
                  % для каждого момента времени
end;
S1=S1';

% визуализация точного и численного решений
i=1:N;
x=(i-1)*h;
i=1:NT+1;
t=(i-1)*dt;
[X Y]=meshgrid(x,t);

figure
surf(X,Y,S)

figure
surf(X,Y,S1)

% визуализация разности между точным и
% численным решениями
figure
surf(X,Y,S1-S)
```

Рис. 8.32. Численное решение уравнения (8.147) с $S = 0$, $\Delta t = 10^{-3}$

Рис. 8.33. Разность между численным и точным решениями уравнения (8.147) с $S = 0$, $\Delta t = 10^{-3}$

Рис. 8.34. Конфигурация системы в момент времени $t = 2000$

Результат выполнения, приведенных выше команд представлен на рис. 8.32–8.34.

Уменьшение шага по времени приводит, как и ожидалось, к увеличению точности численного решения, что иллюстрирует рис. 8.35, на котором представлена разность между численным и точным решениями.

Отметим еще одну особенность численного решения уравнения (8.147) с $S = 0$, $\Delta t = 10^{-3}$, вычисленного на временном интервале $[0, 0{,}18]$ (180 шагов по времени), которая видна из рис. 8.36. В конце рассматриваемого временного интервала численное решение становится осциллирующим, что противоречит физическим представлениям об изучаемом процессе, в связи с чем имеются веские основания считать это решение неустойчивым.

Дадим аналитические пояснения полученного результата. Пусть ψ_λ — собственные функции дискретного оператора, а ϵ_λ — соответствующие собственные значения. Так как оператор H эрмитов, все его собственные значения действительны, а собственные векторы могут быть ортонормированными. Решение уравнения в любой момент времени можно разложить по базису, состоящему из собственных функций ψ_λ:

$$\varphi^n = \sum_\lambda \varphi^n_\lambda \psi_\lambda.$$

Рис. 8.35. Разность между численным и точным решениями уравнения (8.147) с $S = 0$, $\Delta t = 10^{-4}$

Рис. 8.36. Разность между численным и точным решениями уравнения (8.147) с $S = 0$, $\Delta t = 10^{-3}$, $t \in [0, 0{,}18]$

Точная эволюция во времени решения рассматриваемого уравнения определяется выражением

$$\varphi^n = e^{-nH\Delta t}\varphi^0.$$

Следовательно, каждая компонента решения должна изменяться во времени по закону

$$\varphi_\lambda^n = e^{-n\varepsilon_\lambda \Delta t}\varphi_\lambda^0.$$

Данный закон соответствует правильному поведению решений уравнения диффузии. При этом коротковолновые компоненты решения (соответственно, с большими собственными числами) затухают быстрее, и потому решение с течением времени стремится стать более гладким.

Однако из (8.149) видно, что эволюция численного решения по явной схеме будет определяться коэффициентами разложения

$$\varphi_\lambda^n = (1 - \varepsilon_\lambda \Delta t)^n \varphi_\lambda^0. \tag{8.152}$$

Из (8.152) видно, что пока шаг Δt выбирается достаточно малым, множитель

$$(1 - \varepsilon_\lambda \Delta t)^n$$

достаточно хорошо аппроксимирует функцию

$$\exp\left[-n\varepsilon_\lambda \Delta t\right]$$

и коротковолновые решения, в полном соответствии с физическими представлениями о процессе диффузии, будут затухать с течением времени. Однако при достаточно большом шаге по времени оказывается, что один или несколько сомножителей $(1 - \varepsilon_\lambda \Delta t)$ по абсолютной величине может превосходить единицу. В этом случае погрешности вычислений будут нарастать с течением времени и вскоре станут доминирующими.

Получим оценку предельного значения Δt в рассматриваемой задаче. Непосредственной подстановкой в уравнение (8.149) можно убедиться в том, что функции

$$(\psi_\lambda)_i = \sin\frac{\lambda \pi i}{N}$$

являются ненормированными собственными функциями оператора H с точными граничными условиями на сетке из $N+1$ узлов для

$$\lambda = 1, 2, \ldots, N-1,$$

а соответствующие собственные значения равны

$$\varepsilon_\lambda = \frac{4}{h^2}\sin^2\frac{\lambda\pi}{2N}.$$

Наибольшее собственное значение оператора H равно $\varepsilon_{N-1} \approx 4/h^2$. Оно соответствует собственному вектору, компоненты которого меняют знак при переходе от

одного сеточного узла к другому. Из условия $|1 - \varepsilon_{N-1}\Delta t| < 1$ получаем искомое ограничение на временной шаг Δt:

$$\Delta t \leq \frac{h^2}{2}. \tag{8.153}$$

Для рассмотренного примера $\Delta t < 8,6806 \cdot 10^{-4}$.

***Задача* 8.19**

Используя файл `Glava8_7.m`, листинг которого приведен выше:

1. *Проверьте справедливость соотношения (8.153).*
2. *Покажите, что неустойчивость возникает и при других начальных условиях, а ошибки округления приводят к неустойчивости даже при использовании в качестве собственной функции оператора H с наименьшим собственным значением точного решения.*

Один из способов улучшения устойчивости рассмотренной выше явной схемы состоит в том, чтобы заменить вторую производную по пространству на производную от решения в новый момент времени, сохранив при этом общий вид уравнения (8.148):

$$\frac{\varphi_i^{n+1} - \varphi_i^n}{\Delta t} = \frac{1}{h^2}\left(\delta^2 \varphi^{n+1}\right)_i + S_i^n. \tag{8.154}$$

Схема вычислений, использованная в (8.154), называется неявной, поскольку неизвестные значения φ^{n+1} входят в обе части данного уравнения. Разрешив данное уравнение относительно φ^{n+1}, получаем следующую вычислительную формулу:

$$\varphi^{n+1} = \frac{1}{1 + H\Delta t}\left[\varphi^n + S^n \Delta t\right]. \tag{8.155}$$

Вычислительная схема (8.155) эквивалентна с точностью до членов порядка Δt вычислительной схеме (8.149). Однако она оказывается боле предпочтительной, поскольку здесь можно использовать бóльшие шаги по времени, собственные шаги оператора $(1 + H\Delta t)^{-1}$ равны $(1 + \varepsilon_\lambda \Delta t)^{-1}$ и, следовательно, все эти собственные значения по абсолютной величине оказываются меньше единицы для любого шага Δt. Следовательно, каждая компонента решения затухает на каждом шаге по времени. Отметим, однако, что закон затухания наиболее быстрых колебательных компонентов оказывается неэкспоненциально быстрым, но данное обстоятельство оказывается не столь существенным, поскольку при правильной дискретизации по пространственной координате амплитуда данных компонентов оказывается небольшой. Как следствие, данные компоненты не усиливаются с течением времени. Изменение во времени медленно меняющихся компонентов решения, соответствующих малым собственным значениям, напротив, оказывается экспоненциальным.

Можно получить еще один вариант устойчивой неявной схемы, используя в (8.154) среднее значение второй производной по двум равно отстоящим моментам времени:

$$\frac{\varphi_i^{n+1} - \varphi_i^n}{\Delta t} = \frac{1}{h^2}\left[\delta^2\left(\frac{1}{2}\left[\varphi^{n+1} - \varphi^n\right]_i\right)\right] + S_i^n. \tag{8.156}$$

Из (8.156) видно, что эволюция решения в данном случае определяется уравнением

$$\varphi^{n+1} = \frac{1}{1+\frac{1}{2}H\Delta t}\left[\left(1-\frac{1}{2}H\Delta t\right)\varphi^n + S^n\Delta t\right]. \tag{8.157}$$

Отметим один из потенциальных недостатков вычислительной схемы (8.155), состоящий в том, что для нахождения φ^{n+1} на каждом шаге по времени приходится решать трехдиагональную систему линейных уравнений. Данная процедура эквивалентна применению матрицы, обратной к матрице $1 + H\Delta t$, к вектору $\varphi^n + S^n\Delta t$. Поскольку элементы обратной матрицы не зависят от времени, данная матрица может быть найдена только один раз в начале вычисления численного решения, и далее применяться на каждом шаге по времени. Напомним, что непосредственное применение обратной матрицы требует порядка N^2 арифметических операций на каждом шаге по времени. Однако для трехдиагональных матриц, возникающих, например, при решении (8.155), существует алгоритм, требующий порядка N операций.

Проведем обсуждение данного алгоритма, называемого методом прогонки, на примере решения системы линейных уравнений, возникающей при использовании вычислительной схемы (8.155):

$$A\vec{x} = \vec{b}, \tag{8.158}$$

где A — трехдиагональная матрица, $\vec{x}$ — вектор-столбец неизвестных, $\vec{b}$ — вектор-столбец свободных членов. Запишем данную систему в виде, аналогичном (8.158):

$$A_j^-\varphi_{j-1} + A_j^0\varphi_j + A_j^+\varphi_{j+1} = b_j, \tag{8.159}$$

где

$$b_j = \varphi_j^n + S_j^n\Delta t,$$
$$A_j^0 = 1 + \frac{2\Delta t}{h^2},$$
$$A_j^\pm = -\frac{\Delta t}{h^2}.$$

Для решения системы уравнений (8.159) предположим, что искомое решение удовлетворяет рекуррентному соотношению, связывающему значения x_j в соседних узлах j и $j+1$:

$$\varphi_{j+1} = \alpha_j\varphi_j + \beta_j. \tag{8.160}$$

где α_j, β_j — неизвестные коэффициенты, которые необходимо определить.

Подставляя (8.160) в (8.159), получим

$$A_j^-\varphi_{j-1} + A_j^0\varphi_j + A_j^+\left(\alpha_j\varphi_j + \beta_j\right) = b_j. \tag{8.161}$$

Разрешив (8.161) относительно φ_j, получим

$$\varphi_j = \gamma_j A_j^-\varphi_{j-1} + \gamma_j\left(A_j^+\beta_j - b_j\right), \tag{8.162}$$

где

$$\gamma_j = -\frac{1}{A_j^0 + A_j^+\alpha_j}. \tag{8.163}$$

Сравнивая (8.162) и (8.160), видим, что они тождественны, если коэффициенты α_j, β_j удовлетворяют следующим рекуррентным соотношениям:

$$\alpha_{j-1} = \gamma_j A_j^- , \tag{8.164}$$

$$\beta_{j-1} = \gamma_j \left(A_j^+ \beta_j - b_j \right) . \tag{8.165}$$

Таким образом, решение трехдиагональной системы уравнений осуществляется следующей последовательностью действий:

1. Вычисление коэффициентов γ_j, α_j, β_j по формулам (8.163)–(8.165), в которых $j = N-2, N-3, \ldots, 0$, α_{N-1}, $\beta = \varphi_N$ (обратный ход).

2. Вычисление φ_j по формуле (8.162), в которой $j = 0, 1, \ldots, N-1$, где φ_0 известно из начальных условий (прямой ход).

Описанный алгоритм решения системы линейных уравнений, состоящий из обратного и прямого ходов, требует $2N$ арифметических операций. Для реализации описанного алгоритма необходимо выполнить последовательность команд, сохраненную нами в файле `Glava8_8.m`:

```
% листинг файла Glava8_8.m
N=25; % число узлов
      % координатной сетки
h=1/(N-1); % шаг координатной сетки
dt=10^-3; % шаг по времени
dTH=dt/h^2; % отношение
            % шага по времени
            % к квадрату шага по координате
NT=180; % число шагов по времени

% задание начальных условий
t=0;
Phi(1)=0;
Phi(N)=0;
for i=2:N-1
    Phi(i)=FNEXACT(t,(i-1)*h);
end;
S=Phi';
S1=Phi';

% вычисление коэффициентов
% alpha и gamma
AP=-dTH; % коэффициент A[0,j]
AZ=1+2*dTH; % коэффициент A[+-,j]
Alpha(N)=0; % начальное значение
            % коэффициента alpha
```

```
Gamma(N)=-1/AZ; % начальное значение
                % коэффициента gamma

% обратный ход
i=N;
while i>=2
    Alpha(i-1)=Gamma(i)*AP;
    Gamma(i-1)=-1/(AZ+AP*Alpha(i-1));
    i=i-1;
end;

% вычисление численного решения
% уравнения (8.147)
for m=1:NT % цикл по времени
    % вычисление beta
    Beta(N)=0; % начальное значение beta
    % обратный ход
     i=N;
    while i>=2
        Beta(i-1)=Gamma(i)*(AP*Beta(i)-Phi(i));
        i=i-1;
    end;

    % вычисление новых значений phi
    Phi(1)=0;
    for i=2:N
        Phi(i)=Alpha(i-1)*Phi(i-1)+Beta(i-1);
    end;
    S=[S,Phi']; % сохранение численного решения
                % для каждого момента времени
end;
S=S';

% вычисление точного решения
% уравнения (8.147)
for i=2:NT+1
     for m=2:N-1
         Phi(m)=FNEXACT((i-1)*dt,(m-1)*h);
     end;
    S1=[S1,Phi'];% сохранение точного решения
                 % для каждого момента времени
end;
S1=S1';
```

```
% визуализация разности между точным и
% численным решениями
i=1:N;
x=(i-1)*h;
i=1:NT+1;
t=(i-1)*dt;
[X Y]=meshgrid(x,t);
figure
surf(X,Y,S1-S)
```

Результат выполнения приведенной выше последовательности команд представлен на рис. 8.37.

Рис. 8.37. Разность между точным решением и численным решением уравнения (8.147) с $S = 0$, $\Delta t = 10^{-3}$, полученным при использовании неявной разностной схемы

Из рис. 8.37 видно, что, в отличие от случая использования явной разностной схемы, полученное численное решение оказывается устойчивым.

Задача 8.20

1. *Модифицируйте файл* `Glava8_8.m`*, листинг которого приведен выше, для решения уравнения по вычислительной схеме (8.157) и численно исследуйте устойчивость данной схемы.*

2. *Исследуйте влияние на решение различных линейных комбинаций φ^n и φ^{n+1} в правой части уравнения (8.156). Например,*

$$\frac{3}{4}\varphi^n + \frac{1}{4}\varphi^{n+1}.$$

***Задача* 8.21**

Простой способ задания граничных условий Неймана

$$\left.\frac{\partial\varphi}{\partial x}\right|_0 = g$$

сводится к условию $\varphi_1 = \varphi_0 + hg$.

1. *Покажите, что в этом случае начальное значение в формуле для прямого хода (8.160) равно*

$$\varphi_0 = \frac{hg - \beta_0}{\alpha_0 - 1}.$$

2. *Запишите аналогичное выражение для φ_0, вытекающее из более точного условия*

$$\varphi_1 = \varphi_{-1} + 2hg.$$

3. *Какие значения следует использовать для начала расчета коэффициентов α и β по формулам обратного хода (8.163)–(8.163), если на правой границе задано условие Неймана?*

4. *Модифицируйте файл* `Glava8_9.m`*, листинг которого приведен выше, так, чтобы включить граничное условие*

$$\left.\frac{\partial\varphi}{\partial x}\right|_{x=0} = 0,$$

которое с физической точки зрения соответствует наличию изолятора в тепловой задаче.

5. *Покажите, что рассмотренная выше схема обращения трехдиагональных матриц неприменима в случае периодических граничных условий:*

$$\varphi_0 = \varphi_N.$$

Таким образом, проведенные выше исследования показывают, что для решения одномерного уравнения диффузии наиболее пригодным оказываются неявные схемы, требующие обращение трехдиагональной матрицы. Полученные выше результаты свидетельствуют о целесообразности распространения данного подхода на случай двух и более переменных.

Дискретизация двумерного уравнения диффузии по аналогии с одномерной задачей описывается уравнением

$$\varphi^{n+1} = \frac{1}{1 + H\Delta t}\varphi^n, \tag{8.166}$$

где

$$(H\varphi)_{ij} = -\frac{1}{h^2}\left[\left(\delta_i^2\varphi\right)_{ij} + \left(\delta_j^2\varphi\right)_{ij}\right].$$

Несмотря на то что матрица H является сильно разреженной, она оказывается не трехдиагональной, в отличие от одномерного случая, поэтому рассмотренный ранее алгоритм в данной ситуации оказывается неприменим, так как можно показать, что никакой перестановкой строк или столбцов H матрицу нельзя привести к трехдиагональному виду.

Однако из того, что оператор H может быть записан в виде суммы операторов, каждый из которых содержит разности только по индексу i или j:

$$H = H_i + H_j, \tag{8.167}$$

где

$$H_{i,j} = -\frac{1}{h^2}\delta_{i,j}^2,$$

следует, что (8.166) с точностью до членов порядка Δt может быть записано в виде

$$\varphi^{n+1} = \frac{1}{1 + H_i\Delta t}\frac{1}{1 + H_j\Delta t}\varphi^n. \tag{8.168}$$

Из (8.168) видно, что полученное уравнение может быть решено точно, поскольку каждое из требуемых обращений сводится к решению системы с трехдиагональной матрицей. В частности, если ввести вспомогательную функцию $\varphi^{n+1/2}$, то (8.168) можно записать в виде

$$\begin{aligned} \varphi^{n+1/2} &= \frac{1}{1 + H_j\Delta t}\varphi^n, \\ \varphi^{n+1} &= \frac{1}{1 + H_i\Delta t}\varphi^{n+1/2}. \end{aligned} \tag{8.169}$$

Из (8.169) видно, что оператор $(1 + H_j\Delta t)^{-1}$ действует по формулам прямого и обратного ходов по индексу j независимо для каждого значения i, аналогичным образом действует оператор — по индексу i независимо для каждого значения. Описанная вычислительная схема относится к классу итерационных методов переменных направлений. Можно показать, что вычислительная схема (8.169) является устойчивой при любых значениях шага по времени и легко обобщается на трехмерный случай.

Завершив обсуждение общих идей решения параболических ДУ, вернемся к решению система ДУ (8.116), состоящей из двух уравнений первого порядка параболического типа. Для нахождения ее численного решения необходимо создать файлы `Coeff.m`, содержащий описание функции, возвращающей значения коэффициентов α и β, используемых в методе продольно-поперечной прогонки, и `Density.m`, содержащий описание функции, возвращающей численные решения (8.116). Ниже представлены листинги данных файлов.

```
% листинг файла Coeff.m
function S=Coeff(Nx,Ny,C);
% функция, возвращающая значения
% коэффициентов Alpha и Gamma,
% используемых в методе
% продольно-поперечной прогонки

% вычисление коэффицентов,
% используемых при проходе
% в направлении оси OX

% задание краевых условий
% с нулевыми потоками
ALPHAxx(Nx)=1;
ALPHAyx(Ny)=1;

% обратный ход
i=Nx;
while i>=2
    GAMMAxx(i)=1/(C(6)+C(5)*ALPHAxx(i));
    ALPHAxx(i-1)=-C(5)*GAMMAxx(i);
    GAMMAyx(i)=1/(C(8)+C(7)*ALPHAyx(i));
    ALPHAyx(i-1)=-C(7)*GAMMAyx(i);
    i=i-1;
end;

% вычисление коэффицентов,
% используемых при проходе
% в направлении оси OY

% задание краевых условий
% с нулевыми потоками
ALPHAxy (Ny)=1;
ALPHAyy(Ny)=1;

% обратный ход
j=Ny;
while j>=2
    GAMMAxy(j)=1/(C(6)+C(5)*ALPHAxy(j));
    ALPHAxy(j-1)=-C(5)*GAMMAxy(j);
    GAMMAyy(j)=1/(C(8)+C(7)*ALPHAyy(j));
    ALPHAyy(j-1)=-C(7)*GAMMAyy(j);
    j=j-1;
end;
if not(Nx==Ny)
```

```
    if Nx<Ny
        for i=Nx+1:Ny
            GAMMAxx(i)=0;
            ALPHAxx(i)=0;
            GAMMAyx(i)=0;
            ALPHAyx(i)=0;
        end;
    else
        for i=Ny+1:Nx
            GAMMAxy(i)=0;
            ALPHAxy(i)=0;
            GAMMAyy(i)=0;
            ALPHAyy(i)=0;
        end;
    end;
end;
S=[ALPHAxx ;ALPHAxy ;...
      ALPHAyx; ALPHAyy ;...
      GAMMAxx ;GAMMAxy;...
      GAMMAyx;GAMMAyy];
S=S';

% листинг файла Density.m
function z=Density(Nx,Ny,dT,C,Cc,X,Y)
% Функция, возвращающая
% численное решение системы ДУ (8.116)

% для каждого y выполняется цикл по x
for j=1:Ny+1
    % вычисление коэффициентов BETA
    BETAxx(Nx)=0;
    BETAyx(Nx)=0;
    i=Nx;
    % обратный ход
    while i>=2
        X2=X(i,j)^2*Y(i,j)*dT;
        Sx=X(i,j)*C(4)+C(3)+X2;
        Sy=Y(i,j)+C(2)*X(i,j)-X2;
        BETAxx(i-1)=(Sx-C(5)*BETAxx(i))*Cc(i,5);
        BETAyx(i-1)=(Sy-C(7)*BETAyx(i))*Cc(i,7);
        i=i-1;
    end;

    % прямой ход
    % начальные значения для краевых условий
    % с нулевыми потоками
```

```
    X(1,j)=BETAxx(1)/(1-Cc(1,1));
    Y(1,j)=BETAyx(1)/(1-Cc(1,3));
    for i=1:Nx-1
        X(i+1,j)=Cc(i,1)*X(i,j)+BETAxx(i);
        Y(i+1,j)=Cc(i,3)*Y(i,j)+BETAyx(i);
    end;
end;

% для каждого x выполняем цикл по y
for i=1:Nx+1
    BETAxy(Ny)=0;
    BETAyy(Ny)=0;
    % вычисление коэффициентов BETA
    j=Ny;
    % обратный ход
    while j>=2
        BETAxy(j-1)=(X(i,j)-C(5)*BETAxy(j))*Cc(j,6);
        BETAyy(j-1)=(Y(i,j)-C(7)*BETAyy(j))*Cc(j,8);
        j=j-1;
    end;
    % прямой ход
    % начальные значения для краевых условий
    % с нулевыми потоками
    X(i,1)=BETAxy(1)/(1-Cc(1,2));
    Y(i,1)=BETAyy(1)/(1-Cc(1,8));
   % вычисление плотностей первого и
   % второго реагентов
    for j=1:Ny-1
        X(i,j+1)=Cc(j,2)*X(i,j)+BETAxy(j);
        Y(i,j+1)=Cc(j,4)*Y(i,j)+BETAyy(j);
     end;
end;
z=[X Y];
```

Далее для нахождения численного решения (8.116) необходимо выполнить следующую последовательность команд, сохраненную нами в файле `Glava8_9.m`:

```
% листинг файла Glava8_9.m
% задание концентраций реагентов
A=2;
B=7;

% задание коэффициентов диффузии реагентов
Dx=10^-3;
Dy=10^-3;
% задание числа узлов координатной сетки
```

```
Nx=25;
Ny=25;

% задание шага по времени
dT=2*10^-2;

% создание нулевых матриц,
% в которые будет возвращаться
% численное решение системы ДУ (8.116)
X=zeros(Nx+1,Ny+1);
Y=zeros(Nx+1,Ny+1);

% генерация матрицы, содержащей
% случайные числа с равномерным
% законом распределения
% на интервале [-0.2,0.2]
Noise=-0.2+0.4*rand(Nx+1,Ny+1);

% инициализация начального распределения
% плотности первого реагента
X=A*(1-Noise);

% визуализация распределения плотности
% первого реагента
figure
surf(X)
axis([1 Nx+1 1 Ny+1])
axis off
shading interp
view([0 90])
colormap copper

% генерация матрицы, содержащей
% случайные числа с
% равномерным законом распределения
% на интервале [-0.2,0.2]
Noise=-0.2+0.4*rand(Nx+1,Ny+1);

% инициализация начального распределения
% плотности второго реагента
Y=B/A*(1-Noise);

% визуализация распределения плотности
% второго реагента
figure
```

```
surf(Y)
axis([1 Nx+1 1 Ny+1])
axis off
shading interp
view([0 90])
colormap copper

% вычисление констант,
% используемых далее в программе
C(1)=1/(Nx-1);
C(2)=B*dT;
C(3)=A*dT;
C(4)=1-(B+1)*dT;
C(5)=-Dx*dT/(C(1))^2;
C(6)=1-2*C(5);
C(7)=-Dy*dT/C(1)^2;
C(8)=1-2*C(7);

% вычисление коэффициентов
% продольно-поперечной прогонки
Cc=Coeff(Nx,Ny,C);

% вычисление численного решения
% системы ДУ (8.116)
% на первом шаге интегрирования
S2(:,:,1)=Density(Nx,Ny,dT,C,Cc,X,Y);

% вычисление численного решения
% системы ДУ (8.116)
Nt=1500; % число шагов по времени
for i=1:Nt
    S2(:,:,i+1)=Density(Nx,Ny,dT,...
                        C,Cc,...
                        S2(:,1:Ny+1,i),...
                        S2(:,Ny+2:2*Ny+2,i));
end;

% вычисление зависимостей концентраций
% первого и второго реагентов в точке
% i=10, j=10 от времени
for i=1:Nt
    X1=S2(:,1:Ny+1,i);
    Y1=S2(:,Ny+2:2*Ny+2,i);
    S(i,1)=(i-1)*dT;
    S(i,2)=X1(10,10);
```

```
    S(i,3)=Y1(10,10);
end;

% визуализация зависимостей
% концентраций первого и
% второго реагентов в точке
% i=10, j=10 от времени
figure
plot(S(:,1),S(:,2),S(:,1),S(:,3))

% создание анимационного клипа,
% демонстрирующего изменение
% во времени распределения
% концентрации второго реагента
figure
k=1;
for i=1:Nt
  surf(S2(:,1:Ny+1,i))
  axis([1 Nx+1 1 Ny+1])
  axis off
  shading interp
  view([0 90])
  F(k)=getframe;
  k=k+1;
end;
movie(F,1)
```

Результаты выполнения, описанной выше последовательности команд, представлены на рис. 8.38–8.41.

Рис. 8.38. Конфигурация системы в момент времени $t = 0$

Рис. 8.39. Конфигурация системы в момент времени $t = 10$

Рис. 8.40. Конфигурация системы в момент времени $t = 2000$

Рис. 8.41. Визуализация концентрации второго реагента в момент времени $t = 30$

***Задача* 8.22**

С помощью файла `Glava8_9.m`, *листинг которого приведен выше, исследуйте решения системы ДУ (8.116) при различных значениях коэффициентов* B, $D_{X,Y}$. *(Разумными значениями коэффициентов диффузии могут быть например значения* $D_X = 10^{-3}$, $D_Y = 10^{-4}$, *далее, используя результаты линейного анализа устойчивости решения, можно определить значения коэффициента* B*).*

***Задача* 8.23**

Модифицируйте файл `Glava8_9.m`, *листинг которого приведен выше, и исследуйте решения системы ДУ (8.116) в условиях, когда на однородную равновесную конфигурацию наложены начальные возмущения:*

1) *в виде гармонической функции;*

2) *в виде функции, принимающей в каждом узле координатной сетки случайные значения.*

Убедитесь в том, что вы можете обнаружить случаи, когда:

1) *система релаксирует к исходному равновесному состоянию;*

2) *система асимптотически приближается к независящему от времени нетривиальному решению (диссипативная структура);*

3) *система имеет решение, осциллирующее в пространстве и во времени. Во всех случаях проверяйте, чтобы выбранный вами шаг интегрирования по времени был достаточен, чтобы получить численные решения системы ДУ (8.116) с приемлемой точностью, сравнивая, например, численные решения, полученные для нескольких различных значений* Δt.

ЛИТЕРАТУРА К ГЛАВЕ 8

1. *Ландау Л. Д., Лифшиц Е. М.* Механика. М.: Наука, 1958.
2. *Мандельштам И. Л.* Лекции по теории колебаний. М.: Наука, 1972.
3. *Матвеев А. Н.* Механика и теория относительности. М.: Высшая школа, 1986.
4. *Гулд Х., Тобочник Я.* Компьютерное моделирование в физике. М.: Мир, 1990. Ч. 1.
5. *Трубецков Д. И.* Введение в синергетику. Колебания и волны. М.: Эдиториал УРСС, 2003.
6. *Рабинович М. И., Трубецков Д. И.* Введение в теорию колебаний и волн. М.; Ижевск: НИЦ «Регулярная и хаотическая динамика», 2000.
7. *Кунин С.* Вычислительная физика. М.: Мир, 1992.
8. *Федоренко Р. П.* Введение в вычислительную физику. М.: МИФИ, 1994.

Глава 9

МОДЕЛИРОВАНИЕ ВОЛНОВЫХ ЯВЛЕНИЙ

9.1. ВВЕДЕНИЕ

В данной главе мы продолжаем рассмотрение колебательных движений, начатое в главе 8, в которой были рассмотрены гармонический осциллятор, а также линейный и нелинейным математические маятники. Здесь в качестве базовой математической модели выбрана линейная цепочка связанных гармонических осцилляторов. С помощью данной модели при достаточно большом числе частиц оказывается возможным изучить основные особенности волновых движений (в первую очередь показать, что энергия передается вдоль цепочки, хотя каждый осциллятор колеблется вблизи своего положения равновесия), осуществить в пределе непрерывной цепочки переход к линейному волновому уравнению. Далее мы рассмотрим такие фундаментальные понятия волновой теории, как бегущая волна, стоячая волна, волновой пакет, фазовая скорость, групповая скорость, дисперсия, интерференция, дифракция, рефракция, поляризация, принцип Ферма.

9.2. МОДЕЛИРОВАНИЕ СВОБОДНЫХ КОЛЕБАНИЙ ЦЕПОЧКИ СВЯЗАННЫХ ОСЦИЛЛЯТОРОВ

Модели, представляющие собой линейные цепочки (рис. 9.1), состоящие из конечного или бесконечного числа связанных осцилляторов, оказались весьма эффективными и в настоящее время используются в различных областях физики: физике твердого тела, физике сплошных сред, химической физике, радиофизике и др. (Например, в [1] для описания системы трех связанных электрических колебательных контуров использована модель, представляющая собой систему шариков с массами m_1, m_2, m_3, связанных между собой пружинками одинаковой жесткости k.) Используя модели линейных цепочек, оказывается возможным естественным образом осуществить переход к волновым процессам и ввести такие понятия, как длина волны, групповая скорость, фазовая скорость, дисперсия и др.

Рис. 9.1. Цепочка связанных гармонических осцилляторов

Отмеченные обстоятельства определяют целесообразность рассмотрения данных моделей в соответствующих курсах физики и компьютерного моделирования. Однако необходимо отметить два важных обстоятельства. Во-первых, аналитические решения уравнений движения длинных линейных цепочек ($N > 3$) могут быть получены только для относительно небольшого числа случаев [2]:

1. $k_0 = k_1 = \ldots = k_{N-1}, m_0 = m_1 = \ldots = m_{N-1}$;
2. $k_0 = k_2 = k_4 = \ldots, k_1 = k_3 = k_5 = \ldots, m_0 = m_1 = \ldots = m_{N-1}$;
3. $k_0 = k_1 = \ldots = k_{N-1}, m_0 = m_2 = m_4 = \ldots, m_1 = m_3 = m_5 = \ldots$;
4. $k_i = k \neq k_N - 1, i = 0, 1, \ldots, N-1, m_0 = m_1 = \ldots = m_{N-1}$;
5. $k_i = k, i = 0, 1, \ldots N, m_i = m \neq m_{N-1}, i = 0, 1, \ldots N-2$.

Во-вторых, большинство этих решений оказываются весьма громоздкими, и для их последующего анализа приходится использовать ПК.

Запишем уравнения движения для каждой массы колебательной системы, представленной на рис. 9.1:

$$\begin{aligned} &m_0\ddot{x}_0 = -k_0x_0 - k_1\left(x_0 - x_1\right), \\ &m_i\ddot{x}_i = -k_i\left(x_i - x_{i-1}\right) - k_{i+1}\left(x_i - x_{i+1}\right), \quad i = 1\ldots N-2, \\ &m_{N-1}\ddot{x}_{N-1} = -k_{N-1}\left(x_{N-1} - x_{N-2}\right) - k_Nx_{N-1}. \end{aligned} \tag{9.1}$$

Для удобства дальнейшего решения запишем уравнение (9.1), введя обозначение $\omega^2_{\alpha\beta} = k_\alpha/m_\beta$, в следующем виде:

$$\begin{aligned} &\ddot{x}_0 = -\omega^2_{00}x_0 - \omega^2_{10}\left(x_0 - x_1\right) \\ &\ddot{x}_i = -\omega^2_{ii}\left(x_i - x_{i-1}\right) - \omega^2_{i+1i}\left(x_i - x_{i+1}\right), \quad i = 1\ldots N-2, \\ &\ddot{x}_{N-1} = -\omega^2_{N-1N-1}\left(x_{N-1} - x_{N-2}\right) - \omega^2_{NN-1}x_{N-1}. \end{aligned} \tag{9.2}$$

Следуя общему подходу к решению рассматриваемой задачи, изложенному в [3], ищем решение системы дифференциальных уравнений (9.2) в виде

$$x_i = A_ie^{j\omega t}, \quad i = 0,\ 1..N-1. \tag{9.3}$$

Подставив (9.3) в систему (9.2), сгруппировав члены, пропорциональные A_i, и записав систему в матричном виде, получим

$$\boldsymbol{B}\cdot\boldsymbol{A} = 0, \tag{9.4}$$

где

$$\boldsymbol{A} = \begin{pmatrix} A_0 \\ A_1 \\ \vdots \\ A_{N-1} \end{pmatrix},$$

$\boldsymbol{B}$ — трехдиагональная матрица, элементы которой вычисляются по следующим правилам:

$$\begin{aligned}
&B_{00} = -\omega^2 + \omega_{00}^2 + \omega_{1\,0}^2, \quad B_{0\,1} = -\omega_{1\,0}^2,\\
&B_{ii-1} = -\omega_{ii}^2,\ B_{ii} = -\omega^2 + \omega_{ii}^2 + \omega_{i+1i}^2,\ B_{ii+1} = -\omega_{i+1i}^2,\ i = 2,..N-2,\\
&B_{N-1N-2} = -\omega_{N-1N-1}^2,\ B_{N-1N-1} = -\omega^2 + \omega_{N-1N-1}^2 + \omega_{NN-1}^2.
\end{aligned} \tag{9.5}$$

Необходимым и достаточным условием существования решения системы уравнений (9.4) является равенство нулю определителя матрицы $\boldsymbol{B}$ системы (9.4).

$$\left|\omega_{ij}^2 - \omega^2\right| = 0. \tag{9.6}$$

Уравнение (9.6), называемое характеристическим уравнением, является уравнением степени $N-1$ относительно ω^2. Оно имеет в общем случае $N-1$ различных вещественных положительных корней ω_α^2. Каждому собственному числу ω_α^2 соответствует собственный вектор $\vec{\Xi}_\alpha$, являющийся решением уравнения

$$\Omega \cdot \vec{\Xi}_\alpha = \omega_\alpha^2 \cdot \vec{\Xi}_\alpha, \tag{9.7}$$

где Ω — трехдиагональная матрица, элементы которой вычисляются по следующим правилам:

$$\begin{aligned}
&\Omega_{00} = \omega_{00}^2 + \omega_{10}^2, \ldots,\ \Omega_{01} = -\omega_{10}^2,\\
&\Omega_{i,i-1} = -\omega_{ii}^2, \Omega_{ii} = \omega_{ii}^2 + \omega_{i+1i}^2, \Omega_{ii} = \omega_{ii}^2 + \omega_{i+1i}^2, \Omega_{ii+1} = -\omega_{i+1i}^2,\ i = 2, \ldots, N-2,\\
&\Omega_{N-1N-2} = -\omega_{N-1N-1}^2,\ \Omega_{N-1N-1} = \omega_{N-1N-1}^2 + \omega_{NN-1}^2.
\end{aligned} \tag{9.8}$$

Частоту $\omega_\alpha, \alpha = 1, 2, \ldots, N-1$, называют частотой нормальных колебаний, а вектор $\vec{\Xi}_\alpha$ — вектором нормального колебания, отвечающего α-й частоте. Вектор нормального колебания $\vec{r}_\alpha$ меняется во времени по закону

$$\vec{r}_\alpha = \vec{\Xi}_\alpha \cos\left(\omega_\alpha t + \varphi_\alpha\right). \tag{9.9}$$

Общее решение системы дифференциальных уравнений (9.2) $\vec{x}(t)$ есть суперпозиция всех векторов нормальных колебаний $\vec{r}_\alpha$:

$$\vec{x}\,(t) = \sum_{\alpha=0}^{N-1} C_\alpha \vec{\Xi}_\alpha \cos\left(\omega_\alpha t + \varphi_\alpha\right), \tag{9.10}$$

где C_α, φ_s — произвольные постоянные, определяемые из начальных условий.

Скорость движения масс можно определить, продифференцировав (9.10) по времени:

$$\dot{\vec{x}}\,(t) = -\sum_{\alpha=0}^{N-1} C_\alpha \vec{\Xi}_\alpha \omega_\alpha \sin\left(\omega_\alpha t + \varphi_\alpha\right). \tag{9.11}$$

Для решения задачи Коши системы дифференциальных уравнений (9.2) необходимо задать значения координат $\vec{x}(0)$ и скоростей $\dot{\vec{x}}(0)$ каждого тела системы в начальный момент времени $t = 0$ и решить систему уравнений

$$\begin{aligned} \vec{x}\,(0) &= \sum_{\alpha=0}^{N-1} C_\alpha \vec{\Xi}_\alpha \cos(\varphi_\alpha), \\ \dot{\vec{x}}\,(0) &= -\sum_{\alpha=0}^{N-1} C_\alpha \vec{\Xi}_\alpha \omega_\alpha \sin(\varphi_\alpha) \end{aligned} \tag{9.12}$$

относительно неизвестных C_α, φ_α.

Запишем (9.12) в матричном виде

$$\begin{pmatrix} \vec{x}\,(0) \\ \dot{\vec{x}}\,(0) \end{pmatrix} = \begin{pmatrix} \Theta & Z \\ Z & \Theta_{\text{v}} \end{pmatrix} \cdot \begin{pmatrix} \vec{C} \\ \vec{C}_{\text{v}} \end{pmatrix}, \tag{9.13}$$

где

$$\Theta = \begin{pmatrix} \Xi_0^{(0)} & \Xi_0^{(1)} & \cdots & \Xi_0^{(N-1)} \\ \Xi_1^{(0)} & \Xi_1^{(1)} & \cdots & \Xi_1^{(N-1)} \\ \vdots & \vdots & \cdots & \vdots \\ \Xi_{N-1}^{(0)} & \Xi_{N-1}^{(1)} & \cdots & \Xi_{N-1}^{(N-1)} \end{pmatrix}, \tag{9.14}$$

$$\Theta_{\text{v}} = \begin{pmatrix} -\omega_0 \Xi_0^{(0)} & -\omega_1 \Xi_0^{(1)} & \cdots & -\omega_{N-1} \Xi_0^{(N-1)} \\ -\omega_0 \Xi_1^{(0)} & -\omega_1 \Xi_1^{(1)} & \cdots & -\omega_{N-1} \Xi_1^{(N-1)} \\ \vdots & \vdots & \cdots & \vdots \\ -\omega_0 \Xi_{N-1}^{(0)} & -\omega_1 \Xi_{N-1}^{(1)} & \cdots & -\omega_{N-1} \Xi_{N-1}^{(N-1)} \end{pmatrix}, \tag{9.15}$$

$$\vec{C} = \begin{pmatrix} C_0 \cos(\varphi_0) \\ \vdots \\ C_{N-1} \cos(\varphi_{N-1}) \end{pmatrix}, \tag{9.16}$$

$$\vec{C}_{\text{v}} = \begin{pmatrix} C_0 \sin(\varphi_0) \\ \vdots \\ C_{N-1} \sin(\varphi_{N-1}) \end{pmatrix}, \tag{9.17}$$

Z — нулевая матрица, размерности $(N-1) \times (N-9)$.

Система уравнений (9.13) оказывается нелинейной, однако блочная структура матрицы позволяет найти решение данной системы, не прибегая к численным методам. Для этого сначала, решив две линейные системы уравнений

$$\Theta \cdot \vec{C}_1 = \vec{x}\,(0)\,, \tag{9.18}$$

$$\Theta_{\text{v}} \vec{C}_2 = \dot{\vec{x}}\,(0)\,, \tag{9.19}$$

найдем векторы $\vec{C}_1$, $\vec{C}_2$, затем координаты вектора $\vec{C}$

$$C_i = \sqrt{C_i^2 + C_{\mathrm{v}i}^2},\ i = 0, 1, \ldots, N-1, \tag{9.20}$$

и далее значения начальных фаз каждого нормального колебания:

$$\varphi_i = \arctan\left(\frac{C_i}{C_{\mathrm{v}_i}}\right). \tag{9.21}$$

Отметим, что функция arctan на интервале $[0; 2\pi]$ является двузначной (рис. 9.2), поэтому для выбора правильных значений данной функции необходимо контролировать знаки числителя и знаменателя дроби в выражении (9.21).

Рис. 9.2. Цепочка связанных гармонических осцилляторов

Как очевидно из рис. 9.2, правильное значение угла выбирается по следующим правилам:

$$\vartheta = \begin{cases} \arctan\left(\frac{\sin(x)}{\cos(x)}\right) & \text{если } \sin(x) \geq 0, \cos(x) \geq 0, \\ \pi + \arctan\left(\frac{\sin(x)}{\cos(x)}\right) & \text{если } \sin(x) < 0, \cos(x) > 0, \\ \pi + \arctan\left(\frac{\sin(x)}{\cos(x)}\right) & \text{если } \sin(x) < 0, \cos(x) < 0, \\ 2\pi + \arctan\left(\frac{\sin(x)}{\cos(x)}\right) & \text{если } \sin(x) > 0, \cos(x) < 0. \end{cases} \tag{9.22}$$

Предваряя описание решения задачи об описании колебаний цепочки связанных осцилляторов, приведем алгоритм ее решения:

1. Задать число тел, образующих цепочку N.
2. Задать массы тел m_i, $i = 0, 1, \ldots, N-9$.
3. Задать значения коэффициентов жесткости пружин k_i, $i = 0, 1, N$. (Отметим, что для описания движения цепочки со свободным концом достаточно положить $k_0 = 0$ или $k_N = 0$.)
4. Вычислить элементы матрицы Ω в соответствии с (9.8).
5. Найти собственные числа ${\omega_\alpha}^2$ матрицы Ω.

6. Найти собственные векторы $\vec{\Xi}_\alpha$, соответствующие набору собственных частот ${\omega_\alpha}^2$.

7. Задать начальные условия $\vec{x}(0)$, $\dot{\vec{x}}(0)$.

8. Решить систему линейных уравнений (9.18), (9.19) относительно векторов $\vec{C}_1$ и $\vec{C}_2$ соответственно.

9. Вычислить координаты вектора $\vec{C}$ в соответствии с (9.20).

10. Вычислить значения начальных фаз нормальных колебаний φ_i в соответствии с (9.21).

11. Определить законы движения тел, образующих колебательную систему в соответствии с (9.10) и (9.11).

12. Провести анализ полученных законов движения.

Данный алгоритм в MATLAB реализуется следующей последовательностью действий, сохраненной нами в файле `Glava9_1.m`:

```
% листинг файла Glava9_1.m
clear all

% задание параметров
% колебательной системы
N=3; % число тел
     % колебательной системы
m=[1 2 1]; % массы тел
           % колебательной системы
k=[1 1 1 1]; % жесткости пружин
             % колебательной системы
R0=[-0.2 0 -0.3]; % смещения тел
                  % в момент времени t = 0
v0=[1 -3 0]; % скорости тел
             % в момент времени t = 0

% вычисление элементов матрицы
for alpha=1:N+1
     for beta=1:N
         omega(alpha,beta)=k(alpha)/m(beta);
    end;
end;
i=1:N;
j=1:N;

% вычисление элементов матрицы OMEGA
% в соответствии с (9.8)
```

```
OMEGA(1:N,1:N)=0;
for i=1:N
    if i==1
        OMEGA(i,i)=omega(1,1)+omega(2,1);
        OMEGA(1,2)=-omega(2,1);
    end;
   if i>1
      if i<N
          OMEGA(i,i-1)=-omega(i,i);
          OMEGA(i,i)=omega(i,i)+omega(i+1,i);
          OMEGA(i,i+1)=-omega(i+1,i);
      else
          OMEGA(i,i-1)=-omega(i,i);
          OMEGA(i,i)=omega(i,i)+omega(i+1,i);
      end;
   end;
end;

% вычисление собственных значений и
% собственных векторов матрицы OMEGA
[Sigma,Teta]=eig(OMEGA);

% вычисление собственных частот
Teta=Teta^0.5;
for i=1:N
      for j=1:N
           SigmaV(j,i)=-Teta(i,i)*Sigma(j,i);
      end;
end;
C1=Sigma^-1*R0'; % решение системы уравнений (9.18)
C2=SigmaV^-1*v0'; % решение системы уравнений (9.19)
C=(C1.^2+C2.^2).^0.5; % вычисление координат вектора C
clear alpha

% вычисление фазы нормальных колебаний
% в соответствии с (9.21), (9.22)
for i=1:N
    if C(i)==0
       alpha(i)=0;
     else
       alpha(i)=atan(C2(i)./C1(i));
       if C1(i)<0
         alpha(i)=pi+alpha(i);
       end;
       if C1(i)>0
```

```
         if C2(i)<0
            alpha(i)=2*pi+alpha(i);
         end;
       end;
    end;
 end;

N=length(OMEGA);
N1=2^13; % число узлов временной сетки
j=1:N1;
Tmax=80; % правая граница
         % временного интервала
t(j)=(j-1)/(N1-1)*Tmax; % координаты узлов
                        % временной сетки

% вычисление значений координат тел
% в узлах временной сетки
for j=1:N1
    s=zeros(N,1);
    for i=1:N
        s=s+C(i)*Sigma(:,i).*cos(Teta(i,i)*t(j)+alpha(i));
    end;
    X(:,j)=s;
end;

% вычисление значений скоростей тел
% в узлах временной сетки
for j=1:N1
    s=zeros(N,1);
    for i=1:N
        s=s+...
          C(i)*Sigma(:,i).*Teta(i,i)*sin(Teta(i,i)*t(j)+...
          alpha(i));
    end;
    Xv(:,j)=-s;
 end;

% визуализация зависимостей
% мгновенных значений смещений и
% скорости от времени
figure(1);
plot(t,X(1,:),'-k',...
     t,X(2,:),'--k',...
     t,X(3,:),':k')
```

```
figure(2);
plot(t,Xv(1,:),'-k',...
     t,Xv(2,:),'--k',...
     t,Xv(3,:),':k')

% построение траекторий
% движения тел на фазовой плоскости
figure(3);
plot(X(1,:),Xv(1,:))

figure(4);
plot(X(2,:),Xv(2,:))

figure(5);
plot(X(3,:),Xv(3,:))

% вычисление спектров
% зависимостей смещений
% тел системы от времени
c1=fft(X(1,:));
c2=fft(X(2,:));
c3=fft(X(3,:));
j=2:N1/2;

% вычисление спектральной плотности
% зависимостей смещений тел
% от времени
Cm1(j-1)=abs(c1(j-1))/(N1/2);
Cm2(j-1)=abs(c2(j-1))/(N1/2);
Cm3(j-1)=abs(c3(j-1))/(N1/2);
Freq(j-1)=(j-1)/Tmax; % вычисление частот
                      % спектральных гармоник

% визуализация спектральных плотностей
% зависимостей смещений тел
% от времени
figure(6);
semilogy(Freq,Cm1,'-k',...
         Freq,10*Cm2,'--k',...
         Freq,500*Cm3,':k')
axis([0 2.5 10^-3 2000])
```

Результаты выполнения приведенной выше последовательности команд представлены на рис. 9.3—9.8.

Рис. 9.3. Зависимости мгновенных значений смещений тел от времени

Рис. 9.4. Зависимости мгновенных значений скорости движения тел от времени

Рис. 9.5. Фазовая траектория первого тела

Рис. 9.6. Фазовая траектория второго тела

Рис. 9.7. Фазовая траектория третьего тела

Рис. 9.8. Спектры функций $x = x(t)$ (масштаб по оси OY — полулогарифмический, для большей наглядности графики сдвинуты друг относительно друга)

Замечания

1. Описанная последовательность команд позволяет проводить анализ движения линейной цепочки с произвольным числом масс. Однако вводить данные при больших значениях N недостаточно удобно. Поэтому для анализа колебаний длинных линейных цепочек нужно сначала создать файлы, содержащие значения масс, коэффициентов жесткости пружин, начальные смещения и координаты вектора начальной скорости, используя, например, программу электронных таблиц Excel. При этом следует расположить исходные данные в одном столбце и сохранить их в файле в формате «Текстовый файл (разделители-пробелы)». В этом случае созданный файл будет иметь расширение txt. Обратите внимание, что файл нужно размещать в папке, имеющей название, состоящее из букв английского алфавита, имя файла также должно состоять из букв английского алфавита. Чтение данных из файлов такого типа осуществляется в командой `dlmread`. Например, если значения масс пружинок сохранены в файле `mass.txt`, находящемся на диске `C` в папке `Data`, считывание данных осуществляется следующей командой:

   ```
   m = dlmread('C:\Data\mass.txt').
   ```

   Обратите внимание, что при создании файла необходимо использовать настройку, при которой для разделения целой и дробных частей действительных чисел используется точка. Для установки соответствующего стандарта ввода действительных чисел необходимо выполнить следующую последовательность команд операционной системы **Windows**: **Пуск** ⇒ **Настройка** ⇒ **Панель управления** ⇒ **Язык и региональные стандарты** ⇒ **Настройка**, затем в поле **Разделитель целой и дробной части установить** «.».

2. Для нахождения собственных чисел и собственных векторов матрицы нами использовалась функция `eig`. При обращении

   ```
   » d = eig(A)
   ```

   функция возвращает вектор-столбец, содержащий собственные значения матрицы `A`. При обращении

   ```
   » [V,D] = eig(A)
   ```

   функция возвращает две матрицы, диагональная матрица `V` содержит собственные значения матрицы `A`, матрица `D` — собственные векторы, записанные в виде векторов-столбцов.

***Задача* 9.1**

1. *Исследуйте свободные колебания системы, состоящей из двух связанных осцилляторов. Вычислите зависимости* $x_0(t)$, $\dot{x}_0(t)$, $x_1(t)$, $\dot{x}_1(t)$ *для начальных условий* $x_0(0) = 0,5$, $\dot{x}_0(0) = 0$, $x_1(0) = 0$, $\dot{x}_1(0) = 0$, *считая, что* $k_0 = k_2 = 1$, $k_1 = 0,8$, $m_0 = m_1 = 1$, $k_0 = k_2 = 1$, $k_1 = 1$, $m_0 = m_1 = 1$ *на временном интервале* $[0; 800]$, *выбрав временную сетку, состоящую из* 2^{14} *точек. Что представляют собой изучаемые движения?*

2. *Проанализируйте удвоенные спектры зависимостей $x_0(t)$, $x_1(t)$. (Здесь множитель 2 учитывает тот факт, что в спектре функции одной частотной компоненте соответствуют две гармоники с положительной и отрицательной частотами (см. Раздел 7.7).) Как соотносятся координаты экстремумов спектров зависимостей $x_0(t)$, $x_1(t)$ с собственными частотами, значения которых возвращаются в вектор $\vec{\Xi}$? Как соотносятся значения экстремумов спектров зависимостей, со значениями коэффициентов, входящих в выражение (9.11)?*

   Указание: *в выбранных в описанном выше документе обозначениях зависимости координат тел колебательной системы имеют вид*

$$x_0(t) = \Theta_{0,0} C_0 \cos(\Xi_0 t + \alpha_0) + \Theta_{0,1} C_1 \cos(\Xi_1 t + \alpha_1),$$

$$x_1(t) = \Theta_{1,0} C_1 \cos(\Xi_0 t + \alpha_0) + \Theta_{1,1} C_1 \cos(\Xi_1 t + \alpha_1).$$

3. *В чем причина различий между значениями коэффициентов, входящих в выражения для $x_0(t)$, $x_1(t)$, и значениями экстремумов соответствующих спектров?*

4. *Как меняется точность спектральной оценки коэффициентов, входящих в выражения для $x_0(t)$, $x_1(t)$ при изменении шага временной сетки?*

5. *Как меняется точность спектральной оценки коэффициентов, входящих в выражения для $x_0(t)$, $x_1(t)$ при длительности сигнала?*

***Задача* 9.2**

Решите задачу 9.1, считая, что $k_0 = k_1 = 1$, $k_1 = 0,2$, $m_0 = m_1 = 1$. Как меняются частоты каждого колебания при $k_1 = 0,1$?

***Задача* 9.3**

1. *Решите задачи 9.1, 9.2 для начальных условий $x_0(0) = 0,5$, $\dot{x}_0(0) = 0$, $x_1(0) = 0$, $\dot{x}_1(0) = 0$.*

2. *Получите выражения для кинетической и потенциальной энергий каждой частицы.*

   (a) *Пользуясь законом сохранения энергии, справедливого для всей системы, проверьте полученные выражения.*

   (b) *Вычислите и изобразите энергию каждой частицы как функцию времени, графически и качественно опишите ее поведение.*

3. *Исследуйте зависимость частот колебаний от жесткости пружин $k_0 = k_1$, k_2. Объясните полученные результаты.*

***Задача* 9.4**

1. *Решите задачи 9.1, 9.2 для начальных условий $x_0(0) = 0,5$, $\dot{x}_0(0) = 0$, $x_1(0) = -0,5$, $\dot{x}_1(0) = 0$.*

2. *Вычислите и изобразите графически энергию каждой частицы как функцию времени и качественно опишите ее поведение.*

3. *Проведите исследование и определите зависимость частот колебаний от жесткости пружин $k_0 = k_2, k_1$. Объясните полученные результаты. Сравните полученное значение частоты с соответствующим значением, найденным в предыдущей задаче.*

***Задача* 9.5**

Решите задачи 9.1–9.2, считая, что $k_0 = k_2 = 1$, $k_1 = 0,2$, $m_0 = 1$, $m_2 = 2$. Как меняются частоты каждого колебания при k_1? Исследуйте зависимость частот собственных колебаний от массы тел m_1, m_2 колебательной системы.

***Задача* 9.6**

1. *Дополните описанный выше документ модулем, позволяющим решать обратную задачу: по известным значениям собственных частот и соответствующим им собственным векторам определять начальные условия, при которых возбуждается только одна мода колебаний.*

2. *Проверьте правильность работы модуля, задав соответствующие начальные условия, вычислив зависимости $x_0(t)$, $x_1(t)$ и проанализировав спектр данных зависимостей.*

***Задача* 9.7**

Как известно [4], при сложении двух колебаний $x_1(t) = A\cos(\omega_1 t)$, $x_2(t) = A\cos(\omega_2 t)$ с равными амплитудами и близкими частотами ω_1, ω_2, отвечающих следующему условию: $\omega_2 = \omega_1 + \Delta\omega$, где $\Delta\omega \ll \omega_1, \omega_2$, в рассматриваемой системе всегда возникают биения, то есть зависимость координаты от времени описывается зависимостью

$$x\left(t\right) = \left[2A\cos\left(\omega_{mod}t\right)\right]\cos\left(\omega_{midl}t\right),$$

где

$$\omega_{midl} = (\omega_1 + \omega_2)/2,$$

$$\omega_{mod} = |\omega_1 - \omega_2|/2.$$

1. *Используя результаты задачи 9.3, выберите значения коэффициентов жесткости пружин, обеспечивающих возникновение в системе биений. Используя результаты задачи 9.6, определите начальные условия, при которых в системе связанных осцилляторов возникают биения. Для найденных начальных условий вычислите зависимости $x_0(t)$, $\dot{x}_0(t)$, $x_1(t)$, $\dot{x}_1(t)$.*

2. *Определите значения ω_{midl}, ω_{mod}, задайте функцию $x(t)$, вычислите мгновенные значения смещения, описываемого данной функцией, и отобразите их на графике. Сравните полученные результаты с результатами расчетов по формуле (9.10).*

Таким образом, в результате решения задач 9.1—9.6 можно сделать следующие выводы:

1. Движение колебательной системы, состоящей из двух связанных осцилляторов, всегда представляет сумму двух движений, имеющих различные частоты (нормальные колебания).

2. Существуют начальные условия, при которых в системе возбуждается только одно нормальное колебание.

3. При $k_0 = k_2 \neq k_1$ нижняя частота нормальных колебаний вычисляется по формуле
$$\omega_1 = \sqrt{k_0/m},$$
верхняя частота нормальных колебаний вычисляется по формуле
$$\omega_2 = \sqrt{(k_0 + 2k_1)/m}.$$

4. При возбуждении нормальных колебаний с частотой ω_1 частицы колеблются в фазе.

5. При возбуждении нормальных колебаний с частотой ω_2 частицы колеблются в противофазе.

6. При определенных сочетаниях значений масс, коэффициентов жесткости пружин и начальных условий в колебательной системе возникают биения.

***Задача* 9.8**

Рассмотрите движение колебательной системы, состоящей из трех связанных осцилляторов, считая, что $k_0 = k_4 = 1$, $k_1 = k_2 = k_3 = 0,2$, $m_i = 1$, $i = 0, 1, 2$ *с ненулевыми начальными условиями. Исследуйте зависимость смещения частиц от времени.*

***Задача* 9.9**

Решите задачу 9.9. с начальными условиями, приведенными в таблице 9.1, полагая начальные скорости всех тел системы равными нулю.

Таблица 9.1. Начальные условия в задачах 9.9, 9.10

	$x_0(0)$	$x_1(0)$	$x_2(0)$
Случай 1	0,5	0,5	0,5
Случай 2	0,5	-0,5	0,5
Случай 3	0,5	0	-0,5

***Задача* 9.10**

Рассмотрите движение колебательной системы, состоящей из N *связанных осцилляторов* ($N \gg 10$), *считая, что* $m_i = 1$, $i = 0, 1, \ldots, N-1$, $k_j = 1$, $j = 0, 1, \ldots, N$ *для начальных условий, представленных в таблице 9.1.*

1. *Сравните численные значения собственных частот ω_s с известным аналитическим результатом*

$$\omega_s = 2\sqrt{\frac{k}{m}}\sin\left(\frac{\pi\, s}{2(N+1)}\right),\ s = 1, 2, \ldots, N,$$

полученным в [*2, задача № 7.1*], *где*

$$K_s = \frac{\pi\, s}{2(N+1)}$$

называют волновым вектором s*-й моды колебаний.*

2. *Постройте зависимость $\omega_s = \omega_s(K)$.*

3. *Величину $v_f = \omega_s/K_s$ называют фазовой скоростью колебаний. Постройте график зависимости $v_f = v_f(K_s)$ и убедитесь в том, что при $s \approx 1$ (длинноволновой предел) фазовая скорость не зависит от волнового вектора.*

***Задача* 9.11**

Рассмотрите движение колебательной системы, состоящей из N связанных осцилляторов ($N \geq 10$), считая, что $m_i = 1$, $i = 0, 1, \ldots, N-1$, $k_j = 0, 1, \ldots, N$. Задайте начальные условия

$$x_0(0) = 0,5,$$

$$x_i(0) = 0,\ i = 1, 2, \ldots, N-1,$$

$$\dot{x}_i(0) = 0,\ i = 0, 1, \ldots, N.$$

1. *Вычислите мгновенные значения функций $x_i(t)$, $i = 0, 1, \ldots, N-1$.*

2. *Используя мгновенные значения функций $x_i(t)$, $i = 0, 1, \ldots, N-1$, определите скорость распространения возмущений в линейной цепочке.*

   Указание: *для нахождения скорости распространения возмущения следует определить временные интервалы t_n ($n = 1, 2, \ldots, N-1$), в течение которых смещение n-го тела превысит заданное значение (например, 10^{-2}), затем по зависимости $n = f(t_n)$, используя линейную регрессию*[1] *, оценить угловой коэффициент линии регрессии.*

[1] В MATLAB для нахождения наклона линии регрессии в смысле наименьших квадратов необходимо выполнить следующую последовательность команд:

```
% задание исходных данных
» t=[0 1 2 3 4]';
» y=[0.1 0.9 2.1 2.9 4.15]';
% решение системы уравнений метода наименьших квадратов
» X=[ones(size(t)) t];
» a=X\y;
» T=0:0.01:4;T=T'; % задание временной сетки для вычисления значений
                   % аппроксимирующей функции
» Y=[ones(size(T)) T]*a; % вычисление значений аппроксимирующей функции
» plot(t,y,'o',T,Y,'-'); % визуализация исходных данных и аппроксимирующей функции
```

3. *Для того чтобы убедиться в независимости вашей оценки скорости распространения возмущений от N, повторите расчеты для бóльших значений N.*

4. *Проведите исследование и определите зависимость скорости распространения возмущения от массы тел, образующих колебательную систему, и жесткости пружин.*

5. *Рассмотрите «неупорядоченную» систему при помощи приписывания телам колебательной системы различных масс (например, с помощью генератора случайных чисел). Как влияет беспорядок на скорость распространения возмущения?*

***Задача* 9.12**

Используя описанный выше документ, решите задачу 7.4 из задачника [2].

9.3. МОДЕЛИРОВАНИЕ ВЫНУЖДЕННЫХ КОЛЕБАНИЙ ЦЕПОЧКИ СВЯЗАННЫХ ГАРМОНИЧЕСКИХ ОСЦИЛЛЯТОРОВ

Для описания движения цепочки связанных гармонических осцилляторов под действием вынуждающей силы следует добавить в правую часть системы уравнений движения (9.2) соответствующие выражения для внешней силы. Дальнейшее рассмотрение ограничим случаем, когда сила приложена к точке A колебательной системы (рис. 9.1), что для достаточно длинных цепочек не приводит к потере общности.

Система дифференциальных уравнений в данном случае принимает следующий вид:

$$\begin{aligned}
&\ddot{x}_0 = -\omega_{00}^2 x_0 - \omega_{10}^2 (x_0 - x_1) + \frac{F(t)}{m_0}, \\
&\ddot{x}_i = -\omega_{ii}^2 (x_i - x_{i-1}) - \omega_{i+1i}^2 (x_i - x_{i+1}), \; i = 1, \ldots, N-2, \\
&\ddot{x}_{N-1} = -\omega_{N-1N-1}^2 (x_{N-1} - x_{N-2}) - \omega_{NN-1}^2 x_{N-1}.
\end{aligned} \tag{9.23}$$

Будем искать численное решение системы (9.23). Напомним, что ранее нам уже приходилось решать системы дифференциальных уравнений второго порядка (см. главы 4, 6). Для этого нами использовался следующий алгоритм:

1) задать вектор-функцию, возвращающую значения первых производных системы ДУ (размерность функции $2N$);

2) задать вектор, содержащий начальные условия:

$$\{x_i(0)\},$$
$$\{\dot{x}_i(0)\},$$

где $i = 1, 2, \ldots, N-1$.

3) обратиться к одной из функций, возвращающих таблицу, содержащую численное решение системы ДУ, например функции `ode45`;

4) провести визуализацию полученных численных решений.

Описание функции, возвращающей значения первых производных системы ДУ (9.23), мы разместили в файле `Euler2.m`, листинг которого приводится ниже.

```
% листинг файла Euler2.m
function z=euler2(t,x)
% описание функции, возвращающей
% значения первых производных в (9.23)

global k m A Omega

N=length(m); % число тел колебательной системы
z=zeros(2*N,1);
z(1)=x(2);
z(2)=-(k(1)/m(1))*x(1)-...
     (k(2)/m(1))*(x(1)-x(3))+...
      F(t,A,Omega);
K=3;
for i=2:N-1
   z(K)=x(K+1);
   z(K+1)=-(k(i)/m(i))*(x(K)-x(K-2))-...
           (k(i+1)/m(i))*(x(K)-x(K+2));
   K=K+2;
end;
z(2*N-1)=x(2*N);
z(2*N)=-(k(N)/m(N))*(x(2*N-1)-x(2*N-3))-...
         k(N+1)/m(N)*x(2*N-1);

function f=F(t,A,Omega)
% функция, описывающая
% внешнюю вынуждающую силу
f=A*sin(Omega*t);
```

Далее для нахождения и визуализации численного решения системы ДУ (9.23), описывающих систему, совершающую свободные колебания, необходимо выполнить следующую последовательность команд, сохраненную нами в файле `Glava9_2.m`:

```
% листинг файла Glava9_2.m
clear all;
global k m A Omega

% задание параметров колебательной системы
N=16; % число тел колебательной системы
n=1:N;
m(n)=1; % задание масс тел
i=1:N+1;
```

```
k(i)=1; % задание жесткостей пружин колебательной системы
n=1:2*N;

% начальные условия
>> R0(n)=0;
>> R0(1)=0.5
>> A=0; % амплитуда внешней
        % вынуждающей силы
>> Omega=0.0; % частота внешней
              % вынуждающей силы
>> Tfin=50*pi; % правая граница
               % временной сетки

>> Np=2^13-1; % число узлов временной сетки
>> [T,M]=ode45('euler2',[0:Tfin/Np:Tfin],R0); % вычисление
                                              % численного
                                              % решения (9.23)

% визуализация решений ДУ (9.23)
subplot(4,4,1);
plot(T,M(:,1));
axis([0 40 -0.5 0.5]);
title('n=1');

subplot(4,4,5);
plot(T,M(:,3));
axis([0 40 -0.5 0.5]);
title('n=2');

subplot(4,4,9);
plot(T,M(:,5));
axis([0 40 -0.5 0.5]);
title('n=3');

subplot(4,4,13);
plot(T,M(:,7));
axis([0 40 -0.5 0.5]);
title('n=4');

subplot(4,4,2);
plot(T,M(:,9));
axis([0 40 -0.5 0.5]);
title('n=5');

subplot(4,4,6);
```

```
plot(T,M(:,11));
axis([0 40 -0.5 0.5]);
title('n=6');

subplot(4,4,10);
plot(T,M(:,13));
axis([0 40 -0.5 0.5]);
title('n=7');

subplot(4,4,14);
plot(T,M(:,15));
axis([0 40 -0.5 0.5]);
title('n=8');

subplot(4,4,3);
plot(T,M(:,17));
axis([0 40 -0.5 0.5]);
title('n=9');

subplot(4,4,7);
plot(T,M(:,19));
axis([0 40 -0.5 0.5]);
title('n=10');

subplot(4,4,11);
plot(T,M(:,21));
axis([0 40 -0.5 0.5]);
title('n=11');

subplot(4,4,15);
plot(T,M(:,23));
axis([0 40 -0.5 0.5]);
title('n=12');

subplot(4,4,4);
plot(T,M(:,25));
axis([0 40 -0.5 0.5]);
title('n=13');

subplot(4,4,8);
plot(T,M(:,27));
axis([0 40 -0.5 0.5]);
title('n=14');

subplot(4,4,12);
```

```
plot(T,M(:,29));
axis([0 40 -0.5 0.5]);
title('n=15');

subplot(4,4,16);
plot(T,M(:,31));
axis([0 40 -0.5 0.5]);
title('n=16');
```

Результат выполнения приведенной выше последовательности команд представлен на рис. 9.9.

Рис. 9.9. Зависимость мгновенных значений смещения тел колебательной системы от времени

Одной из основных проблем численного решения ДУ и систем ДУ является проблема выбора шага интегрирования Δt, поскольку при достаточно большом шаге интегрирования возникают неустойчивые решения, то есть решения, погрешность которых начинает возрастать во времени экспоненциально быстро. Один из способов проверки устойчивости метода заключается в контроле величины полной энергии, которая в случае свободных колебаний должна сохраняться, поэтому для проверки пра-

вильности выбора шага интегрирования Δt можно использовать следующий алгоритм:

1. Задать начальные смещения и скорости тел цепочки связанных осцилляторов.
2. Задать временной интервал, на котором ищется решение системы ДУ.
3. Задать число точек, в которых ищется численное решение системы ДУ.
4. Найти решение системы ДУ.
5. Вычислить значения энергии системы связанных осцилляторов в каждый момент времени.
6. Проанализировать изменение энергии системы во времени на заданном временном интервале и оценить точность выполнения закона сохранения энергии.
7. При неудовлетворительной точности решения повторить п. 3—6.
8. При удовлетворительной точности решения перейти к анализу вынужденных колебаний.

Как очевидно, для реализации описанного выше алгоритма необходимо уметь вычислять энергии каждого из тел системы в заданные моменты времени. Для решения данной задачи можно использовать функцию `En`, описание которой мы сохранили в файле `En.m`.

```
% листинг файла En.m
function E=En(N,m,k,M)
K=length(M); % число строк
             % матрицы решений
K1=1;
for j=1:N
  for i=1:K
    if j>1
      if j<N
        e(i)=0.5*m(j)*M(i,2*j).^2+...
             0.25*k(j)*(M(i,2*j-1)-...
             M(i,2*j-3)).^2+...
             0.25*k(j+1)*(M(i,2*j-1)-M(i,2*j+1)).^2;
      end;
    end;
    if j==1
      e(i)=0.5*m(1)*M(i,2)^2+...
           0.5*k(1)*M(i,1)^2+...
           0.25*k(2)*(M(i,1)-M(i,3))^2;
    end;
    if j==N
      e(i)=0.5*m(N)*M(i,2*N)^2+...
```

```
            0.5*k(N+1)*M(i,2*N-1)^2+...
            0.25*k(N)*(M(i,2*N-1)-M(i,2*N-3))^2;
    end;
  end;
  en(:,K1)=e';
  K1=K1+1;
end;
E=en;
```

Для вычисления мгновенных значений энергии тел колебательной системы и ее полной энергии необходимо выполнить следующую последовательность команд, сохраненную нами в файле `Glava9_3.m`:

```
% Листинг файла Glava9_3.m
clear all

global k m A Omega

N=16; % число тел колебательной системы
n=1:N;
m(n)=1; % создание вектора, содержащего
        % значения масс тел
        % колебательной системы
i=1:N+1;
k(i)=1; % создание вектора,
        % содержащего
        % жесткости пружин
        % колебательной системы

% задание начальных условий
n=1:2*N;
R0(n)=0;
R0(1)=0.5;

% задание параметров
% внешней вынуждающей силы
A=0.0; % амплитуда
Omega=0.0;

% решение системы ДУ (9.23)
Tfin=50*pi;
Np=2^13-1;
[T,M]=ode45('euler2',[0:Tfin/Np:Tfin],R0);

% вычисление мгновенных значений
% энергии тел колебательной системы
```

```
E=En(N,m,k,M);

% визуализация зависимости
% мгновенных значений энергии от времени
% 1-го, 4-го, 8-10-го, 12-го тел
% колебательной системы

figure(1);
plot(T,E(:,1),T,E(:,4),...
     T,E(:,8),T,E(:,12))
axis([0 30 0 0.25])

% вычисление и визуализация
% мгновенных значений
% полной энергии системы  от времени
for i=1:Np+1
    s=E(i,:);
    Efull(i)=sum(s);
end;

figure(2);
plot(T,Efull)
```

Результаты выполнения приведенной выше последовательности команд представлены на рис. 9.10, 9.11.

Рис. 9.10. Зависимость значений энергий выбранных тел колебательной системы от времени

Рис. 9.11. Зависимость мгновенных значений полной энергии системы от времени

Анализ зависимости мгновенной энергии колебательной системы от времени, представленной на рис. 9.11, показывает, что полная энергия отклоняется от своего первоначального значения по линейному закону. При этом максимальная величина от-

клонения, характеризующая точность численного решения, на выбранном временном интервале не превосходит $0,24\%$.

***Задача* 9.13**

1. *Используя файл* `Glava9_1.m`*, описанный в разделе 9.2, найдите точные значения зависимостей координат тел системы связанных осцилляторов от времени, совершающей свободные колебания.*

2. *Сравните зависимости координат тел системы связанных осцилляторов от времени, найденные численным решением системы ДУ (9.2), с точными значениями.*

3. *Оцените точность численного решения.*

Для анализа движения системы связанных осцилляторов под действием вынуждающей силы следует в приведенной выше последовательности команд задать отличными от нуля значения переменных `A` и `Omega`.

***Задача* 9.14**

Рассмотрите движение колебательной системы, состоящей из N связанных осцилляторов ($N \geq 10$), считая, что $m_i = 1$, $i = 0, 1, \ldots, N-1$, $k_j = 1$, $j = 0, 1, \ldots, N$, при начальных условиях $x_i(0) = 0$, $\dot{x}_i(0) = 0$ под действием внешней силы $f(t) = A\sin(\Omega t)$, приложенной в точке A.

1. Н*айдите частоты собственных колебаний ω_s, $s = 0, 1, \ldots, N-1$ рассматриваемой системы связанных гармонических осцилляторов.*

2. *Исследуйте движение каждого из тел системы связанных осцилляторов при $\Omega = \omega_s$, $s = 0, 1, \ldots, N-1$, используя для этого зависимости $x_i(t)$, $v_i(t)$, $E_i(t)$, $i = 0, 1, \ldots, N-1$ и их спектры.*

3. *Исследуйте движение каждого из тел системы связанных осцилляторов при условии:*
$$\Omega \in [\min(\omega_s); \max(\omega_s)],$$
$$\Omega \neq \omega_s, s = 0, 1, \ldots, N-1.$$

4. *Опишите движение колебательной системы при*
$$\Omega \ll \min(\omega_s)$$
и
$$\Omega \gg \max(\omega_s).$$
Как зависит амплитуда колебаний от номера частицы при $\Omega \gg \max(\omega_s)$?

5. *Опишите движение колебательной системы при $\Omega < \min(\omega_s)$ и $\Omega > \max(\omega_s)$.*

6. *Исследуйте зависимость скорости распространения возмущений в линейной цепочке от частоты возбуждающей силы. Объясните полученный результат.*

7. *Исследуйте зависимость скорости распространения возмущений в линейной цепочке от массы тел системы связанных осцилляторов.*

8. *Исследуйте зависимость скорости распространения возмущения в линейной системе от жесткости пружин.*

В результате решения задачи 9.15 можно сделать следующие выводы:

1. В линейной системе связанных гармонических осцилляторов происходит передача энергии вдоль цепочки от левого конца цепочки к правому.

2. Рассматриваемая система обладает фильтрующими свойствами, не пропуская колебания с частотами $\Omega \ll \min(\omega_s)$ и $\Omega \gg \min(\omega_s)$.

3. При воздействии на систему внешней вынуждающей силы с частотой Ω, отвечающей условию $\Omega \ll \min(\omega_s)$, система совершает колебания как единое целое с частотой, равной частоте внешней вынуждающей силы.

4. При воздействии на систему внешней вынуждающей силы с частотой Ω, отвечающей условию $\Omega \gg \min(\omega_s)$, амплитуды колебаний частиц убывают к правому концу цепочки.

***Задача* 9.15**

Решите задачу 9.15, считая, что

$$m_i = 1,\ i = 0, 2, \ldots, N-1,$$

$$m_i = 2,\ i = 1, 3, \ldots, N,$$

$$k_j = 1,\ j = 0, 1, \ldots, N.$$

***Задача* 9.16**

Решите задачу 9.15, считая, что

$$m_i = 1,\ i = 0, 1, \ldots, N-1,$$

$$k_j = 1,\ j = 0, 2, \ldots, N,$$

$$k_j = 2,\ j = 1, 3, \ldots, N-1.$$

***Задача* 9.17**

Решите задачу 9.15, считая, что система состоит из осцилляторов различных масс (задавая массы, например, с помощью генератора случайных чисел).

Как влияет беспорядок на поведение системы и скорость распространения возмущения?

9.4. МОДЕЛИРОВАНИЕ ВОЛНОВЫХ ДВИЖЕНИЙ

В предыдущем разделе мы обнаружили, что в системе из N связанных осцилляторов колебания отдельных тел приводят к распространению энергии вдоль цепочки на произвольные расстояния. Данный факт указывает на аналогию между колебанием цепочки связанных осцилляторов и волнами, распространяющимися в непрерывных средах. Для того чтобы осуществить переход от колебаний дискретной цепочки осцилляторов массой m, связанных пружинками одинаковой жесткости k, длина которых в равновесном состоянии равна a, к волновому движению непрерывной среды рассмотрим уравнение движения i-го тела:

$$\frac{d^2u_i}{dt^2} = -\frac{k}{m}\left(2u_i - u_{i+1} - u_{i-1}\right),\ i = 2, 3, \ldots, N-2, \tag{9.24}$$

где u_i — смещение i-го тела относительно положения равновесия.

При $N \to \infty$, $a \to 0$ и постоянной длине цепочки можно заменить в (9.24) $u_i(t)$, где i — дискретная переменная на функцию $u(x,t)$, x — непрерывная переменная. Данная замена позволяет записать (9.24) в следующем виде:

$$\frac{\partial^2 u\,(x,t)}{\partial\, t^2} = \frac{ka^2}{m}\frac{1}{a^2}\left[u\,(x+a,t) - 2u\,(x,t) + u\,(x-a,t)\right]. \tag{9.25}$$

Разложив функцию $u(x,t)$ в ряд Тэйлора в точках $x \pm a$:

$$u\,(x \pm a, t) = u(x,t) \pm a\frac{\partial u}{\partial x} + \frac{a^2}{2}\frac{\partial^2 u}{\partial x^2} + \ldots$$

и подставив в (9.25), получим

$$\frac{\partial^{2\cdot} u\,(x,t)}{\partial\, t^2} = \frac{ka^2}{m}\frac{\partial^2 u}{\partial\, x^2}. \tag{9.26}$$

Вводя обозначения $\mu = m/a$, $T = k/a$ (T — натяжение, m — линейная плотность массы), $v^2 = T/\mu$, запишем (9.26) в следующем виде:

$$\frac{\partial^2 u\,(x,t)}{\partial\, t^2} = \frac{1}{v^2}\frac{\partial^2 u}{\partial\, x^2}. \tag{9.27}$$

Уравнение (9.27) называется волновым уравнением. Непосредственной подстановкой в (9.27) легко убедиться в том, что любые функции вида $f(x - vt)$, $f(x + vt)$ являются его решениями. При этом на прямых, определяемых уравнениями $x \pm vt = \mathrm{const}$, решения уравнения остаются постоянными. Так как волновое уравнение (9.27) является линейным, решения волнового уравнения удовлетворяют принципу суперпозиции, то есть любая функция вида

$$\psi\,(x,t) = \sum_{i=1}^{N}\left[f_i\,(x - vt) + f_i\,(x + vt)\right] \tag{9.28}$$

также является решением волнового уравнения (9.27).

Следовательно, поведение волны произвольной формы можно описать, представляя ее в виде набора синусоидальных волн (ряда Фурье). Данный подход мы используем в дальнейшем для анализа движения волнового пакета в среде с дисперсией.

Анализ особенностей решений волнового уравнения, следуя выбранному в нашей книге подходу, будем проводить численно. Для описания эволюции решений волнового уравнения в MATLAB в качестве первого шага необходимо создать три файла:

1) файл `Wave.m`, содержащий функцию, описывающую решение волнового уравнения в момент времени $t = 0$;
2) файл `WaveP.m`, содержащий функцию, описывающую волну, распространяющуюся в положительном направлении оси OX;
3) файл `WaveN.m`, содержащий функцию, описывающую волну, распространяющуюся в отрицательном направлении оси OX.

Ниже приведены листинги каждого из упомянутых выше файлов.

```
% листинг файла Wave.m
function z=Wave(A,x)
% функция, возвращающая
% решение волнового уравнения (9.27)
% в момент времени t = 0

z=A*exp(-x.^2/4);

% листинг файла WaveP.m
function z=WaveP(A,x,v,t)
% функция, описывающая волну,
% распространяющуюся в
% положительном направлении оси OX

z=Wave(A,x-v*t);

% листинг файла WaveN.m
function z=WaveN(A,x,v,t)
% функция, описывающая волну,
% распространяющуюся в
% отрицательном направлении оси OX

z=Wave(A,x+v*t);
```

Далее для вычисления значений соответствующих функций и визуализации решений волнового уравнения необходимо выполнить следующую последовательность команд, сохраненную нами в файле `Glava9_4.m`:

```
% листинг файла Glava9_4.m:
clear all;

% задание пространственной сетки
```

```
xmin=-2*pi;
xmax=2*pi;
Nx=101;
i=1:Nx;
x(i)=xmin+(xmax-xmin)/(Nx-1)*(i-1);

% задание временной сетки
tmin=0;
tmax=50;
v=0.05;
Nt=101;
j=1:Nt;
t(j)=tmin+(tmax-tmin)/(Nt-1)*(j-1);

% вычисление значений функций,
% описывающих решение
% волнового уравнения
for i=1:Nt
    for j=1:Nx
        M1(i,j)=WaveP(1,x(j),v,t(i));
        M2(i,j)=WaveN(1,x(j),v,t(i));
     end;
end;

% визуализация решений волнового уравнения
figure(1);
plot(x,M1(1,:),...
     x,M1(40,:),'--',...
     x,M1(80,:),':');

figure(2);
plot(x,M2(1,:),...
     x,M2(40,:),'--',...
     x,M2(80,:),':');

[X,Y]=meshgrid(x,t); % задание двумерных матриц,
                     % используемых для
                     % визуализации функций,
                     % зависящих от
                     % двух переменных

figure(3);
surf(X,Y,M1);
colormap gray
```

```
figure(4);
surf(X,Y,M2);
colormap gray

figure(5);
contourf(M1,15);
colormap gray

figure(6);
contourf(M2,15);
colormap gray
```

Результаты выполнения приведенной выше последовательности команд представлены на рис. 9.12–9.17.

Рис. 9.12. Функция $u_p(x,t)$ в фиксированные моменты времени $t = 0, 40, 80$

Рис. 9.13. Функция $u_n(x,t)$ в фиксированные моменты времени $t = 0, 40, 80$

Рис. 9.14. Поверхность, описываемая функцией $u_p(x,t)$

Рис. 9.15. Поверхность, описываемая функцией $u_n(x,t)$

Рис. 9.16. Карта линий уровня функции $u_p(x,t)$

Рис. 9.17. Карта линий уровня функции $u_n(x,t)$

Как видно из рис. 9.12–9.17, функции $u_p(x,t)$, $u_n(x,t)$ описывают волны, распространяющиеся соответственно в положительном и отрицательных направлениях оси OX. (Такие волны называются бегущими.) Отметим, что форма данных волн во времени остается неизменной.

***Задача* 9.18**

Замените в описанном выше документе функцию

$$\exp\left(-x^2/2\right)$$

функцией

$$\cos\left(\frac{2\pi}{\lambda}\cdot x\right).$$

1. *Проанализируйте поведение решений волнового уравнения для различных значений λ.*

2. *Рассмотрите решение волнового уравнения при $t=0$ — функцию $u(x,0)$. Чему равен период функции $u(x,0)$, называемый длиной волны?*

3. *Рассмотрите решение волнового уравнения при $x=0$ — функцию $u(0,t)$. Чему равен период функции $u(0,t)$, называемый периодом волны?*

4. *Убедитесь в том, что отношение периода функции $u(x,0)$ к периоду функции $u(0,t)$ равно скорости распространения волны v. Получите этот же результат аналитически.*

***Задача* 9.19**

Проанализируйте особенности распространения волны, описываемой следующей функцией:

$$u(x,t)=\frac{4}{\pi}\left[1+\sin(x-vt)+\frac{1}{3}\sin 3(x-vt)+\frac{1}{5}\sin 5(x-vt))\right]. \qquad (9.29)$$

Отметим, что наряду с длиной волны λ и периодом волны T в физике также часто используют следующие величины:

1) $k = 2\pi/\lambda$, *называемую волновым вектором*;

2) $f = 1/T$, *называемую частотой волны;*

3) $\omega = 2\pi T = 2\pi/f$, *называемую циклической частотой.*

В данных обозначениях функция, описывающая гармоническую волну, записывается в виде

$$u\,(x,t) = A\cos(kx - \omega\,t). \tag{9.30}$$

Используя (9.30), определите, какие волны (волновые числа и частоты волн) представлены в формуле (9.29).

***Задача* 9.20**

Проанализируйте особенности распространения волны

$$u\,(x,t) = \sin(x - vt) + \sin(x + vt).$$

Опишите получающуюся в результате волну.

***Задача* 9.21**

Рассмотрите случай двух гармонических волн одинаковой амплитуды, распространяющихся с одинаковой скоростью $v = 1$ в положительном направлении оси OX и имеющих разные частоты $\omega_1 = 1$ и $\omega_2 = 9,05$.

1. *Опишите форму $u(x,t)$ для различных времен t.*
2. *Оцените скорость распространения мелкомасштабных горбов амплитуды — фазовую скорость.*
3. *Оцените скорость распространения низкочастотной огибающей — групповую скорость.*
4. *Сравните значения фазовой и групповой скоростей.*

9.5. ФУРЬЕ-АНАЛИЗ ВОЛНОВЫХ ПАКЕТОВ, ДВИЖУЩИХСЯ В СРЕДЕ С ДИСПЕРСИЕЙ

Анализ выражения (9.30) показывает, что можно ввести фазовую функцию косинусоидальной бегущей волны, распространяющейся в положительном направлении оси OX, как аргумент волновой функции $\cos(\omega t - kx)$:

$$\varphi\,(x,t) = \omega\,t - kx. \tag{9.31}$$

Если мы хотим следить за каким-либо гребнем волны (максимумом функции $\cos\varphi(x,t)$) или за ее впадиной (минимум функции $\cos\varphi(x,t)$), по мере увеличения времени переходить к все большим значениям x так, чтобы фаза $\varphi(x,t)$ была постоянной.

Условие постоянства фазы с математической точки зрения означает, что полный дифференциал функции $\varphi(x,t)$, имеющий вид

$$d\varphi = \left(\frac{\partial\varphi}{\partial\, t}\right) dt + \left(\frac{\partial\varphi}{\partial\, x}\right) dx = \omega\, dt - k dx, \tag{9.32}$$

равен нулю.

Приравнивая (9.32) нулю, находим условие постоянства фазы:

$$\frac{dx}{dt} = v_f = \frac{\omega}{k}, \tag{9.33}$$

где v_f — фазовая скорость волны.

Выражение (9.33) дает связь между фазовой скоростью волны, частотой волны и волновым вектором. Вообще говоря, условия распространения волны определяются свойствами среды. При этом ω, а следовательно, и фазовая скорость v_f могут зависеть от волнового вектора k. Волны, у которых фазовая скорость зависит от волнового вектора, называются диспергирующими. В противном случае волны называются недиспергирующими. Диспергирующая волна, представляющая собой суперпозицию бегущих волн с различными волновыми числами, будет менять свою форму по мере распространения в пространстве, так как составляющие с различными длинами волн распространяются с различными скоростями.

Продемонстрируем влияние дисперсии на распространение волн на примере задачи о движении волнового пакета в среде с дисперсией, под которым мы будем понимать некоторую синусоидальную волну, имеющую конечную протяженность в пространстве или во времени. Решение данной задачи основано на представлении волнового пакета в виде суперпозиции гармонических функций (метод Фурье) [6].

Рассмотрим задачу об описании движения волнового пакета, локализованного в момент времени $t = 0$ в пространстве. Данное условие означает, что огибающая пакета является функцией, достаточно быстро стремящейся к нулю при $x \to \pm\infty$. Введем обозначение

$$f(x) = u(x, 0). \tag{9.34}$$

Условие «быстрого стремления функции к нулю» обеспечивает возможность ее разложения в ряд Фурье. Если функция $f(x)$ не периодическая по координате, то суперпозиция составляющих ее функций непрерывна по переменной k и выражается через интеграл Фурье:

$$f\,(x) = \int\limits_{-\infty}^{\infty} A\,(k)\, e^{ikx} dk, \tag{9.35}$$

где

$$A\,(k) = \int\limits_{-\infty}^{\infty} f(x) e^{-ikx} dx. \tag{9.36}$$

Каждая гармоническая составляющая в (9.35) определяет собственную гармоническую волну с частотой $\omega = \omega(k)$, то есть каждая частотная составляющая бегущей

волны распространяется со своей собственной фазовой скоростью

$$v_f = \frac{\omega\,(k)}{k}. \tag{9.37}$$

Искомая функция $u(x,t)$, описывающая бегущую волну, является суперпозицией гармонических бегущих волн, образующих начальное возмущение. Это означает, что найти $u(x,t)$ можно заменой kx на $kx - \omega(k)t$ в каждой гармонической составляющей суперпозиции (9.35):

$$u(x,t) = \int\limits_{-\infty}^{\infty} A\,(k)\, e^{i(kx-\omega(k)t)} dt. \tag{9.38}$$

Таким образом, алгоритм аналитического решения задачи достаточно прост:

1. Задать функцию $f(x)$, описывающую начальное возмущение при $t = 0$.
2. Задать функцию $\omega = \omega(k)$.
3. Определить функцию $A(k)$, вычислив интеграл (9.36).
4. Определить функцию $u(x,t)$, вычислив интеграл (9.38).

Однако для того чтобы использовать данный алгоритм при нахождении численных значений функции, требуется провести его доработку, связанную прежде всего с тем, что вычисление интегралов в (9.35), (9.36), (9.38) приходится проводить на интервалах конечной длительности, то есть проводить разложение функции не в интеграл, а в ряд Фурье.

В данном случае, как было показано в главе 7, вместо (9.35), (9.36), (9.38) следует использовать следующие выражения:

$$f\,(x) = \sum_{n=0}^{N-1} A\,(n)\;\, e^{i\,k\,n\,x}, \tag{9.39}$$

где $k = \dfrac{2\pi}{x_{\max} - x_{\min}}$,

$$A\,(n) = \frac{1}{x_{\max} - x_{\min}} \int\limits_{x_{\min}}^{x_{\max}} f\,(x)\;\, e^{-i\,k\,n\,x} dx, \tag{9.40}$$

где $n = 0, 1, \ldots, N-1$,

$$u\,(x,t) = \sum_{n=0}^{N} A(n) e^{i\,(k\,n(x-x_{\min})-\omega(kn))}, \tag{9.41}$$

где N — число значений функции $f(x)$.

Для ускорения вычислений коэффициентов $A(n)$ можно использовать БПФ.

Таким образом, численное решение задачи о движении волнового пакета может быть найдено в соответствии со следующим алгоритмом:

1. Задать функцию $f(x)$, описывающую начальное возмущение при $t = 0$.

2. Задать функцию $\omega = \omega(k)$.
3. Задать границы пространственного интервала, на котором ищется решение задачи.
4. Задать число узлов пространственной сетки.
5. Вычислить значения функции $f(x)$ в узлах пространственной сетки.
6. Вычислить коэффициенты разложения функции $f(x)$ в ряд Фурье.
7. Вычислить значения функции $u(x,t)$ в заданный момент времени в соответствии с (9.41).

Для реализации данного алгоритма в MATLAB необходимо создать файл `Omega.m`, содержащий описание функции, возвращающей мгновенные значения зависимости циклической частоты от волнового вектора. Ниже приведен листинг данного файла.

```
% листинг файла Omega.m
function z=Omega(k)
% функция, возвращающая значения
% зависимости циклической частоты
% от волнового вектора

z=k+0.5*k.^2;
```

Далее необходимо выполнить последовательность команд, сохраненную нами в файле `Glava9_5.m`, листинг которого приведен ниже.

```
% листинг файла Glava9_5.m
clear all;

% задание пространственной
% координатной сетки
Nx=1024;
j=1:Nx;
xmin=-10*pi;
xmax=60*pi;
x(j)=xmin+(xmax-xmin)/(Nx-1)*(j-1);

% задание параметров волнового пакета
A=1; % амплитуда волнового пакета
k=0.5; % волновой вектор пакета

% вычисление значений комплексной
% функции, описывающей пакет в
% момент времени t = 0
```

```
z=A*exp(i*k.*x).*exp(-(x/10).^2);

% визуализация волнового пакета
% момент времени t = 0

figure(1);
plot(x,abs(z))

figure(2);
plot(x,real(z),x,imag(z),'--')

% вычисление и визуализация
% спектра волнового пакета
s=fft(z);
m=2:200;
m=1:200;
s1(m)=s(m+1);
figure(3);
plot(2*pi*m/(xmax-xmin),abs(s1))


% вычисление мгновенных значений
% решения волнового уравнения в
% момент времени t = 50
k=2*pi/(xmax-xmin);
t1=50;
for j=1:Nx
    su=0;
    for n=1:Nx
        su=su+...
           s(n)/Nx*exp(i*(k*n*(x(j)-xmin)-...
           Omega(k.*n)*t1));
    end;
    psi(j)=su;
end;

% вычисление мгновенных
% значений решения
% волнового уравнения в
% момент времени t = 90
t1=90;
for j=1:Nx
    su=0;
    for n=1:Nx
        su=su+...
```

```
        s(n)/Nx*exp(i*(k*n*(x(j)-xmin)-...
        Omega(k.*n)*t1));
    end;
    psi1(j)=su;
 end;

% визуализация огибающей
% волнового пакета
% в момент времени t = 0,50,90
figure(4);
plot(x,abs(z),...
     x,abs(psi),...
     x,abs(psi1));
```

Результаты выполнения приведенных выше команд представлены на рис. 9.18–9.21.

Рис. 9.18. Мнимая и действительная части начального возмущения

Рис. 9.19. Огибающая волнового пакета в момент времени $t = 0$

Рис. 9.20. Спектр функции $f(x)$

Рис. 9.21. Огибающие волнового пакета в моменты времени $t = 0, 50, 90$

Анализ результатов, представленных на рис. 9.18–9.21, позволяет сделать следующие выводы:

1. Волновой пакет локализован в пространстве на отрезке $[-20, 20]$.
2. В момент времени $t = 0$ пакет представляет собой суперпозицию синусоидальных волн, волновые числа которых находятся в диапазоне $[0; 1]$.
3. Спектр имеет максимальное значение при $r_m = 0,5$, ширина спектра $\Delta k \approx 1$.
4. В процессе движения меняется форма волнового пакета, происходит уменьшение высоты огибающей пакета при одновременном увеличении его ширины (расплывание волнового пакета в пространстве).
5. Изменение формы волнового пакета происходит в соответствии с законом сохранения энергии
$$\sum_{n=0}^{N} u(t_i, x_n)^2 = \text{const}. \tag{9.42}$$

***Задача* 9.22**

1. *Оцените фазовую скорость гармоники с волновым числом, равным k_m.*
2. *Оцените фазовые скорости гармоник v_f, волновые векторы $k_{1,2}$ которых удовлетворяют условиям*
$$|S(k_{1,2})| = 0.5\,|S(k_m)|\,,$$
$$|S(k_{1,2})| = 0.1\,|S(k_m)|\,.$$
3. *Оцените разброс фазовых скоростей в каждом из рассмотренных в предыдущем задании случае.*
4. *Оцените групповую скорость движения пакета v_p (скорость движения его огибающей). Сравните данную величину с ее точным значением, полученным по известной формуле [5]:*
$$v_p = \left.\frac{d\omega(k)}{dk}\right|_{k = k_m}. \tag{9.43}$$
5. *Оцените групповые скорости гармоник v_p, волновые векторы $k_{1,2}$ которых удовлетворяют условиям*
$$|S(k_{1,2})| = 0,5\,|S(k_m)|\,,$$
$$|S(k_{1,2})| = 0,1\,|S(k_m)|\,.$$
6. *Оцените разброс фазовых скоростей в каждом из рассмотренных в предыдущем задании случае.*

***Задача* 9.23**

Дополните описанный выше документ фрагментом, позволяющим вычислять мгновенные значения энергии волнового пакета, и проверьте выполнение закона сохранения энергии для пакета, локализованного при $t = 0$ в пространстве.

***Задача* 9.24**

1. *Дополните описанный выше документ фрагментом, позволяющим определять ширину волнового пакета Δu при заданном значении его огибающей.*

2. *Используя описанный выше документ, оцените закон изменения ширины волнового пакета во времени $(\Delta u)_t$.*

3. *Сравните полученный результат с известной оценочной формулой, приведенной в [5]:*
$$(\Delta u)_t = (\Delta u)_0 + \Delta v_p t, \tag{9.44}$$
где $(\Delta u)_0$ — ширина пакета при $t = 0$, Δv_p — разброс групповых скоростей:
$$\Delta v_p = \left(\frac{dv_g}{dk}\right)\Bigg|_{\ k = k_m} \Delta k = \left(\frac{d^2\omega}{dk}\right)\Bigg|_{\ k = k_m} \Delta k\ . \tag{9.45}$$

4. *Убедитесь в справедливости принципа неопределенности для волновых пакетов*
$$\Delta k(\Delta u)_t \geq 2\pi. \tag{9.46}$$

***Задача* 9.25**

Решите задачи 9.22—9.24 для волновых пакетов с прямоугольной и треугольными формами огибающей.

***Задача* 9.26**

Задайте начальное возмущение в виде двух прямоугольных импульсов шириной Δx, разнесенных в пространстве на расстояние ΔX. Определите, через какое время (время перекрывания импульсов) разрешение данных импульсов окажется невозможным.

Рассмотрите случаи:

1) $\Delta x \geq X$;

2) $\Delta x \leq X$.

Как зависит данный результат от значения k_m?

Рассмотрим другую постановку задачи о движении волнового пакета. Предположим, что передатчик, расположенный в точке $x = 0$, действует на непрерывную одномерную однородную открытую систему таким образом, что функция, описывающая бегущую волну в точке $x = 0$, имеет известную зависимость от времени $f(t)$:
$$f(t) = u(0, t). \tag{9.47}$$

При этом будем считать, что волновой пакет локализован во времени, то есть огибающая пакета является функцией, достаточно быстро стремящейся к нулю при $t \to \infty$. Сделанное предположение позволяет представить функцию $f(t)$ в виде интеграла Фурье:

$$f\,(t) = \int\limits_0^\infty A\,(\omega)\;\, e^{i\omega t} d\omega, \tag{9.48}$$

где

$$A\,(\omega) = \int\limits_0^\infty f\,(t)\;\, e^{-i\omega t} dt. \tag{9.49}$$

Каждая гармоника, как и в рассмотренном выше случае, определяет собственную гармоническую бегущую волну с волновым числом k, значение которого определяется из дисперсионного соотношения

$$k = k(\omega). \tag{9.50}$$

Здесь каждая частотная составляющая бегущей волны распространяется с фазовой скоростью

$$v_f = \frac{\omega}{k\,(\omega)}. \tag{9.51}$$

Искомая функция $u(x,t)$, описывающая бегущую волну, является суперпозицией данных гармонических бегущих волн. Это означает, что найти $u(x,t)$ можно заменой ωt на $\omega t - kx$ в (9.48) каждой гармонической составляющей:

$$u\,(x,t) = \int\limits_0^\infty A\,(\omega)\;\, e^{i(\omega\, t - k(\omega) x)} d\omega. \tag{9.52}$$

Таким образом, алгоритм аналитического решения задачи, как и в предыдущем случае, достаточно прост:

1. Задать функцию $f(t)$, описывающую начальное возмущение при $x = 0$.
2. Задать функцию $k = k(\omega)$.
3. Определить функцию $A(\omega)$, вычислив интеграл (9.49).
4. Определить функцию $u(x,t)$, вычислив интеграл (9.52).

Однако для того чтобы использовать данный алгоритм при нахождении численных значений функции $u(x,t)$, требуется его некоторая доработка, связанная прежде всего с тем, что вычисление интегралов в (9.48), (9.49), (9.52) приходится проводить на временных интервалах конечной длительности T, то есть проводить разложение функции не в интеграл, а в ряд Фурье. В данном случае, как было показано в главе 7, вместо (9.48), (9.49), (9.52) необходимо использовать следующие выражения:

$$f\,(t) = \sum_{n=0}^{N-1} A\,(n)\;\, e^{i(2\pi/T)\, n\, t}, \tag{9.53}$$

$$A(n) = \frac{1}{T} T \int\limits_0^T f(t)\ e^{-i(2\pi/T)nt} dt, \tag{9.54}$$

где $n = 0, 1, \ldots, N-1$,

$$u(x,t) = \sum_{n=0}^{N} A(n)\, e^{i[(2\pi/T)nt - k((2\pi/T)n)x]}, \tag{9.55}$$

где N — число значений функции $f(x)$.

Для ускорения вычислений коэффициентов $A(n)$ можно использовать БПФ.

Таким образом, численное решение задачи о движении волнового пакета может быть найдено в соответствии со следующим алгоритмом:

1. Задать функцию $f(t)$, описывающую начальное возмущение при $t = 0$.
2. Задать функцию $k = k(\omega)$. (Отметим, что при невозможности аналитического обращения дисперсионного соотношения $\omega = \omega(k)$ следует для каждого заданного значения частоты $\omega_s = 2\pi s/T$, $s = 1, 2, \ldots, N$, соответствующее значение волнового числа находить численно как корень уравнения $\omega_s = \omega(k_s)$.)
3. Задать границы временного интервала, на котором ищется решение задачи.
4. Задать число узлов временной сетки.
5. Вычислить значения функции $f(t)$ в узлах временной сетки.
6. Вычислить коэффициенты разложения функции $f(t)$ в ряд Фурье.
7. Вычислить значения функции $u(x,t)$ в заданный момент времени в соответствии с (9.55).

Для реализации данного алгоритма для волнового пакета с дисперсионным соотношением вида $\omega(k) = k + \alpha k^2$ необходимо создать файла `K.m`, содержащий описание функции, возвращающей значения функции (9.50). Ниже приведен листинг данного файла.

```
% листинг файла K.m
function z=K(Omega,Nu,alpha)
% описание функции (9.50)
if alpha==0
   z=Omega/Nu;
else
   z=-1/(2*alpha)+...
     ((1/2*alpha).^2+...
     Omega./(alpha*Nu))^0.5;
end;
```

Далее необходимо выполнить последовательность команд, сохраненную нами в файле `Glava9_6.m`, листинг которого приведен ниже.

```
% листинг файла Glava9_6.m
clear all

% задание временной сетки
tmin=0;
tmax=300;
Nt=1024;
j=1:Nt;
t(j)=tmin+(tmax-tmin)/(Nt-1)*(j-1);

% вычисление и визуализация
% значений функции,
% описывающей волновой пакет
% в момент времени t = 0
Omega=0.7;
A=1;
z=A*exp(i*Omega*t).*exp(-((t-40)/10).^2);

figure(1);
plot(t,real(z),...
     t,imag(z))

figure(2)
plot(t,abs(z))

% вычисление и визуализация
% спектра начального возмущения
s=fft(z);
j=1:200;
s1(j)=s(j+1);
freq=2*pi/(tmax-tmin)*j;

figure(3);
plot(freq,abs(s1))

v=1; % скорость движения пакета
alpha=0.5; % коэффициент, входящий
           % в дисперсионное соотношение

% вычисление значений функций,
% описывающих волновой пакет в
% выбранных точках пространства,
% и их визуализация
omega=2*pi/(tmax-tmin);
```

```
x=50;
for j=1:Nt
    su=0;
    for n=1:Nt
         su=su+s(n)/Nt*...
             exp(-i*(K(omega*n,v,alpha)*x-omega*n*t(j)));
    end;
    psi(j)=su;
end;

x=150;
for j=1:Nt
    su=0;
    for n=1:Nt
         su=su+s(n)/Nt*...
             exp(-i*(K(omega*n,v,alpha)*x-omega*n*t(j)));
    end;
    psi1(j)=su;
end;

figure(4)
plot(t,abs(z),...
     t,abs(psi),...
     t,abs(psi1));
```

Результат выполнения описанной выше последовательности команд представлен на рис. 9.22–9.25.

Рис. 9.22. Мнимая и действительная части начального возмущения

Рис. 9.23. Огибающая волнового пакета в момент времени $t = 0$

Анализ результатов, представленных на рис. 9.22, 9.23, показывает, что волновой пакет локализован на отрезке $[20, 60]$. Анализ спектра, представленного на рис. 9.24, показывает, что в точке $x = 0$, пакет представляет собой суперпозицию синусои-

Рис. 9.24. Спектр функции $f(x)$

Рис. 9.25. Огибающие волнового пакета в точках $x = 0, 50, 150$

дальных волн, частоты которых находятся в диапазоне $[0,3; 9,1]$. Спектр имеет максимальное значение при $\omega_m \cong 0,71$, ширина спектра $\Delta\omega \approx 0,8$. В процессе движения меняется форма волнового пакета: происходит уменьшение высоты огибающей пакета при одновременном увеличении его ширины (расползание волнового пакета в пространстве). Изменение формы волнового пакета происходит в соответствии с законом сохранения энергии (9.42).

***Задача* 9.27**

1. *Оцените фазовую скорость гармоники с частотой, равной* ω_m.

2. *Оцените фазовые скорости гармоник* v_f, *волновые векторы* k_1, k_2 *которых удовлетворяют условиям*

$$|S(\omega_{1,2})| = 0,5\,|S(\omega_m)|\,,$$

$$|S(\omega_{1,2})| = 0,1\,|S(\omega_m)|\,.$$

3. *Оцените разброс фазовых скоростей в каждом из рассмотренных в предыдущем задании случае.*

4. *Оцените групповую скорость движения пакета* v_f *(скорость движения его огибающей). Сравните данную величину с ее точным значением, полученным по известной формуле [5]:*

$$v_p = \left.\frac{d\omega(k)}{dk}\right|_{\;k\,=\,k(\omega_m)}\,. \tag{9.56}$$

5. *Оцените групповые скорости гармоник* v_f, *частоты* ω_1, ω_2 *которых удовлетворяют условиям*

$$|S(\omega_{1,2})| = 0,5\,|S(\omega_m)|\,,$$

$$|S(\omega_{1,2})| = 0,1\,|S(\omega_m)|\,.$$

6. *Оцените разброс фазовых скоростей в каждом из рассмотренных в предыдущем разделе случае.*

Задача 9.28

Дополните описанный выше документ фрагментом, позволяющим вычислять мгновенные значения энергии волнового пакета, и проверьте выполнение закона сохранения энергии для пакета, локализованного в точке $x = 0$ во времени.

Задача 9.29

Дополните описанный выше документ фрагментом, позволяющим определять ширину волнового пакета Δu при заданном значении его огибающей.

1. *Используя описанный выше документ, оцените закон изменения ширины волнового пакета во времени $(\Delta u)_t$.*

2. *Сравните полученный результат с известной оценочной формулой, приведенной в [5]:*
$$(\Delta u)_t = (\Delta u)_0 + \Delta\omega\, t, \tag{9.57}$$
где $(\Delta z)_0$ — ширина пакета при $t = 0$, $\Delta\omega$ — ширина спектра пакета.

3. *Убедитесь в справедливости принципа неопределенности для волновых пакетов*
$$\Delta\omega(\Delta u)_t \geq 2\pi. \tag{9.58}$$

Задача 9.30

Решите задачи 9.27—9.29 для волновых пакетов с прямоугольной и треугольными формами огибающей.

Задача 9.31

Антенна, находящаяся в точке $x = 0$, излучает два прямоугольных импульса шириной ΔT, разнесенных во времени на интервал ΔT_0. Определите, на каком удалении от антенны разделение данных импульсов окажется невозможным. Рассмотрите случаи:

1. $\Delta T \geq \Delta T_0$,

2. $\Delta T \leq \Delta T_0$,

Как зависит время перекрытия импульсов от значения ω_m?

Задача 9.32

Модифицируйте описанный выше документ, дополнив его фрагментом, позволяющим вычислять значения волновых чисел, для пакетов с произвольным законом дисперсии. Используя модифицированный документ, опишите распространение волн на поверхности глубокой воды (длина волны много меньше слоя воды), имеющих следующее дисперсионное соотношение:
$$\omega = gk + \frac{T}{\rho}k^3, \tag{9.59}$$

где $g = 980$ *см/с,* $\rho \approx 1,0$ *г/см*^{3}, $T \approx 72$ *дин/см (коэффициент поверхностного натяжения).*

1. *Постройте графики зависимостей фазовой и групповой скоростей движения волны от длины волны. При какой длине волны эти скорости равны?*

2. *Исследуйте движения волновых пакетов, распространяющихся по поверхности глубокой воды, аналогично тому, как это было сделано в данном разделе выше.*

Указание: *корни уравнения (9.59) в MATLAB можно найти следующей последовательностью действий:*

1. *Создать файл* `dispw.m`, *содержащий описание функции*

$$f(\omega, k) = \omega - gk + \frac{T}{\rho}k^3 :$$

```
% листинг файла dispw.m
function z=dispw(k)
% описание функции
global omega g T ro

z=omega-g*k+T/ro*k^3;
```

2. *Выполнить следующую последовательность команд, сохраненную нами в файле* `Glava9_7.m`:

```
% листинг файла Glava9_7.m:

global omega g T ro;
g=980;
T=72;
ro=1;
omega=1000;

fzero('dispw',0)
```

Здесь для нахождения корня уравнения мы использовали функцию `fzeros`, *обращение к которой может быть выполнено несколькими способами:*

(a) `fzero('имя_файла',начальное_приближение);`

(b) `fzero('имя_файла',[x1 x2])`
Здесь x1, x2 — начальная и конечная точки интервала, на котором ищется решение уравнения, выбираемые из условия

$$(x1) \cdot f(x2) < 0;$$

(c) `fzero('имя_файла',начальное_приближение,tol)`
`fzero('имя_файла',[x1 x2],tol)`
Здесь `tol` *— погрешность вычисления корня уравнения;*

(d) `options = optimset('Display','iter');`
`a = fzero(('имя_файла',начальное_приближение,options)`
`a = fzero('имя_файла',[x1 x2],options)`
(вывод на экран значений корня уравнения на каждом шаге итерационного процесса).

9.6. МОДЕЛИРОВАНИЕ ЯВЛЕНИЙ ИНТЕРФЕРЕНЦИИ И ДИФРАКЦИИ

Под явлением интерференции понимают сложение волн, приходящих по отдельности от одного и того же точечного источника, либо волн, испущенных двумя точечными источниками и имеющих одинаковую частоту и постоянную разность фаз. Классическим примером интерференции световых волн является опыт Юнга с двойной щелью, схема которого представлена на рис. 9.26.

Рис. 9.26. Схема опыта Юнга

Источник света S, излучающий свет только одной частоты (монохроматический источник), помещен на одинаковом расстоянии от двух одинаковых отверстий, имеющих координаты $(0, 0, -d/2)$, $(0, 0, d/2)$ соответственно. При этом размеры отверстий таковы, что их можно считать точечными источниками света, излучающими волны с одинаковой частотой и фазой (когерентные источники). На расстоянии L от точки с координатами $(0, 0, 0)$ расположен экран, на котором наблюдается распределение интенсивности волны, являющейся результатом интерференции волн, излученных когерентными точечными источниками.

Для ответа на вопрос, какая картина будет наблюдаться на экране, получим выражение, описывающее распределение интенсивности света от двух точечных источников. Электромагнитная волна, излучаемая монохроматическим источником света, является сферической волной. Напряженность электрического поля $E(r, t)$ в точке, удаленной от источника на расстояние r, определяется следующим выражением:

$$E\,(r, t) = \frac{A}{r} \cos\left(kr - \omega\, t + \varphi\right), \tag{9.60}$$

где A — амплитуда волны, φ — начальная фаза волны, которую далее для когерентных источников будем полагать равной нулю.

Множитель $1/r$ отражает тот факт, что интенсивность света убывает с увеличением расстояния между источником и точкой наблюдения. Из принципа суперпозиции известно, что полное электрическое поле, создаваемое двумя источниками в точке P, равно

$$E\left(t\right) = E_1 + E_2 = \frac{A}{r_1}\cos\left(kr_1 - \omega\, t\right) + \frac{A}{r_2}\cos\left(kr_2 - \omega\, t\right). \tag{9.61}$$

Напряженность электрического поля в точке P, создаваемого N источниками света, положение которых в пространстве относительно системы координат XOY задано их радиусами-векторами $\vec{R}_i$, $i = 0, 1, \ldots, N-1$, определяется по формуле

$$E\left(t, \vec{r}\right) = \sum_{i=0}^{N-1} \frac{A}{\left|\vec{r} - \vec{R}_i\right|}\cos\left(\frac{2\pi}{\lambda}\left|\vec{r} - \vec{R}_i\right| - \omega\, t\right), \tag{9.62}$$

где $\vec{r}$ — радиус-вектор точки наблюдения.

Наблюдаемая интенсивность волны I в точке P равна

$$I = < E^2 >, \tag{9.63}$$

где угловые скобки $< >$ означают усреднение по времени:

$$\left\langle E^2 \right\rangle = \frac{1}{T}\int\limits_0^T E(r,t)^2 dt. \tag{9.64}$$

Для вычисления среднего оказывается удобным преобразовать (9.64) следующим образом, поскольку в соответствии с (9.60)

$$E = E(r, \omega t) = E(r, \frac{2\pi}{T}t),$$

то можно ввести безразмерную переменную $\xi = t/T$ и записать интеграл (9.64) в следующем виде:

$$\left\langle E^2 \right\rangle = \int\limits_0^1 E(r,\xi)^2 d\xi = \int\limits_0^1 \left(\sum_{i=0}^{N-1} \frac{A}{\left|\vec{r} - \vec{R}_i\right|}\cos\left(\frac{2\pi}{\lambda}\left|\vec{r} - \vec{R}_i\right| - 2\pi\xi\right)\right)^2 d\xi. \tag{9.65}$$

Вычисление интеграла в (9.65) для произвольного количества источников не встречает принципиальных трудностей, однако получающиеся в результате выражения оказываются весьма громоздкими, и для их анализа приходится использовать ПК. В этих условиях представляется оправданным проводить вычисления интенсивности в соответствии с (9.65) сразу в численном виде. Алгоритм решения данной задачи достаточно очевиден:

1. Задать функцию, описывающую интенсивность световой волны, в соответствии с (9.65).

2. Задать количество источников света и длину излучаемой световой волны.

3. Задать расположение источников света (набор векторов $\vec{R}_i$, $i = 0, 1, \ldots, N-1$).

4. Задать пространственную сетку, в узлах которой производится вычисление интенсивности световой волны.

Реализация описанного алгоритма осуществляется в MATLAB следующей последовательностью действий:

1. Создание файла `Intensity.m`, содержащего описание функции, возвращающей значение интенсивности в заданной точке пространства в соответствии с (9.65). Ниже представлен листинг данного файла.

```
% листинг файла Intensity.m
function E=Intensity(Lambda,N,A,R0,r)
% описание функции, возвращающей
% значение интенсивности в заданной
% точке пространства в соответствии с (9.65)
% аргументы функции:
% Lambda - длина волны
% N - число источников
% A - вектор, содержащий значения
%     интенсивностей источников света
% R0 - матрица, столбцы которой содержат
%      координаты источников света
% r  - вектор-столбец, содержащий
%      координаты точки наблюдения

su=0;
ksi=0:10^-3:1; % задание сетки для
               % вычисления интеграла в (9.65)
Ni=length(ksi); % число узлов сетки

% вычисление длин радиусов-векторов
% от источников света до точки наблюдения
% R(1) - расстояние от первого источника света
% до точки наблюдения
% R(2) - расстояние от второго источника света
% до точки наблюдения
for i=1:N
    Rr(i)=norm(r-R0(:,i));
end;

% вычисление значений
% подынтегральной функции в (9.65)
for i=1:Ni
```

```
    su=0;
  for j=1:N
      su=su+A(j)/Rr(j)*cos(2*pi/Lambda*Rr(j)-2*pi*ksi(i));
  end;
  f(i)=su^2;
end;
E=trapz(ksi,f); % вычисление интеграла в (9.65)
                % методом трапеций
```

2. Выполнение следующей последовательности команд, сохраненной нами в файле `Glava9_8.m`:

```
% листинг файла Glava9_8.m
clear all

N=2; % число источников света

% задание параметров и координат
% источников света
A(1)=1;
A(2)=1;
Lambda=5000*10^-7;
R0=[0 0 -0.1];
R1=[0 0 0.1];
R=cat(2,R0',R1');% создание матрицы,
                 % столбцы которой содержат
                 % координаты источников света

% задание сетки для вычисления
% распределения интенсивности
% световой волны
Np=201;
i=1:Np;
zmin=-5;
zmax=5;
z=zmin+(zmax-zmin)/(Np-1)*(i-1);
L=200;
% вычисление и визуализация интенсивности
% световой волны на отрезке
% x = 0, y = L, -zmin<=z<=zmax

for i=1:Np
     r=[0 L z(i)]';
     I(i)=Intensity(Lambda,N,A,R,r);
end;
```

```
Imax=max(I);
plot(z,I/Imax);
```

Результаты выполнения описанной выше последовательности команд представлены на рис. 9.27.

Рис. 9.27. Распределение интенсивности световой волны на отрезке $x = 0, y = L, z \in [z\min, z\max]$

Как показывает практика работы с данными файлами, на вычисление интеграла (9.65) даже вдоль одной прямой MATLAB требуется достаточно большое количество времени, которое, как очевидно, будет в Np раз больше при анализе распределения интенсивности на плоскости $y = L$. Можно устранить отмеченный недостаток, если вместо вычисления интеграла (9.65) проводить усреднение напряженности суммарного электрического поля по конечному числу точек интервала $[0, 1]$. Функция, реализующая данное усреднение, имеет следующий вид:

$$E1\left(\lambda, A, N, R0, r, Nb\right) = \frac{1}{Nb}\sum_{j=1}^{Nb}\left(\sum_{i=1}^{N}\frac{A}{|r - R0_i|}\cdot\cos\left(\frac{2\cdot\pi}{\lambda}\cdot|r - R0_i| - \frac{2\cdot\pi}{Nb}\cdot j\right)\right)^2, \qquad (9.66)$$

здесь λ — длина волны, A — амплитуда волны, N — количество источников света, $R0$ — составной массив, содержащий радиусы-векторы источников света, r — радиус-вектор точки наблюдения, Nb — число точек усреднения.

Для реализации вычислений в соответствие с (9.66) необходимо создать файл, названный нами `Intensity1.m`, содержащий описание соответствующей функции.

```
% листинг файла Intensity1.m
function E=Intensity1(Lambda,A,N,R0,r,Nb)
% описание функции, возвращающей
% значение интенсивности в заданной
% точке пространства в соответствии с (9.66)
% аргументы функции:
% Lambda - длина волны
```

```
% N - число источников
% A - вектор, содержащий значения
%     интенсивностей источников света
% R0 - матрица, столбцы которой
%      содержат координаты источников света
% r  - вектор-столбец, содержащий
%      координаты точки наблюдения

% вычисление длин радиусов-векторов
% от источников света до точки наблюдения
% R(1) - расстояние от первого источника света
%        до точки наблюдения
% R(2) - расстояние от второго источника света
%        до точки наблюдения
for i=1:N
    Rr(i)=norm(r-R0(:,i));
end;

% вычисление значений
% подынтегральной функции в (9.66)
for i=1:Nb
  su=0;
  for j=1:N
       su=su+A(j)/Rr(j)*cos(2*pi/Lambda*Rr(j)-2*pi/Nb*i);
  end;
  f(i)=su.^2;
end;
E=mean(f); % вычисление среднего значения
           % подынтегральной функции в (9.66)
```

Для вычисления распределения интенсивности в соответствии с (9.66) необходимо выполнить следующую последовательность команд, которую мы сохранили в файле `Glava9_9.m`:

```
% листинг файла Glava9_9.m
N=2; % число источников света
% задание параметров и координат
% источников света
A(1)=1;
A(2)=1;
Lambda=5000*10^-7;
R0=[0 0 -0.1];
R1=[0 0 0.1];
R=cat(2,R0',R1');% создание матрицы,
                 % столбцы которой содержат
                 % координаты источников света
```

```
% задание сетки для вычисления
% распределения интенсивности
% световой волны
Np=201;
i=1:Np;
zmin=-5;
zmax=5;
z=zmin+(zmax-zmin)/(Np-1)*(i-1);
L=200;
% вычисление и визуализация
% интенсивности световой волны на отрезке
% x = 0, y = L, -zmin<=z<=zmax
Nb=3;
for i=1:Np
     r=[0 L z(i)]';
     I1(i)=Intensity1(Lambda,A,N,R,r,Nb);
end;
Imax=max(I1);
plot(z,I1/Imax);
```

Анализ разности между значениями распределений интенсивностей `I`, `I1`, вычисленными на отрезке $x = 0$, $y = L$, $z \in [z\min, z\max]$ (рис. 9.28), показывает, что уже при усреднении по трем точкам абсолютное значение разности не превосходит $1 \cdot 10^{-10}$, поэтому далее при решении задач следует использовать функцию `Intensity1`.

Рис. 9.28. Абсолютное значение разности между значениями распределений интенсивностей `I`, `I1`, вычисленными на отрезке $x = 0$, $y = L$, $z \in [z\min, z\max]$

Для анализа пространственного распределения интенсивности необходимо выполнить следующую последовательность команд, сохраненную нами в файле `Glava9_10.m`:

```
%  листинг файла Glava9_10.m
clear all
```

```
N=2; % число источников света
% задание параметров и координат
% источников света
A(1)=1;
A(2)=1;
Lambda=5000*10^-7;
R0=[0 0 -0.1];
R1=[0 0 0.1];
R=cat(2,R0',R1');% создание матрицы,
                 % столбцы которой содержат
                 % координаты источников света

% задание пространственной сетки
% по осям OX, OZ
Np=201; % число узлов по осям OX, OZ
i=1:Np;
xmin=-5;
xmax=5;
x(i)=xmin+(xmax-xmin)/(Np-1)*(i-1);

i=1:Np;
zmin=-5;
zmax=5;
z(i)=zmin+(zmax-zmin)/(Np-1)*(i-1);
L=200;

% вычисление и визуализация
% распределения интенсивности
% в соответствии с (9.66)
for i=1:Np
    for j=1:Np
        r=[x(j) L z(i)]';
        I1(i,j)=Intensity1(Lambda,A,N,R,r,3);
    end;
end;

% визуализация распределения
% интенсивности световой волны на
% отрезке  x=-4.0452, y=L, -5<z<=5
figure (1)
plot(z,I1(:,20)/max(I1(:,20)))

% визуализация распределения
% интенсивности световой волны на
% отрезке  -5<=x<=5,  y=L, z=-4.0452
```

```
figure(2)
plot(z,I1(20,:)/max(I1(20,:)))

% Визуализация  распределения
% интенсивности плоскости y=L
[X Z]=meshgrid(x,z);
figure(3)
contourf(X,Z,I1)
colormap gray
xlabel('x');
ylabel('z')
```

Результаты выполнения описанной последовательности команд представлены на рис. 9.29–9.31.

Рис. 9.29. Распределение интенсивности освещенности световой волны на отрезке $x = -4,0452$, $y = L$, $z \in [-5, 5]$

Рис. 9.30. Распределение интенсивности освещенности световой волны на отрезке $x = -4,0452$, $y = L$, $z \in [-5, 5]$

Рис. 9.31. Распределение интенсивности на отрезке $x \in [-5, 5]$, $z = -4,0452$, $y = L$

***Задача* 9.33**

1. *Используя описанный выше документ, проверьте, что координаты минимумов интенсивности волны для выбранных геометрических размеров системы в плоскости $y = L$ находятся из условия*

$$\frac{\pi\, d \sin\Theta}{\lambda} = \frac{\pi}{2}, \tag{9.67}$$

где $\sin\Theta \approx z_n/L$.

2. *Получите выражение (9.67) аналитически, проинтегрировав (9.65).*

3. *Исследуйте, как меняется положение первого минимума при изменении λ в диапазоне* $[4 \cdot 10^{-4}; 6 \cdot 10^{-4}]$ *мм.*

4. *Зафиксируйте значение λ и проварьируйте L от* 1 *до* 300 *мм. Как зависит положение максимума интенсивности от L при*

$$L \gg (x_{\max} - x_{\min}), (y_{\max} - y_{\min})?$$

***Задача* 9.34**

Проведите анализ распределения интенсивности в плоскости $y = L$, для случаев когда число щелей $N = 3, 4, \ldots, 20$, считая, что $\lambda = 5 \cdot 10^{-4}$ мм, $L = 200$ мм, $d = 0.09$. (Описанная модель является моделью дифракционной решетки.)

1. *Постройте зависимость максимального значения интенсивности от числа щелей N.*

2. *Как зависит расстояние между пиками от числа щелей?*

3. *Проварьируйте L в диапазоне от 1 до 300 мм. Как зависит положение максимума интенсивности от L при*

$$L \gg (x_{\max} - x_{\min}), (y_{\max} - y_{\min})?$$

***Задача* 9.35**

На практике ширина реальных щелей гораздо больше длины волны, поэтому более адекватной является модель, в которой щель рассматривается как набор когерентных источников сферических волн (принцип Гюйгенса).

1. *Рассчитайте распределение интенсивности света с длиной волны $\lambda = 5 \cdot 10^{-4}$ мм от одной щели конечного размера, заменяя его $N = 20$ точечными источниками, отстоящими друг от друга на* $0,001$ *мм. Определите ширину центрального пика интенсивности при $L = 200$ мм. Как соотносится ширина центрального пика с шириной щели? Как зависят полученные результаты от числа источников N?*

2. *Как зависит положение первого минимума дифракционной картины от длины волны, ширины щели и расстояния до экрана.*

3. *Проведите сравнительный анализ распределений интенсивности для* $L = 1$ *мм и* $L = 50$ *мм.*

Задача 9.36

Исследуйте распределение интенсивности света при интерференции на системе двух щелей, используя модель щели конечной ширины. Для этого замените каждую из щелей 20 *точечными источниками, отстоящими друг от друга на* $1 \cdot 10^{-4}$ *мм, расстояния между серединами щелей задайте равными* $0,1$ *мм. Как соотносится распределение интенсивности в данной модели с аналогичными распределениями для двух точечных щелей? Как зависит распределение интенсивности от длины волны?*

Задача 9.37

Подход, использованный в задачах 9.35, 9.36, может быть применен для вычисления распределения интенсивностей волн, прошедших через отверстия конечной ширины и высоты (дифракция). Для этого следует представить соответствующее отверстие как набор N *когерентных источников сферических волн, равномерно распределенных по плоскости отверстия. Рассчитайте распределение интенсивности для прямоугольного отверстия размером* 1×3 *мм, освещаемого светом длиной волны* $\lambda = 5 \cdot 10^{-4}$ *мм. Сравните распределение интенсивностей для различных значений* $L \in [3, 200]$ *мм, определив предварительно каждого* L *минимальные значения* N*, начиная с которых распределение интенсивности не зависит от числа источников. Постройте зависимость числа* N *от расстояния* L*.*

Исследуем влияние размеров источника когерентного излучения на распределение интенсивности при интерференции на системе двух точечных щелей (рис. 9.32).

Рис. 9.32. К расчету интерференционной картины от протяженного источника

Предваряя численные расчеты, проведем качественную оценку, позволяющую ввести величину, определяющую распределение интенсивности света в плоскости $y = L$. Заменим протяженный источник света двумя точечными источниками, расположенными соответственно в точках $(0, -L_s, a_s/2)$, $(0, -L_s, -a_s/2)$, и найдем распределение интенсивности света в плоскости $y = L$. В соответствии с принципом Гюйгенса

напряженность поля электромагнитной волны в точке с координатами $(0, L, z)$ есть сумма 4 волн:

1) волны, излученной источником, расположенным в точке

$$(0, -L_s, a_s/2),$$

и прошедшей через отверстие S_1;

2) волны, излученной источником, расположенным в точке

$$(0, -L_s, a_s/2),$$

и прошедшей через отверстие S_2;

3) волны, излученной источником, расположенным в точке

$$(0, -L_s, -a_s/2),$$

и прошедшей через отверстие S_1;

4) волны, излученной источником, расположенным в точке

$$(0, -L_s, -a_s/2),$$

и прошедшей через отверстие S_2.

При этом фазы интерферирующих волн оказываются различными, поэтому для нахождения напряженности суммарного поля следует учитывать начальную фазу, которую ранее для когерентного источника мы положили равной нулю (см. формулы (9.61), (9.62)). Обозначим начальные фазы этих волн φ_{11}, φ_{12}, φ_{21}, φ_{22} соответственно. Как очевидно из рис. 9.32, значения начальных фаз, зависящие от геометрических размеров рассматриваемой системы, определяются следующими выражениями

$$\varphi_{11} = \varphi_{22} = \sqrt{\left(\frac{d}{2} - \frac{a_s}{2}\right)^2 + L_s^2}, \tag{9.68}$$

$$\varphi_{12} = \varphi_{21} = \sqrt{\left(\frac{d}{2} - \frac{a_s}{2}\right)^2 + L_s^2}. \tag{9.69}$$

С учетом начальных фаз интерферирующих волн выражение для напряженности поля в точке $(0, L, z)$

$$\begin{aligned} E(\vec{r}, t) = &\frac{A}{|\vec{r}_1|}\left[\cos\left(\frac{2\pi}{\lambda}|\vec{r}_1| - \omega\, t + \varphi_{11}\right) + \cos\left(\frac{2\pi}{\lambda}|\vec{r}_1| - \omega\, t + \varphi_{21}\right)\right] + \\ &+ \frac{A}{|\vec{r}_2|}\left[\cos\left(\frac{2\pi}{\lambda}|\vec{r}_2| - \omega\, t + \varphi_{12}\right) + \cos\left(\frac{2\pi}{\lambda}|\vec{r}_2| - \omega\, t + \varphi_{22}\right)\right]. \end{aligned} \tag{9.70}$$

Используя известные тригонометрические формулы

$$\cos a + \cos b = \frac{1}{2}\cos\left(\frac{a+b}{2}\right)\cos\left(\frac{a-b}{2}\right),$$

приводим (9.70) к виду

$$\begin{aligned} E(\vec{r},t) =& \frac{A}{|\vec{r}_1|}\left[\cos\left(\frac{2\pi}{\lambda}|\vec{r}_1| - \omega\, t + \frac{1}{2}(\varphi_{11}+\varphi_{21})\right)\cos\left(\frac{1}{2}(\varphi_{11}-\varphi_{21})\right)\right] + \\ &+ \frac{A}{|\vec{r}_2|}\left[\cos\left(\frac{2\pi}{\lambda}|\vec{r}_2| - \omega\, t + \frac{1}{2}(\varphi_{12}+\varphi_{22})\right)\cos\left(\frac{1}{2}(\varphi_{12}-\varphi_{22})\right)\right]. \end{aligned} \tag{9.71}$$

Полагая, что d, $a_s \ll L_s$, из (9.68), (9.69) найдем, что

$$\frac{\varphi_{11}-\varphi_{21}}{2} = \frac{\varphi_{12}-\varphi_{22}}{2} = \frac{\pi\, da_s}{2\lambda L_s}, \tag{9.72}$$

Подставив (9.72) в (9.71), окончательно получим

$$\begin{aligned} E(\vec{r},t) =& \frac{A}{|\vec{r}_1|}\cos\left(\frac{2\pi}{\lambda}|\vec{r}_1| - \omega\, t + \frac{2\pi L_s}{\lambda}\right) + \\ &+ \frac{A}{|\vec{r}_2|}\cos\left(\frac{2\pi}{\lambda}|\vec{r}_2| - \omega\, t + \frac{2\pi L_s}{\lambda}\right)\cos\left(\frac{\pi\, da_s}{2\lambda L_s}\right). \end{aligned} \tag{9.73}$$

Из (9.73) видно, что напряженность электрического поля волны двух разнесенных когерентных источников отличается от напряженности электрического поля точечного источника множителем $\cos\left(\frac{\pi\, da_s}{2\lambda L_s}\right)$, поэтому условие, при котором отличие распределений данных интенсивностей невелико, можно записать в виде

$$\frac{\pi\, da_s}{2\lambda L_s} \le \frac{\pi}{2}. \tag{9.74}$$

Выражение (9.74) устанавливает связь между угловым размером источника $\Theta_s = a_s/L_s$ и угловым размером области волнового фронта Θ_{coh}, в которой излучение протяженного источника когерентно [7]:

$$\Theta_{coh} \le \frac{\lambda}{\Theta_s}. \tag{9.75}$$

***Задача* 9.38**

Проведите в опыте Юнга численные исследования распределения интенсивности волны, создаваемой протяженным источником когерентного излучения, представив его в виде набора N точечных источников, положение которых задается набором радиусов-векторов Rs_i, $i = 0, 1, \ldots, N-1$. Подтверждают ли результаты моделирования оценку размера области (9.75)?

Указание: *фазы волн φ_{0i}, φ_{1i}, излучаемые каждым точечным источником соответственно в точках S_1 и S_2, определяются выражением*

$$\phi_{mi} = \frac{2\pi}{\lambda}\left|\vec{R}_m - Rs_i\right|,$$

где $i = 0, 1, \ldots, N-1$, $m = 0, 1$, а напряженность электрического поля волны — выражением:

$$E(\vec{r}, t) = \sum_{m=0}^{1}\sum_{i=0}^{N-1}\left(\frac{A}{\left|\vec{r} - \vec{R}_m\right|}\cos\left(\frac{2\pi}{\lambda}\left|\vec{r} - \vec{R}_m\right| - \omega\, t + \varphi_{mi}\right)\right).$$

9.7. ГЕОМЕТРИЧЕСКАЯ ОПТИКА

В общем случае для описания процесса распространения волн необходимо решить волновое уравнение, являющееся дифференциальным уравнением в частных производных, с заданными начальными и граничными условиями. Методы ее решения, изучаемые в специальном разделе математики, который называется «Уравнения математической физики», в большинстве случаев оказываются весьма сложными. Однако существует важный частный случай, когда длина волны много меньше всех характерных размеров системы. В этом случае оказывается возможным использовать модель, в которой волна распространяется аналогично движению пучка частиц, движущихся по определенным траекториям-лучам, поэтому данное приближения получило название геометрической оптики. Распространение световых лучей подчиняется принципу Ферма (принцип наименьшего времени): ***луч света идет между двумя точками по такому пути, который требует наименьшего времени.***

Применим принцип Ферма к задаче преломления, в которой свет падает на поверхность раздела двух сред, в которых скорость света различна (рис. 9.33).

Принято выражать скорость света в среде v через скорость света в вакууме c и показатель преломления среды n

$$v = \frac{c}{n}. \qquad (9.76)$$

Время прохождения света из точки A в точку B, как видно из рис. 9.33, равно

$$t = \frac{\sqrt{a^2 + x^2}}{v_1} + \frac{\sqrt{b^2 + (L - x)^2}}{v_2}, \qquad (9.77)$$

Рис. 9.33. К иллюстрации принципа Ферма

где $v_1 = c/n_1$, $v_2 = c/n_2$.

В соответствии с принципом Ферма dx/dt, поэтому

$$\frac{1}{v_1}\frac{x}{\sqrt{a^2 + x^2}} - \frac{1}{v_2}\frac{L - x}{\sqrt{b^2 + (L - x)^2}} = 0. \qquad (9.78)$$

Так как $x\Big/\sqrt{a^2+x^2} = \sin\alpha$, $(L-x)\Big/\sqrt{b^2+(L-x)^2} = \sin\beta$, то (9.78) можно записать в следующем виде (закон Снелла):

$$n_1 \sin\alpha = n_2 \sin\beta. \tag{9.79}$$

Рассмотрим задачу о распространении луча в среде с неоднородным коэффициентом преломления. Будем считать, что луч падает на прозрачную среду с коэффициентом преломления n, зависящим от y ($n = n(y)$), под малым углом к нормали в точке $y = 0$ (рис. 9.34).

Рис. 9.34. Траектория луча, падающего нормально на полупространство с неравномерным коэффициентом преломления

Для нахождения уравнения, описывающего форму луча, рассмотрим луч света, проходящий через несколько плоскопараллельных пластинок с различными коэффициентами преломления (рис. 9.35). В соответствии с законом Снелла (9.79)

$$n_0 \sin\alpha = n_1 \sin\beta_1 = n_2 \sin\beta_2 = n_3 \sin\beta_3 = \ldots \tag{9.80}$$

Рис. 9.35. Графическая иллюстрация выражения (9.80)

Так как соотношение (9.80) не зависит от числа и ширины отдельных слоев, оно будет справедливо и при непрерывном изменении коэффициента преломления в направлении оси OY:

$$n(y) \sin\beta(y) = n_0 \sin\alpha, \tag{9.81}$$

где $\beta(y)$ — угол между направлением луча и осью OY (рис. 9.36).

Рис. 9.36. К получению уравнения, описывающего распространение луча в неоднородной среде

Как видно из рис. 9.36,

$$\mathrm{tg}\alpha\,(y) = \mathrm{ctg}\beta\,(y) = dy/dx,$$

поэтому

$$\sin\beta\,(y) = \frac{1}{\sqrt{1+\mathrm{ctg}^2\beta\,(y)}} = \frac{1}{\sqrt{1+(dy/dx)^2}}. \tag{9.82}$$

Подставляя (9.81) в (9.82) и учитывая, что $\alpha \cong \pi/2$, приводим (9.82) к виду

$$\frac{n\,(y)}{\sqrt{1+(dy/dx)^2}} = n_0. \tag{9.83}$$

Решив (9.83) относительно dx/dy, получим дифференциальное уравнение первого порядка с разделяющимися переменными

$$\frac{dy}{dx} = \sqrt{\left(\frac{n\,(y)}{n_0}\right)^2 - 1}. \tag{9.84}$$

Его решение имеет вид

$$x = \int\limits_{y_0}^{y} \frac{dy}{\sqrt{\left(\frac{n(y)}{n_0}\right)^2 - 1}}. \tag{9.85}$$

При произвольной зависимости $n(y)$, вообще говоря, интеграл в (9.85) может не выражаться через элементарные функции, поэтому для анализа траектории движения луча будем вычислять его численно. Для этого можно использовать следующий алгоритм:

1. Задать функцию $n(y)$.
2. Задать координату точки падения y_0.
3. Задать координату точки падения $y_{\max}$.

4. Вычислить координаты y_i, $i = 0, 1, \ldots, N-1$, точек разбиения отрезка $[y_0, y_{\max}]$ на N частей.

5. Вычислить N значений интеграла (9.85) на интервалах $[y_0, y_i]$.

6. Отобразить графически зависимость $y = y(x)$.

Для реализации данного алгоритма необходимо выполнить следующую последовательность действий:

1. Создать файл `CoeffRefraction.m`, содержащий описание функции, возвращающей значение функции $n(y)/n_0$, где n_0 — коэффициент преломления в точке падения луча.

```
% листинг файла CoeffRefraction.m
function z=CoeffRefraction(y)
% функция, возвращающая
% значение функции (1+4*y)^1/2
z=(1+4*y).^0.5;
```

2. Выполнить следующую последовательность команд, сохраненную нами в файле `Glava9_11.m`:

```
% листинг файла Glava9_11.m
clear all

% задание координатной сетки по оси OY
>> ymin=10^-5;
>> ymax=20;
>> Np=100;
>> i=1:Np;
>> y(i)=ymin+(ymax-ymin)/(Np-1)*(i-1);
>> Nk=1000; % число узлов
            % для вычисления
            % интеграла с переменным
            % верхним пределом

% вычисление значений
% интеграла с переменным
% верхним пределом (9.85)
>> for i=1:Np
        su=0;
        for k=1:Nk
           Y(k)=ymin+(y(i)-ymin)/(Nk-1)*(k-1);
           Z(k)=1/(CoeffRefraction(Y(k))^2-1)^0.5;
        end;
        Xb(i)=trapz(Y,Z);
```

```
        Yb(i)=Y(Nk);
    end;
>> plot(Xb,Yb)
```

Результат выполнения описанной выше последовательности команд представлен на рис. 9.37.

Рис. 9.37. Траектория движения луча в среде с переменным коэффициентом преломления

Полученный результат выглядит достаточно неожиданно. Можно было ожидать, что луч, падающий на границу раздела, должен идти вдоль оси OX, но не по искривленной траектории. Полученный результат требует дополнительного объяснения. В геометрической оптике под лучом понимается узкий пучок света, который с некоторым приближением можно считать участком фронта плоской волны (принцип Гюйгенса). При падении плоской волны на оптическую неоднородность во второй среде возбуждаются вторичные волны (рис. 9.38). Вторичная волна движется в среде с неоднородным показателем преломления с различными скоростями: тем медленнее, чем больше n, и наоборот. Это приводит к искажению фронта вторичных волн.

Рис. 9.38. К объяснению причины искажения луча при движении волны в среде с неоднородным коэффициентом преломления

Таким образом, лучи, рассматриваемые в геометрической оптике, представляют собой пример математической модели, используемой для описания ряда оптических явлений. Важно понимать, что в реальной природе «лучей» не существует. При невозможности решения задачи с точки зрения геометрической оптики, следует представить луч как перпендикуляр к участку фронта плоской волны, ширина которого значительно больше длины волны, и рассмотреть явление с точки зрения геометрической оптики.

***Задача* 9.39**

Используя описанный выше документ, проведите сравнительный анализ траекторий луча для различных законов изменения коэффициента преломления $f(y) = (1 + ay^{\alpha})^{\beta}$. *Почему в ряде случаев вычисление численного значения интеграла (9.85) оказывается невозможным, но при использовании в качестве нижней границы интегрирования некоторой величины* $\epsilon \ll 1$ *интеграл оказывается сходящимся?*

9.8. ПОЛЯРИЗАЦИЯ

В предыдущих разделах при рассмотрении волновых явлений нас не интересовало направление колебаний волны. В данном разделе мы рассмотрим явления, зависящие от поляризации волны. Отметим, что понятие поляризации используется только для описания поперечных волн, поскольку задание направления распространения продольной волны автоматически определяет направление колебаний среды. Для поперечной волны единственным ограничением направления колебаний является условие, в соответствии с которым они должны происходить в плоскости, перпендикулярной направлению распространения, поэтому, например, электрическое поле электромагнитной волны, распространяющейся в направлении оси OY, может быть представлено двумерной векторной функцией $\vec{E}(y,t) = (E_x(y,t), E_z(y,t))$. Если электромагнитная волна является монохроматической, то функции $E_x(y,t)$, $E_z(y,t)$ изменяются независимо друг от друга с одинаковой частотой. Отмеченное обстоятельство позволяет представить данные функции в следующем виде:

$$E_x(y,t) = E_{x0} \cos\left(\frac{2\pi}{\lambda} y - \omega\, t - \varphi\right), \tag{9.86}$$

$$E_y(y,t) = E_{y0} \cos\left(\frac{2\pi}{\lambda} y - \omega\, t\right), \tag{9.87}$$

где E_{x0}, E_{y0} — амплитуды соответствующих напряженностей.

Анализ выражений (9.86), (9.87) показывает, что оказывается удобным ввести безразмерные переменные: координату $\tilde{y} = y/\lambda$ и время $\tilde{t} = t/(2\pi/\omega)$, используя которые можно представить выражения (9.86), (9.87) в следующем виде:

$$E_x(\tilde{y},\tilde{t}) = E_{x0} \cos\left(2\pi\, \tilde{y} - 2\pi\, \tilde{t} - \varphi\right), \tag{9.88}$$

$$E_z(\tilde{y},\tilde{t}) = E_{y0} \cos\left(2\pi\, \tilde{y} - 2\pi\, \tilde{t}\right). \tag{9.89}$$

Для изучения основных свойств поперечных волн вычислим в выбранный момент времени t составляющие напряженности электрического поля волны, распространяющейся в положительном направлении оси OY. Решение рассматриваемой задачи в MATLAB находится выполнением следующей последовательности команд, сохраненной нами в файле `Glava9_12.m`:

```
% листинг файла Glava9_12.m
clear all
% задание координатной сетки
% для вычисления составляющих напряженности
% электрического поля
Np=100;
i=1:Np;
ymin=0;
ymax=2;
y(i)=ymin+(ymax-ymin)/(Np-1)*i;

% вычисление составляющих
% напряженности электрического поля
Ex0=1;
Ez0=1;

t=2.1; % выбранный момент времени
Ex(i)=Ex0*cos(2*pi*y(i)-2*pi*t+pi/2);
Ez(i)=Ez0*cos(2*pi*y(i)-2*pi*t);

% визуализация напряженности
% электрического поля в узлах координатной сетки
compass(Ex(1),Ez(1))
hold on
compass(Ex(10),Ez(10))
compass(Ex(20),Ez(20))
compass(Ex(30),Ez(30))
compass(Ex(40),Ez(40))
```

Результат выполнения приведенной выше последовательности команд представлен на рис. 9.39.

Для визуализации напряженности электрического поля, вычисленной в узлах координатной сетки, в трехмерном пространстве необходимо выполнить следующую последовательность команд, сохраненную нами в файле `Glava9_13.m`:

```
% листинг файла Glava9_13.m
clear all
% задание пространственной
% координатной сетки
Np=53; % число узлов сетки
i=1:Np;
x(i)=0;
```

Рис. 9.39. Визуализация проекции вектора напряженности магнитного поля на плоскость OY в момент времени $t = 2,1$

```
ymin=0;
ymax=2;
y(i)=ymin+(ymax-ymin)/(Np-1)*(i-1);
z(i)=0;
% вычисление составляющих
% напряженности электрического поля
Ex0=1;
Ez0=1;
t=2.1;
Ex(i)=Ex0*cos(2*pi*y(i)-2*pi*t+pi/2);
Ez(i)=Ez0*cos(2*pi*y(i)-2*pi*t);
Ey(i)=0;
quiver3(x,y,x,Ex,Ey,Ez)
view([50 36])
```

Результат выполнения приведенной выше последовательности команд представлен на рис. 9.40.

Из рис. 9.39, 9.40 видно, что вектор напряженности электрического поля описывает окружность и поворачивается при движении вдоль оси OY по часовой стрелке (если смотреть со стороны положительного направления оси OY). О такой волне говорят, что она имеет левую круговую поляризацию.

Рис. 9.40. Визуализация в трехмерном пространстве вектора напряженности магнитного поля в момент времени $t = 2,1$

***Задача* 9.40**

Используя описанный выше документ, определите поляризацию электромагнитной для следующих случаев:

1. $E_{x0} = 1$, $E_{y0} = 1$, $\varphi = 0, \pi$;
2. $E_{x0} = 0,5$, $E_{y0} = 1$, $\varphi = 0, \pi$;
3. $E_{x0} = 1$, $E_{y0} = 1$, $\varphi = \pi/4, \pi/2$;
4. $E_{x0} = 0,5$, $E_{y0} = 1$, $\varphi = -\pi/2$.

***Задача* 9.41**

Решите задачу 9.40 для электромагнитной волны, у которой составляющие электромагнитного поля изменяются по законам

$$E_x(y,t) = E_{x0}\cos\left(\omega\, t - \frac{2\pi}{\lambda}y - \varphi\right), \tag{9.90}$$

$$E_y(y,t) = E_{y0}\cos\left(\omega\, t - \frac{2\pi}{\lambda}y\right). \tag{9.91}$$

При распространении электромагнитной волны в среде, имеющей показатель преломления отличный от единицы, уравнения (9.90), (9.91) принимают следующий вид:

$$E_x(y,t) = E_{x0}\cos\left(\frac{2\pi}{\lambda}n_y - \omega\, t - \varphi\right), \tag{9.92}$$

$$E_y(y,t) = E_{y0}\cos\left(\frac{2\pi}{\lambda}n_y - \omega\, t\right). \tag{9.93}$$

где λ — длина электромагнитной волны в вакууме.

В некоторых веществах (например, кварц, кварцит) скорость распространения волны зависит от направления ее распространения. Такие вещества называются веществами с двойным лучепреломлением. Если показатели преломления вдоль осей OX — n_x и OY — n_y различны, то уравнения для составляющих напряженности имеют вид

$$E_x(y,t) = E_{x0}\cos\left(\frac{2\pi}{\lambda}n_x y - \omega\, t - \varphi\right), \tag{9.94}$$

$$E_y(y,t) = E_{y0}\cos\left(\frac{2\pi}{\lambda}n_y y - \omega\, t\right). \tag{9.95}$$

***Задача* 9.42**

Предположим, что на кристалл с двойным лучепреломлением падает электромагнитная волна с круговой поляризацией. Задайте показатели преломления $n_x = 9,1$, $n_y = 9,2$ и определите, при какой толщине кристалла волна, имеющая на входе круговую поляризацию, становится на выходе линейно поляризованной. Как зависит ориентация вектора E_y от толщины пластины. Что получается, если через кристалл той же толщины пропускать линейно поляризованную электромагнитную волну?

ЛИТЕРАТУРА К ГЛАВЕ 9

1. *Гетманова Е. Г., Костарев Д. Б.* Резонансные явления в системе связанных осцилляторов // Электромагнитные волны и электронные системы. 2002. Т. 6. №5.

2. *Коткин Г. Л., Сербо В. Г.* Сборник задач по классической механике. М.: Наука, 1977.

3. *Ландау Л. Д., Лифшиц Е. М.* Механика. М.: Наука, 2000.

4. *Крауфорд Ф.* Волны. М.: Наука, 1974.

5. *Гулд Х., Тобочник Я.* Компьютерное моделирование в физике. М.: Мир, 1990. С. 220.

6. *Фарлоу С.* Уравнения с частными производными для научных работников и инженеров. М.: Мир, 1985.

7. *Мешков И. Н., Чириков Б. В.* Электромагнитное поле. Ч. 2. Электромагнитные волны и оптика. Новосибирск: Наука, 1987.

Глава 10

МОДЕЛИРОВАНИЕ СИСТЕМ, СОСТОЯЩИХ ИЗ БОЛЬШОГО ЧИСЛА ЧАСТИЦ, МЕТОДОМ МОЛЕКУЛЯРНОЙ ДИНАМИКИ

10.1. ВВЕДЕНИЕ

В предыдущих разделах нами рассматривались системы, состоящие из небольшого количества частиц (за исключением главы 9). Однако большинство реальных физических систем состоят из большого числа взаимодействующих друг с другом частиц. Например, в 1 $см^{-3}$ кислорода при $T = 300$ K, $p = 1,01 \cdot 10^5$ $Па$ находится $2,5 \cdot 10^{19}$ молекул. В этой связи возникает закономерный вопрос — как решить задачу об описании поведения этой сложной многочастичной системы? В этой главе рассматривается наиболее очевидный подход к решению данной задачи – прямое решение микроскопических уравнений движения частиц, взаимодействующих друг с другом (метод молекулярной динамики). При этом наибольший интерес представляют не сами траектории, но вычисляемые с их использованием характеристики изучаемых систем, например термодинамических характеристик, вычисляемых усреднением по траекториям.

Необходимо отметить, что возможности современных компьютеров ограничивают метод молекулярной динамики (МД) по числу частиц моделируемой системы. Однако оказывается, что исследование поведения систем с использованием метода МД даже для относительно небольшого числа (от нескольких сотен до нескольких тысяч) дает достаточное количество информации для понимания наблюдаемых свойств газов, жидкостей и твердых тел.

Отметим, что вопросы, рассматриваемые в настоящей главе, изучаются в специальном разделе физики — статистической физике. Детальное рассмотрение всех тем, традиционно включаемых в известные учебные курсы [1–5], безусловно выходит за рамки нашей книги, поэтому мы ограничиваемся рассмотрением некоторых наиболее принципиальных вопросов для понимания поведения статистических систем и применения вычислительных алгоритмов.

10.2. МАТЕМАТИЧЕСКАЯ МОДЕЛЬ СТАТИСТИЧЕСКОЙ СИСТЕМЫ

Ставя перед собой целью понимание качественных свойств систем, состоящих из большого количества частиц, упростим задачу, предположив, что молекулы являются химически инертными, а их движение является классическим. Кроме того, будем

считать, что сила взаимодействия двух молекул зависит только от расстояния между ними, поэтому полная потенциальная энергия U определяется суммой энергий двух частичных взаимодействий $V(r_{ij})$:

$$U = V(r_{12}) + V(r_{13}) + \ldots + V(r_{23}) + \ldots = \sum_{i<j=1}^{N} V(r_{ij}), \tag{10.1}$$

где $V(r_{ij})$ зависит только от абсолютной величины расстояния r_{ij} между i-й и j-й частицами

$$V(r_{ij}) = V(|r_i - r_j|).$$

Модель парного взаимодействия адекватно описывает «простые» жидкости, например жидкий аргон.

Для электрически нейтральных атомов теоретически возможно, используя законы квантовой механики, получить аналитическое выражение для функции $V(r)$. Однако, во-первых, такой расчет оказывается весьма громоздким, во-вторых, для большинства задач оказывается достаточным использовать простую феноменологическую формулу, учитывающую, что при малых r сила взаимодействия между молекулами $\vec{F}(\vec{r}) = -\vec{\nabla}U(\vec{r})$ является силой отталкивания, а при больших r — силой притяжения. Отталкивание в соответствии с квантово-механическими представлениями обусловлено правилом запрета Паули [6]. Слабое притяжение при больших r главным образом обусловлено взаимной поляризацией каждого атома. Результирующая сила притяжения называется силой Ван-дер-Ваальса. Таким образом, при использовании модели двухчастичного взаимодействия задача описания поведения статистической системы сводится к выбору вида потенциала $V(\vec{r})$ и решению задачи Коши системы дифференциальных уравнений

$$m\frac{d^2\vec{r}_i}{dt^2} = -\vec{\nabla}\sum_{i<j=1}^{N} V(\vec{r}_{ij}). \tag{10.2}$$

Одной из наиболее употребительных феноменологических формул для описания потенциала межмолекулярного взаимодействия является потенциал Ленарда–Джонса

$$V(r) = 4V_0\left[\left(\frac{\sigma}{r}\right)^{12} - \left(\frac{\sigma}{r}\right)^{6}\right], \tag{10.3}$$

где σ определяет «характерную» длину потенциала, а V_0 — глубину потенциальной ямы. Продифференцировав (10.3) по r и решив уравнение

$$\frac{dU}{dr} = 0,$$

находим, что потенциал $V(r)$ достигает своего минимального значения $-V_0$ в точке $r_{\min} = 2^{\frac{1}{6}}\sigma$, $V(r) = 0$ в точке $r_0 = \sigma$.

В дальнейшем выберем в качестве единиц измерения расстояния σ и энергии V_0:

$$\tilde{r} = \frac{r}{\sigma},$$

$$\tilde{V} = \frac{V(r)}{V_0}.$$

В выбранных единицах измерения потенциал Ленарда–Джонса принимает вид

$$\tilde{V}(\tilde{r}) = 4\left[\left(\frac{1}{\tilde{r}}\right)^{12} - \left(\frac{1}{\tilde{r}}\right)^{6}\right]. \tag{10.4}$$

График безразмерного потенциала Ленарда–Джонса представлен на рис. 10.1.

Рис. 10.1. Зависимость безразмерного потенциала Ленарда–Джонса от координаты

Из рис. 10.1 видно, что потенциал является короткодействующим, то есть при $\tilde{r} \geq 2,5$

$$\tilde{V}(\tilde{r}) \cong 0.$$

Для выбора переменной T, используемой для обезразмеривания системы уравнений движения (10.2), разложим потенциал $V(r)$ в ряд Тейлора вблизи минимума потенциала ($r_{\min} = 2^{\frac{1}{6}}\sigma$). Сохранив члены ряда, пропорциональные первой и второй производным, а также приведя подобные члены, окончательно получим

$$V(\Delta r) \approx -V_0 + \frac{18 \cdot 2^{\frac{2}{3}}}{\sigma^2} V_0 \Delta r^2, \tag{10.5}$$

где $|\Delta r| \ll r_{\min}$. Сравнив (10.5) с известным выражением для потенциальной энергии гармонического осциллятора

$$E = \frac{1}{2} k x^2, \tag{10.6}$$

приходим к выводу, что величина

$$\chi = \frac{2 \cdot 18 \cdot 2^{\frac{2}{3}}}{\sigma^2} V_0 \tag{10.7}$$

является аналогом коэффициента жесткости пружины гармонического осциллятора, то есть частица, находящаяся в потенциале Ленарда–Джонса при малых смещениях от точки минимума будет совершать линейные гармонические колебания с периодом $T = 2\pi\sqrt{m/\chi}$. (Например, для жидкого аргона, у которого $V_0/k_B = 119,8K$ (k_B — постоянная Больцмана), $\sigma = 3,405 \cdot 10^{-8}$ *см*, масса $m = 6,69 \cdot 10^{-23}$ *г*, $T = 1,14 \cdot 10^{-11}$ *с*.) Далее будем использовать T для обезразмеривания времени:

$$\tilde{t} = \frac{t}{T}.$$

Перейдя в (10.2) к безразмерным переменным $\tilde{r} = r/\sigma$, $\tilde{V} = V(r)/V_0$, $\tilde{t} = t/T$ и учитывая выражение (10.7), получаем окончательное выражение для безразмерной системы уравнений движения

$$\frac{d^2\vec{\tilde{r}}_i}{d\tilde{t}^2} = -\frac{2^{\frac{7}{3}}\pi^2}{3}\vec{\nabla}\sum_{i<j=1}^{N}\tilde{V}\left(\vec{\tilde{r}}_{ij}\right). \tag{10.8}$$

Система дифференциальных уравнений (10.8), дополненная начальными условиями $\vec{\tilde{r}}_i(0)$, $\vec{\tilde{v}}_i(0)$ является математической моделью рассматриваемой статистической системы.

10.3. ЧИСЛЕННЫЙ АЛГОРИТМ РЕШЕНИЯ СИСТЕМЫ УРАВНЕНИЙ ДВИЖЕНИЯ

После составления математической модели системы, состоящей из большого числа взаимодействующих частиц, следует выбрать численный алгоритм решения, от правильного выбора которого напрямую зависит точность решения. В главе 9 при численном решении уравнений движения мы убедились, что контроль за устойчивостью численного решения можно осуществлять, следя за полной энергией системы, величина которой в идеальном случае должна оставаться постоянной. Отметим, что стандартные функции MATLAB, использованные нами в главе 9, при моделировании колебаний цепочек в данной задаче оказываются неработоспособными вследствие необходимости использования специальных граничных условий (см. более подробно ниже в настоящем разделе). Отмеченные обстоятельства определяют необходимость разработки собственных пользовательских функций, реализующих решение уравнений движения в методе молекулярной динамики. Анализ алгоритмов низкого порядка точности, таких как алгоритм Эйлера и алгоритм Эйлера—Кроммера показывает, что данные алгоритмы не могут обеспечить сохранение энергии на временных интервалах, рассматриваемых при моделировании молекулярной динамики. В этих условиях приходится применять вычислительные алгоритмы, имеющие более высокий порядок точности, одним из которых является алгоритм Верле.

Проиллюстрируем суть данного алгоритма, следуя [7], на примере решения системы уравнений одномерного движения частицы

$$\frac{dv}{dt} = a, \tag{10.9}$$

$$\frac{dx}{dt} = v. \tag{10.10}$$

Запишем разложение зависимостей

$$x_{n+1} \equiv x\left(t_n + \Delta t\right)$$

и

$$v_{n+1} \equiv v\left(t_n + \Delta t\right)$$

в ряд Тейлора

$$x_{n+1} = x_n + \dot{x}\left(t_n\right)\Delta t + \frac{1}{2}\ddot{x}\left(t_n\right)\left(\Delta t\right)^2 + O\left[\left(\Delta t\right)^3\right], \tag{10.11}$$

$$v_{n+1} = v_n + \dot{v}\left(t_n\right)\Delta t + O\left[\left(\Delta t\right)^2\right]. \tag{10.12}$$

Замечая, что

$$\dot{x}\left(t_n\right) = v_n,$$

$$\ddot{x}\left(t_n\right) = \dot{v}\left(t_n\right) = a_n,$$

перепишем (10.11), (10.12) в следующем виде:

$$x_{n+1} = x_n + v_n\Delta t + \frac{1}{2}a_n(\Delta t)^2 + O\left[\left(\Delta t\right)^3\right], \tag{10.13}$$

$$v_{n+1} = v_n + a_n\Delta t + O\left[\left(\Delta t\right)^2\right]. \tag{10.14}$$

По аналогии с (10.13), (10.14) запишем разложение в ряд Тейлора для $x_{n-1} \equiv \equiv x\left(t_n - \Delta t\right)$:

$$x_{n-1} = x_n - v_n\Delta t + \frac{1}{2}a_n(\Delta t)^2. \tag{10.15}$$

Сложив (10.13) и (10.15), получим

$$x_{n+1} + x_{n-1} = 2x_n + a_n(\Delta t)^2, \tag{10.16}$$

откуда

$$x_{n+1} = 2x_n - x_{n-1} + a_n(\Delta t)^2. \tag{10.17}$$

Вычитая (10.15) из (10.13), окончательно получим

$$v_n = \frac{x_{n+1} - x_{n-1}}{2\Delta t}. \tag{10.18}$$

Глобальная погрешность алгоритма Верле, реализуемого формулами (10.17), (10.18), имеет третий порядок для координаты и второй порядок для скорости. Отметим, что скорость не участвует в интегрировании уравнений движения, поэтому в литературе, посвященной численным методам, данный алгоритм называется неявной симметричной разностной схемой. Очевидный недостаток неявной разностной схемы состоит в том, что он не является самостартующим, поэтому приходится использовать другой алгоритм для получения нескольких первых точек.

Отмеченный недостаток можно устранить, добавив и вычтя из обеих частей равенства (10.17) величину $x_{n+1}/2$:

$$\begin{aligned} x_{n+1} &= x_n + \frac{1}{2}\left(x_{n+1} - x_{n-1}\right) - \frac{1}{2}x_{n-1} + \frac{1}{2}x_{n-1} + x_n + a_n(\Delta t)^2 = \\ &= x_n + v_n\Delta t - \frac{1}{2}\left(x_{n+1} - 2x_n + x_{n-1}\right) + a_n(\Delta t)^2. \end{aligned} \tag{10.19}$$

Из (10.16) найдем

$$a_n = \frac{x_{n+1} - 2x_n + x_{n-1}}{(\Delta t)^2}, \tag{10.20}$$

поэтому (10.19) можно записать в виде

$$x_{n+1} = x_n + v_n \Delta t + \frac{1}{2} a_n (\Delta t)^2. \tag{10.21}$$

Поступая аналогично, перепишем (10.18) для v_{n+1} и (10.13) для x_{n+2}:

$$v_{n+1} = \tfrac{1}{2\Delta t}\left(x_{n+2} - x_n\right), \tag{10.22}$$

$$x_{n+2} = 2x_{n+1} - x_n + a_{n+1}(\Delta t)^2 = x_{n+1} + v_{n+1}\Delta t, + \tfrac{1}{2} a_{n+1}(\Delta t)^2 \tag{10.23}$$

соответственно.

Подставив (10.22) в (10.23), получаем

$$\begin{aligned} v_{n+1} &= \frac{1}{2\Delta t}\left(x_n + v_n \Delta t + \frac{1}{2} a_n (\Delta t)^2 + v_{n+1}\Delta t + \frac{1}{2} a_{n+1}(\Delta t)^2 - x_n\right) = \\ &= \frac{1}{2} v_n + \frac{1}{2} v_{n+1} + \frac{1}{4}\left(a_n + a_{n+1}\right)(\Delta t)^2. \end{aligned} \tag{10.24}$$

Далее из (10.24), после очевидных выкладок, окончательно получим

$$v_{n+1} = v_n + \frac{1}{2}\left(a_{n+1} + a_n\right)\Delta t. \tag{10.25}$$

Вычислительная схема, задаваемая выражениями (10.21), (10.25), математически эквивалентна алгоритму Верле, описанному выше. Данная схема, называемая скоростной формой алгоритма Верле, является самостартующей, а потому не требует использования каких-либо дополнительных вычислительных алгоритмов. Описание других вычислительных схем, используемых для решения уравнений движения, приведены в [8].

При использовании метода МД для моделирования поведения газов и жидкостей, как правило, предполагается, что рассматриваемая система находится в некоторой кубической ячейке — МД-ячейке. Будем считать, что МД-ячейка имеет линейный размер L, ее объем $V = L^3$. Использование кубической решетки порождает шесть нежелательных поверхностей. Частицы, отраженные от этих поверхностей, будут возвращаться внутрь ячейки, поэтому грани ячейки будут вносить значительный вклад в макроскопические характеристики системы, особенно для систем с малым количеством частиц. Для уменьшения описываемого эффекта принято вводить периодические граничные условия [8] (рис. 10.2), математическая формулировка которых для любой наблюдаемой величины A имеет вид

$$A\left(\vec{r}\right) = A\left(\vec{r} + \vec{n}L\right), \tag{10.26}$$

где $\vec{n} = (n_1, n_2, n_3)$, а n_1, n_2, n_3 — произвольные целые числа.

Данный алгоритм имеет следующую вычислительную реализацию: при пересечении частицей грани основной ячейки, она возвращается в ячейку через противоположную грань с той же скоростью. Введением периодических граничных условий устраняется влияние граней и вводится квазибесконечный объем для более точного описания

Рис. 10.2. Пример периодических граничных условий в двумерном случае. Правило ближайшей частицы означает, что расстояние между частицами 1 и 2 определяется длиной вектора, обозначенного двунаправленной стрелкой

макроскопической системы, то есть МД-ячейка оказывается «встроенной» в среду. Каждая компонента радиуса-вектора трансляции является числом между нулем и L. Для i-й частицы, находящейся в точке с радиусом-вектором $\vec{r}_i$, имеются отображения частицы в точках с радиусами-векторами $\vec{r}_i + \vec{n}L$, где $\vec{n}$ — целочисленный вектор.

Для выбранных граничных условий потенциальная энергия принимает следующий вид:

$$V(\vec{r}_1, \ldots, \vec{r}_N) = \sum_{i<j} V(\vec{r}_{ij}) + \sum_{\vec{n}} \sum_{i<j} V(|\vec{r}_{-i} - \vec{r}_{-j} + \vec{n}L|). \qquad (10.27)$$

Для того чтобы избежать вычисления бесконечной суммы в (10.27), принимается следующее правило [8]: расстояние $|\vec{r}_{ij}|$ между частицами, расположенными в точках с радиусами-векторами $\vec{r}_i$, $\vec{r}_j$ соответственно, определяется как

$$|\vec{r}_{ij}| = \min(|\vec{r}_{-i} - \vec{r}_{-j} \pm \vec{n}L|)$$

по всем $\vec{n}$. Данное правило означает, что частица, находящаяся в базисной ячейке взаимодействует с каждой из $N-1$ частицей в базисной ячейке или со своими ближайшими отображениями (рис. 10.2). Важно понимать, что использование данного правила приводит к «обрезанию» потенциала на расстояниях

$$r_c > \frac{L}{2}.$$

Это приводит к потере фонового вклада удаленных частиц, поэтому для устранения эффекта конечности системы значение L должно выбираться достаточно большим, чтобы силы, действующие на расстояниях больших $L/2$, были пренебрежимо малы. Отметим, что более правильный подход состоит в учете взаимодействия каждой частицы со своим отображением. Ссылки на оригинальные работы, посвященные данному подходу и описанию вычислительных алгоритмов, его реализующим, приведены в [8].

Сформулируем окончательно алгоритм метода МД:

1. Задать число частиц системы N.

2. Задать начальную конфигурацию системы (совокупность координат $\vec{r}_i(0)$ и скоростей частиц $\vec{v}_i(0)$).

3. Задать h — шаг интегрирования системы дифференциальных уравнений (10.8).

4. Задать Nh — число шагов, в которых вычисляются решения системы дифференциальных уравнений (10.8).

5. Вычислить в соответствии с (10.21), (10.25) и учетом периодических граничных условий значения координат $\vec{r}_i$ и скоростей $\vec{v}_i$, t_n, $i = 0, 1, \ldots, N$ в последовательные моменты времени t_n, $n = 0, 1, \ldots, Nh$.

10.4. МОДЕЛИРОВАНИЕ МЕТОДОМ МОЛЕКУЛЯРНОЙ ДИНАМИКИ СИСТЕМЫ, СОСТОЯЩЕЙ ИЗ БОЛЬШОГО ЧИСЛА ЧАСТИЦ

Рассматриваемая нами статистическая система является детерминированной, поскольку для описания ее поведения решается задача Коши системы линейных дифференциальных уравнений с постоянными коэффициентами. При этом получаемые решения напрямую зависят от начальных условий (начальной конфигурации системы). Отметим, что их правильный выбор оказывается далеко не простой задачей (например, заранее совершенно не очевидно, как выбрать начальную конфигурацию, чтобы изучаемая система вела себя как жидкость с заданной температурой), поэтому сначала мы обсудим особенности эволюции статистической системы из произвольных начальных конфигураций. Один из возможных вариантов задания начальных условий — размещение частиц в узлах некоторой прямоугольной сетки (размер которой, как очевидно, должен быть меньше размера МД-ячейки) и задание векторов их скоростей случайным образом, например с помощью генератора случайных чисел с равномерным законом распределения. Данный подход использован ниже в задаче моделирования статистической системы методом МД.

Для решения поставленной задачи оказывается удобным сначала создать m-файлы, содержащие описания:

1) функции, возвращающей начальную конфигурацию системы (файл `Init.m`);

2) функции, возвращающей мгновенные ускорения каждой частицы системы и мгновенное значение потенциальной энергии (файл `Acc.m`);

3) функции, возвращающей значения координат, составляющих скорости и ускорений вдоль соответствующих координатных осей (файл `Verlet.m`);

4) функции, возвращающей составной массив, содержащий значения координат, проекций скоростей и ускорений на соответствующие координатные оси в узлах временной сетки (файл `Mol_din.m`).

```
% листинг функции Init.m
function z=Init(Lx,Ly,Nx,Ny,Vmax)
% функция, возвращающая начальную
% конфигурацию системы
% Lx - ширина МД-ячейки
% Ly - высота МД-ячейки
% Nx - ширина начальной ячейки
% Ny - высота начальной ячейки
% Vmax - максимальное значение скорости

Pos_row=Ly/(Ny+1);
Pos_col=Lx/(Nx+1);
N=Nx*Ny; % число частиц системы
i=1;
for Rows=1:Ny
  for Col=1:Nx
    % задание начальных координат частиц
    x(i)=Pos_col*Col/2;
    y(i)=Pos_row*Rows;
    % задание начальных скоростей частиц
    Vx(i)=Vmax*(2*rand(1)-1);
    Vy(i)=Vmax*(2*rand(1)-1);
    i=i+1;
  end;
end;

% вычисление проекций скорости
% центра масс системы
Vx_full=mean(Vx);
Vy_full=mean(Vy);
i=1:N;
Vx(i)=Vx(i)-Vx_full;
Vy(i)=Vy(i)-Vy_full;

% возврат матрицы, содержащей
% начальные значения координат и
% проекций скоростей частиц
% статистической системы
z=cat(2,x',y',Vx',Vy');

% листинг файла Acc.m
function z=Acc(R_V,Lx,Ly,N)
% функция, возвращающая мгновенные
% ускорения каждой частицы
% системы и мгновенное значение
```

```
% потенциальной энергии
% R_V - матрица, возвращенная функцией Init
% Lx - ширина МД-ячейки
% Ly - высота МД-ячейки
% N - число частиц системы
% инициализация переменных
Ep=0;
for i=1:N
  Ax(i)=0;
  Ay(i)=0;
  Pe(i)=0;
end;

for i=1:N-1
  for j=i+1:N
       Dx=R_V(i,1)-R_V(j,1);
       % проверка граничных условий
       if abs(Dx)>Lx/2
          Dx=Dx-sign(Dx)*Lx;
      end;
      Dy=R_V(i,2)-R_V(j,2);
      % проверка граничных условий
      if abs(Dy)>Ly/2
          Dy=Dy-sign(Dy)*Ly;
       end;
      % вычисление проекции ускорения на ось OX
      Ax(i)=Ax(i)+Fx(Dx,Dy);
      Ax(j)=Ax(j)-Fx(Dx,Dy); % третий закон Ньютона
      % вычисление проекции ускорения на ось OY
      Ay(i)=Ay(i)+Fy(Dx,Dy);
      Ay(j)=Ay(j)-Fy(Dx,Dy); % третий закон Ньютона
      r = (Dx^2+Dy^2).^0.5;
      % вычисление потенциальной энергии
     Ep=Ep+4*(1/r.^12-1/r.^6);
  end;
end;
Pe(N)=Ep;
z=cat(2,Ax',Ay',Pe');

function z1=Fx(x,y)
% проекция силы, действующей на
% частицу вдоль оси OX
z1=-2^(7/3)*(pi^2)/3*x.*(-2/(x.^2+y.^2).^7+...
      1/(x.^2+y.^2).^4);
```

```
function z2=Fy(x,y)
% проекция силы, действующей на
% частицу вдоль оси OY
z2=-2^(7/3)*pi^2/3*y.*(-2/(x.^2+y.^2).^7+...
       1/(x.^2+y.^2).^4);

% листинг файла Verlet.m
function z=Verlet(R_V,Lx,Ly,N,dt)
% функция, возвращающая значения
% координат, составляющих скорости и
% ускорений вдоль соответствующих
% координатных осей
% R_V - матрица, возвращенная функцией Init
% Lx  - ширина МД-ячейки
% Ly - высота МД-ячейки
% N - число частиц системы
% dt - шаг интегрирования

for i=1:N
     % новое положение частицы
     x(i)=R_V(i,1)+R_V(i,3)*dt+R_V(i,5)*(dt^2)/2;
     y(i)=R_V(i,2)+R_V(i,4)*dt+R_V(i,6)*(dt^2)/2;
     % перемещение при необходимости частицы
     % в центральную ячейку
 if x(i)<0
    x(i)=x(i)+Lx;
  end;
  if x(i)>Lx
    x(i)=x(i)-Lx;
  end;
  if y(i)<0
    y(i)=y(i)+Ly;
  end;
  if y(i)>Ly
    y(i)=y(i)-Ly;
  end;
end;
% изменение скорости с использованием
% "старого" ускорения
for i=1:N
  Vx(i)=R_V(i,3)+0.5*R_V(i,5)*dt;
  Vy(i)=R_V(i,4)+0.5*R_V(i,6)*dt;
end;
% вычисление "нового" ускорения
R_Vnew=cat(2,x',y',Vx',Vy');
Accnew=Acc(R_Vnew,Lx,Ly,N);
```

```
% вычисление скорости с использованием
% "нового" ускорения
for i=1:N
     Vx(i)=Vx(i)+0.5*Accnew(i,1)*dt;
    Vy(i)=Vy(i)+0.5*Accnew(i,2)*dt;
end;
z=cat(2,x',y',Vx',Vy',Accnew);

function z1=Fx(x,y)
% проекция силы, действующей на частицу
% вдоль оси OX
z1=-2^(7/3)*(pi^2)/3*x.*(-2/(x.^2+y.^2).^7+1/(x.^2+y.^2).^4);

function z2=Fy(x,y)
% проекция силы, действующей на частицу
% вдоль оси OY
z2=-2^(7/3)*pi^2/3*y.*(-2/(x.^2+y.^2).^7+1/(x.^2+y.^2).^4);

% листинг файла Mol_din.m
function z=Mol_din(Lx,Ly,N,Ni,dt,A)
function z=Mol_din(Lx,Ly,N,Ni,dt,A)
% функция, возвращающая составной массив,
% содержащий значения координат,
% проекций скоростей и ускорений
% на соответствующие координатные оси
% в узлах временной сетки
% Lx - ширина МД-ячейки
% Ly - высота МД-ячейки
% N - число частиц системы
% Ni - число МД-шагов
% dt - шаг интегрирования уравнений движения
% A - массив, возвращенный функцией Init

Accold=Acc(A,Lx,Ly,N);
A2=cat(2,A,Accold);
A3=cat(3,A2);
k=0;
while k<Ni
     A2=Verlet(A2,Lx,Ly,N,dt);
     A3=cat(3,A3,A2);
     k=k+1;
end;
z=A3;
```

Далее необходимо выполнить следующую последовательность команд, сохраненную нами в файле `Glava10_1.m`:

```
% листинг файла Glava10_1.m
% задание размера начальной ячейки
Nx=5;
Ny=5;
N=Nx*Ny; % число частиц системы
% задание размера МД-ячейки
Lx=8;
Ly=8;

% инициализация начальной конфигурации
Vmax=0.2; % максимальное значение начальной скорости
A=Init(Lx,Ly,Nx,Ny,Vmax);

% задание параметров метода МД
Ni=10^4; % число МД-шагов
dt=1*10^-4; % величина МД-шага

% вызов функции, реализующей метод
% молекулярной динамики
md=Mol_din(Lx,Ly,N,Ni,dt,A);

% визуализация мгновенных конфигураций
% системы в выбранные моменты времени (рис. 10.3-10.6)
figure
plot(md(:,1,1),md(:,2,1),'oK','MarkerSize',3);
axis([0 Lx 0 Ly])

figure
plot(md(:,1,501),md(:,2,501),'oK','MarkerSize',3);
axis([0 Lx 0 Ly])

figure
plot(md(:,1,5001),md(:,2,5001),'oK','MarkerSize',3);
axis([0 Lx 0 Ly])

figure
plot(md(:,1,10001),md(:,2,10001),'oK','MarkerSize',3);
axis([0 Lx 0 Ly])

% вычисление числа частиц
% в левой половине ящика
for i=1:Ni+1
    N1(i)=0;
    tmp=md(:,1,i);
    for j=1:N
```

```
        if tmp(j)<=Lx/2
           N1(i)=N1(i)+1;
        end;
    end;
end;
% визуализация зависимости числа частиц,
% находящихся в левой половине
% ящика от времени (рис. 10.7)
i=1:Ni+1;
L(i)=12.5;
figure
plot((i-1)*dt,N1,'k',i*dt,L,'--k');
axis([0 Ni*dt 0 30])

% вычисление мгновенных значений
% кинетической и потенциальной
% энергий системы
for i=1:Ni+1
    Sum=0;
    tmpVx=md(:,3,i);
    tmpVy=md(:,4,i);
    for j=1:N
         Sum=Sum+0.5*(tmpVx(j)^2+tmpVy(j)^2);
    end;
     Ek(i)=Sum;
     Ep(i)=md(N,7,i);
end;

% визуализация зависимостей кинетической,
% потенциальной и полной
% энергий, приходящихся на одну частицу
% от времени (рис. 10.8-10.10)
i=1:Ni+1;

figure
plot((i-1)*dt,Ek/N,'k') % кинетическая энергия

figure
plot((i-1)*dt,Ep/N,'k') % потенциальная энергия

figure
plot((i-1)*dt,(Ek+Ep)/N,'k') % полная энергия
```

Результаты выполнения описанной последовательности команд представлены на рис. 10.3—10.10.

Рис. 10.3. Начальная конфигурация статистической системы

Рис. 10.4. Конфигурация статистической системы в момент времени $t = 0,05$

Рис. 10.5. Конфигурация статистической системы в момент времени $t = 0,5$

Рис. 10.6. Конфигурация статистической системы в момент времени $t = 1,0$

Рис. 10.7. Зависимость числа частиц, находящихся в левой половине ящика, от времени ($Nleft = Nleft(t)$)

Рис. 10.8. Зависимость кинетической энергии, приходящейся на одну частицу, от времени ($Ek = Ek(t)$)

Рис. 10.9. Зависимость потенциальной энергии, приходящейся на одну частицу, от времени ($Ep = Ep(t)$)

Рис. 10.10. Зависимость полной энергии, приходящейся на одну частицу, от времени ($E = E(t)$)

Анализ зависимостей, представленных на рис. 10.7—10.10, показывает, что с течением времени система стремится к состоянию равновесия (процесс релаксации), в котором примерно одинаковыми остаются число частиц в левой и правой половинах МД-ячейки и полная энергия системы. (Отметим, что поскольку значения начальной скорости задаются с помощью генератора случайных чисел, данные зависимости, полученные при пересчете документа, будут всегда отличаться от зависимостей, представленных на рис. 10.7—10.10, однако качественно они должны вести себя подобным образом.)

***Задача* 10.1**[1]

Используя зависимости $Nleft = Nleft(t)$, $Ek = Ek(t)$, $Ep = Ep(t)$, $E = E(t)$, оцените:

1) *время релаксации статистической системы к состоянию равновесия;*

2) *средние значения кинетической, потенциальной и полной энергий системы в состоянии равновесия;*

3) *оцените величину флуктуаций кинетической, потенциальной и полной энергий системы в состоянии равновесия.*

***Задача* 10.2**

Опишите движение отдельной частицы в процессе релаксации, проанализировав зависимости $x(t)$, $y(t)$, $R(t) = \sqrt{x(t)^2 + y(t)^2}$, $y = y(x)$, $v_x = v_x(t)$, $v_y = v_y(t)$, $v_y = f(v_x)$.

Постройте зависимости $x(t) - x(0)$, $y(t) - y(0)$, $R(t) - R(0)$, $(R(t) - R(0))^2$.

[1] Во всех задачах данной главы, где начальные параметры статистической системы не оговорены специально, они соответствуют параметрам, указанным в тексте главы при описании соответствующх файлов.

***Задача* 10.3**

Опишите движение отдельной частицы после достижения системой в момент времени t_γ равновесного состояния:

1. *Постройте зависимости* $x(t)-x(t_\gamma)$, $y(t)-y(t_\gamma)$, $R(t)-R(t_\gamma)$, $(R(t)-R(t_\gamma))^2$.
2. *Вычислите средние значения* $\langle x(t)-x(t_\gamma)\rangle$, $\langle y(t)-y(t_\gamma)\rangle$, $\langle R(t)-R(t_\gamma)\rangle$.
3. *Считая, что среднеквадратическое отклонение частицы от начального положения меняется во времени по линейному закону* $\langle R(t)-R(t_\gamma) = a(t-t_\gamma)\rangle$, *оцените с помощью метода наименьших квадратов значение коэффициента* a.
4. *Проведите аналогичный анализ для другой частицы и сравните полученные результаты.*

***Задача* 10.4**

Рассмотрите обратный во времени процесс, выбрав в качестве начальных значений координат частиц, соответствующие координаты из последней вычисленной конфигурации статистической системы $Ni(X_i(0) = x_i(dt\cdot Ni)$, $Y_i(0) = y_i(dt\cdot Ni)$, $i = 0,1,\ldots,N)$, *в качестве начальных скоростей соответствующие значения скоростей из последней конфигурации статистической системы, взятых с противоположным знаком* ($Vx_i(0) = -vx_i(dt\cdot Ni)$, $Vy_i(0) = -vy_i(dt\cdot Ni)$). *Каково состояние статистической системы в момент времени* $dt\cdot Ni$?

***Задача* 10.5**

Используя описанный выше документ, исследуйте поведение статистической системы с большим количеством частиц. Как зависят от числа частиц в состоянии равновесия:

1) *время достижения равновесия;*
2) *величина флуктуаций числа частиц в левой половине МД-ячейки* ($Nleft(t) - N/2$);
3) *величина флуктуаций кинетической энергии, приходящейся на одну частицу* $\langle(Ek-\overline{Ek})^2\rangle$ *(здесь угловые скобки означают усреднение по времени,* $\overline{Ek}$ *— среднее значение кинетической энергии);*
4) *величина флуктуаций потенциальной энергии, приходящейся на одну частицу* $\langle(Ep-\overline{Ep})^2\rangle$;
5) *величина флуктуаций полной энергии, приходящейся на одну частицу* $\langle(E-\overline{E})^2\rangle$?

Примечание: *основной величиной, определяющей точность полученных решений, является шаг по времени* h. *Однако уменьшение шага интегрирования при фиксированном интервале времени, на котором исследуется динамическая система, приводит к пропорциональному увеличению общего времени*

вычислений. Общего критерия выбора шага h, к сожалению, не существует, поэтому его величина определяется путем эмпирического подбора, который требует выполнения следующей последовательности действий:

1) *выбор определенного шага;*

2) *расчет;*

3) *анализ достоверности полученных результатов, если полученные результаты недостоверны (поведение соответствующих величин качественно отличается от зависимостей, представленных на рис. 10.7–10.10), то выбирается меньший шаг и повторяются п. 2, 3).*

В статистической физике для характеристики систем, состоящих из большого числа частиц, используют функции распределения по скорости $P(v)$ и энергии $P(E)$. Функция $P(v)$ определяет вероятность $P(v)\Delta v$ обнаружить частицы, скорость которых находится в интервале $[v; v + \Delta v]$.

Функция $P(E)$ определяет вероятность $P(E)\Delta E$ обнаружить частицы, кинетическая энергия которых находится в интервале $[E; E + \Delta E]$. Для вычисления функции распределения по скорости в равновесном состоянии может быть использован следующий алгоритм:

1. Определить минимальное $v_{\min}$ и максимальное $v_{\max}$ значения скорости на временном интервале $[T_1, T_2]$, на котором система находится в равновесном состоянии.

2. Задать Nh — число отрезков, на которые разбивается отрезок

$$[v_{\min}, v_{\max}].$$

3. Подсчитать число частиц, попадающих на отрезки

$$[v_{\min} + \Delta v \cdot i;\ v_{\min} + \Delta v \cdot (i+1)]\,,$$

   где

$$\Delta v = (v_{\max} - v_{\min})/Nh,\ \ i = 0, 1, \ldots, Nh - 1$$

   в каждый момент времени из интервала $[T_1, T_2]$ (вычисление мгновенной функции распределения).

4. Усреднить мгновенные функции распределения.

Описанный выше алгоритм реализован автором в виде m-функции, которая сохранена на жестком диске в файле `Distr.m`.

```
% листинг файла Distr.m
function z=Distr(md,N,Tstart,dt,Flag,Nh)
% функция, возвращающая вероятность
% нахождения проекций скорости
% частицы в интервале
```

```
% md - массив, возвращенный функцией Mol_din
% N - число частиц системы
% Tstart - момент времени, начиная
% с которого проводится вычисление
% мгновенных распределений
% dt - шаг интегрирования уравнений движения
% Flag - строковая переменная,
% значение которой определяет выбор
% оси, вдоль которой
% вычисляется вероятность распределения
% Nh - число интервалов
% разбиения интервала скоростей

Np=length(md); % число МД-шагов
k=1;
if Flag=='x'
      L=3;
  else
      L=4;
end;
Nstart=round(Tstart/dt);
V=md(:,L,Nstart);
% создание матрицы,
% содержащей мгновенные значения проекций
% скорости частиц
for i=Nstart+1:Np
     z1=md(:,L,i);
     V=cat(2,V,z1);
end;
% нахождение минимального и максимального
% значений скорости частиц
V1=min(V);
Vmin=min(V1);
V2=max(V);
Vmax=max(V2);
dv=(Vmax-Vmin)/Nh; % длина интервала
                   % по скорости
m=1:Nh+1;
inter(m)=Vmin+dv*(m-1); % координаты
                        % узлов сетки
g(m)=0;
% вычисление мгновенных распределений
k=1;
for i=Nstart:Np
    v=V(:,k);
```

```
    k=k+1;
    g=g+hist(v,inter);
end;
% вычисление среднего распределения
g=g/((Np-Nstart)*Nh);
z=cat(2,inter',g');
```

Отметим, что при использовании функций `min`, `max` для матриц, возвращаемый результат является вектором, содержащим соответственно минимальные и максимальные значения по каждому столбцу матрицы. Данные функции, будучи примененными к вектору, возвращают скаляр — минимальное/максимальное значение. Для подсчета числа частиц, попадающих в заданные интервалы, мы используем специализированную функцию MATLAB `hist`, которая зависит от двух векторов: первый содержит значения анализируемой последовательности `y`, второй — координаты интервалов разбиения отрезка $[y\min, y\max]$.

Для вычисления распределения частиц по проекциям скорости на координатные оси необходимо выполнить следующую последовательность команд, сохраненную нами в файле `Glava10_2.m`:

```
% листинг файла Glava10_2.m

% размер начальной ячейки
Nx=5;
Ny=5;
N=Nx*Ny; % число частиц системы
% размер МД-ячейки
Lx=8;
Ly=8;
Vmax=0.2; % максимальное значения
          % начальной скорости

% инициализация
% начальной конфигурации
A=Init(Lx,Ly,Nx,Ny,Vmax);
Ni=10^4; % число МД-шагов
dt=1*10^-4; % величина МД-шага
md=Mol_din(Lx,Ly,N,Ni,dt,A);
Tstart=0.25;

Nh=10; % число интервалов
z=Distr(md,N,Tstart,dt,'x',Nh);
z1=Distr(md,N,Tstart,dt,'y',Nh);

% визуализация
% распределений частиц
% по проекциям скорости (рис. 10.11-10.12)
```

```
figure
bar(z(:,1),z(:,2));
colormap white

figure
bar(z1(:,1),z1(:,2));
colormap white
```

Результаты выполнения описанной последовательности команд представлены на рис. 10.11, 10.12.

Рис. 10.11. Вероятность нахождения проекции скоростей частиц на ось OX в интервале скоростей $[v, v + \Delta v]$

Рис. 10.12. Вероятность нахождения проекции скоростей частиц на ось OY в интервале скоростей $[v, v + \Delta v]$

***Задача* 10.6**

Создайте функцию, возвращающую вероятность обнаружить модуль скорости частицы в интервале $[v, v + \Delta v]$. *и вычислите:*

1) *наиболее вероятное значение скорости;*

2) *среднее значение скорости.*

Сравните вероятности обнаружить проекцию скорости на соответствующие оси и вероятность нахождения модуля скорости в заданном диапазоне скоростей.

***Задача* 10.7**

Создайте функцию, возвращающую вероятность обнаружить кинетическую энергию частицы в интервале $[E, E + \Delta E]$. *Сравните вероятности обнаружить проекцию скорости на соответствующие оси, вероятность нахождения модуля скорости в заданном диапазоне скоростей с вероятностью обнаружить кинетическую энергию частицы в интервале* $[E, E + \Delta E]$.

10.5. ОЦЕНКА МАКРОСКОПИЧЕСКИХ ХАРАКТЕРИСТИК СТАТИСТИЧЕСКОЙ СИСТЕМЫ

В предыдущем разделе мы характеризовали состояние газа числом частиц в левой части половине МД-ячейки. Известно, что в термодинамике равновесное состояние характеризуется такими параметрами, как абсолютная температура T, среднее давление P, объем V, полная энергия.

Для определения температуры можно использовать теорему о равнораспределении, согласно которой на каждую переменную, входящую в выражение для энергии равновесной классической системы, приходится по $T/2$ в его энергии. В том случае, если кинетическая энергия частиц значительно больше средней потенциальной энергии их взаимодействия можно считать, что внутренняя энергия системы обусловлена только движением частиц, поэтому

$$\frac{d}{2}NT = \sum_i \left\langle \frac{1}{2}mv_i^2 \right\rangle, \tag{10.28}$$

где d — размеренность системы, скобки $\langle\,\rangle$ означают усреднение по времени.

Отметим, что в (10.28) температура T измеряется в энергетических единицах, перевод которых в градусы Кельвина осуществляется делением температуры T на постоянную Больцмана $k_B = 1,38 \cdot 10^{-23}$ *Дж/К*.

***Задача* 10.8**

Оцените возможность использования модели идеального газа для описания рассматриваемой системы. Дополните описанный выше документ функцией, позволяющей вычислять температуру равновесной статистической системы в соответствии с (10.28).

***Задача* 10.9**

Выражение (10.28) устанавливает связь между средним значением кинетической энергии, вычисляемой как сумма средних значений кинетических энергий частиц (среднее по траектории). Можно предложить альтернативный способ вычисления температуры

$$\frac{d}{2}NT = \left\langle \sum_i \frac{1}{2}mv_i^2 \right\rangle. \tag{10.29}$$

1. *Дополните приведенный выше листинг фрагментом программного кода, позволяющим вычислять температуру равновесной статистической системы согласно (10.29).*

2. *Сравните найденное значение температуры с результатом, полученным в предыдущей задаче. Объясните полученный результат.*

Способ вычисления давления рассматриваемой статистической системы вытекает из теоремы вириала, в соответствии с которой

$$PV = NT + \frac{1}{d}\left\langle \sum_i^N \vec{r}_i\vec{F}_i \right\rangle, \tag{10.30}$$

где $\vec{r}_i$ — радиус-вектор i-й частицы, $\vec{F}_i$ — полная сила, действующая на i-ю частицу со стороны остальных частиц (здесь сумма берется по всем N частицам). Вывод формулы (10.30) приведен в [7, С. 179–180].

Для вычисления давления в соответствии с (10.30):

1) необходимо вычислить мгновенные значения давления;

2) провести усреднение на временном интервале, на котором система является равновесной.

Далее мы приводим листинг файла `Virial.m`, содержащего описание функции, возвращающей мгновенные значения давления.

```
% листинг файла Virial.m
function z=Virial(md,E,Lx,Ly)
% функция, возвращающая
% мгновенные значения давления
% в соответствии с (10.30)
% md - массив, возвращенный функцией Mol_din
% E - вектор, содержащий
% мгновенные значения энергии
% Lx - ширина МД-ячейки
% Ly - высота МД-ячейки

K=length(md); % число МД-шагов
N=size(md,1); % число частиц системы
% вычисление мгновенных значений
% энергии в соответствии с (10.30)
for i=1:K
   V1=E(i)/(Lx*Ly);
   V2=0;
   for j=1:N
     V2=V2+md(j,1,i)*md(j,5,i)+md(j,2,i)*md(j,6,i);
   end;
   V(i)=V1+0.5*V2/(Lx*Ly*N);
end;
z=V;
```

Для вычисления и визуализации зависимости мгновенных значений энергии от времени необходимо выполнить следующую последовательность команд, сохраненную нами в файле `Glava10_3.m`:

```
% листинг файла Glava10_3.m

% вычисление мгновенных значений
% кинетической энергии
for i=1:Ni+1
    Sum=0;
```

```
    tmpVx=md(:,3,i);
    tmpVy=md(:,4,i);
    for j=1:N
        Sum=Sum+0.5*(tmpVx(j)^2+tmpVy(j)^2);
    end;
    Ek(i)=Sum;
end;
% вычисление и визуализация
% мгновенных значений давления
Z=Virial(md,Ek,Lx,Ly);
i=1:Ni+1;
plot((i-1)*dt,Z,'K')
```

Результаты выполнения описанной выше последовательности команд представлены на рис. 10.13.

Рис. 10.13. Зависимость мгновенных значений давления от времени

***Задача* 10.10**

Задайте функцию, позволяющую определять средние значения давления. Оцените точность вычисления давления. Исследуйте, как зависит величина флуктуаций давления от числа частиц статистической системы.

Одна из проблем метода МД — проблема получения равновесных конфигураций с заранее заданной температурой, знание которой необходимо для оценки ряда ее термодинамических характеристик, так как данная величина зависит от целого ряда параметров системы: начальной энергии, числа частиц, размеров системы. Для преодоления отмеченных трудностей обычно используют перенормировку кинетической энергии — процедуру с физической точки зрения эквивалентную добавлению в систему извне энергии или ее отводу из системы. Для ее реализации может быть использован следующий алгоритм:

1. Задание требуемой температуры T_f.

2. Задание начальной равновесной конфигурации.
3. Задание числа шагов по времени Nm, через которые производится нормировка скорости.
4. Интегрирование уравнений движения на временном интервале

$$[dt \cdot j; dt\,(j + Nm)]\,.$$

5. Вычисление средней кинетической энергии

$$Ek = \left\langle \frac{1}{N} \sum_{i=1}^{N} \frac{m v_i^2}{2} \right\rangle. \tag{10.31}$$

6. Вычисление нормировочного коэффициента

$$\beta = \frac{T_f}{Ek}. \tag{10.32}$$

7. Нормировка значений скорости

$$\vec{v}_i = \beta^{1/d} \vec{v}_i. \tag{10.33}$$

8. Повторение действий, описанных в п. 4—7, на последующих временных интервалах.

Ниже представлены листинг файла `Mol_din1.m`, содержащего описание функции, реализующей описанный алгоритм, и последовательность команд, которую необходимо выполнить, сохраненные нами в файле `Glava10_4.m`.

```
% листинг файла Mol_din1.m

function z=Mol_din1(Lx,Ly,Nx,Ny,N,Ni,dt,A,Tf,Nm)
% функция, возвращающая составной массив,
% содержащий значения
% координат, проекций скоростей и
% ускорений на соответствующие
% координатные оси частиц
% статистической системы с
% заданной температурой,
% в узлах временной сетки
% Lx - ширина МД-ячейки
% Ly - высота МД-ячейки
% N - число частиц системы
% Ni - число МД-шагов
```

```
% dt - шаг интегрирования уравнений движения
% A - массив, возвращенный функцией Init
% Tf - температура системы -
% число, задающее частоту применения
% процедуры перенормировки

Accold=Acc(A,Lx,Ly,N);
A2=cat(2,A,Accold);
A3=cat(3,A2);
k=0;
M=1;
Na=length(A)
while k<Ni
  A2=Verlet(A2,Lx,Ly,N,dt);
  if M==Nm
    for i=1:M
       beta=0;
       for j=1:Na
          beta=beta+0.5*(A2(j,3)^2+A2(j,4)^2);
       end;
       beta1(i)=beta;
    end;
    beta2=sum(beta1)/(M*Nx*Ny); % среднее значение
                                % кинетической энергии

    % перенормировка скоростей
    for j=1:Na
       A2(j,3)=A2(j,3)*(Tf/beta2)^0.5;
       A2(j,4)=A2(j,4)*(Tf/beta2)^0.5;
    end;
    M=1;
  else
     M=M+1;
  end;
  A3=cat(3,A3,A2);
  k=k+1;
end;
z=A3;

% листинг файла Glava10_4.m

% размер начальной ячейки
Nx=5;
Ny=5;
N=Nx*Ny; % число частиц системы
% размер МД-ячейки
```

```
Lx=8;
Ly=8;
Vmax=0.2; % максимальное значения
          % начальной скорости
A=Init(Lx,Ly,Nx,Ny,Vmax); % начальная конфигурация
Ni=10^4; % число МД-шагов
dt=1*10^-4; % величина МД-шага
md=Mol_din(Lx,Ly,N,Ni,dt,A);

% вычисление мгновенных значений
% кинетической и потенциальной
% энергий системы
for i=1:Ni+1
    Sum=0;
    tmpVx=md(:,3,i);
    tmpVy=md(:,4,i);
    for j=1:N
        Sum=Sum+0.5*(tmpVx(j)^2+tmpVy(j)^2);
    end;
    Ek(i)=Sum;
    Ep(i)=md(N,7,i);
end;

% визуализация зависимостей
% кинетической, потенциальной и полной
% энергий, приходящихся на одну частицу,
% от времени
i=1:Ni+1;
figure;
plot((i-1)*dt,Ek/N,'k')

figure;
plot((i-1)*dt,Ep/N,'k')

figure
plot((i-1)*dt,(Ek+Ep)/N,'k')

% вычисление и визуализация
% мгновенных значений давления
Z=Virial(md,Ek,Lx,Ly);
figure;
plot((i-1)*dt,Z,'k')
```

Результаты выполнения описанной последовательности команд представлены на рис. 10.14—10.17.

Рис. 10.14. Зависимость мгновенных значений кинетической энергии, приходящейся на одну частицу, от времени

Рис. 10.15. Зависимость мгновенных значений потенциальной энергии, приходящейся на одну частицу, от времени

Рис. 10.16. Зависимость мгновенных значений полной энергии, приходящейся на одну частицу, от времени

Рис. 10.17. Зависимость мгновенных значений давления от времени

Анализ зависимостей, представленных на рис. 10.14–10.17, показывает, что в моменты времени, соответствующие перенормировке, происходит скачок энергии. Между моментами нормировки, когда система не подвергается внешним воздействиям, энергия остается практически постоянной. Флуктуации энергии обусловлены периодическими граничными условиями, которые обеспечивают сохранение только полного числа частиц в МД-ячейке: при выходе частицы за пределы ячейки в нее вводится точно такая же частица. Вновь введенная частица оказывается в другом, нежели предыдущая, месте, следовательно, это приводит к изменению суммарной энергии взаимодействия частиц в системе.

***Задача* 10.11**

Важной термодинамической характеристикой систем, состоящих из большого числа частиц, является их теплоемкость при постоянном объеме

$$C_V = (\partial E/\partial T)_V,$$

где E — полная энергия системы. Данная величина есть мера количества тепловой энергии ΔE, которое необходимо сообщить системе, чтобы увеличить температуру на ΔT. Так как теплоемкость зависит от числа частиц системы, вводят удельную теплоемкость на одну частицу $c_v = C_V/N$. Наиболее просто можно оценить удельную теплоемкость, используя алгоритм, реализующийся следующей последовательностью действий:

1. *Вычислить энергию системы при различных температурах (не менее 4 значений).*

2. *Отобразить зависимость $E = E(T)$ на графике.*

3. *Проанализировав вид зависимости $E = E(T)$, выбрать вид аппроксимирующей кривой и подобрать с помощью метода наименьших квадратов соответствующие коэффициенты.*

4. *Вычислить значение первой производной аппроксимирующей зависимости.*

Используя описанный алгоритм, вычислите удельную теплоемкость изучаемой системы. Как зависит теплоемкость от температуры?

Исследуйте, как зависит удельная теплоемкость c_v от числа частиц и размеров системы.

***Задача* 10.12**

Плотные газы и жидкости описываются приближенно уравнением состояния Ван-дер-Ваальса:

$$P = T\frac{\rho}{1 - b\rho} - a\rho^2. \tag{10.34}$$

Феноменологически параметры a и b связаны с притягивающей и отталкивающей частями потенциала взаимодействия, соответственно, и в первом приближении не зависят от температуры.

1. *Проведите вычисления и постройте зависимость $P = P(T)$.*

2. *Используя зависимость $P = P(T)$, оцените значения коэффициентов a, b.*

3. *Измените размеры МД-ячейки, изменив тем самым плотность газа. Оцените значения коэффициентов a и b для новых значений плотности. Сравните значения коэффициентов a, b с соответствующими значениями, полученными в задании 10.*

10.6. ОЦЕНКА КОЭФФИЦИЕНТОВ ПЕРЕНОСА МЕТОДОМ МОЛЕКУЛЯРНОЙ ДИНАМИКИ

Будем следить за траекторией движения i-й пробной частицы. Пусть в некоторый произвольно выбранный момент времени t_1 ее координата равна $\vec{r}_i(t_1)$, в момент времени $t_2 - \vec{r}_i(t_2)$. Тогда смещение i-й пробной частицы за время $t_2 - t_1$ равно $\vec{r}_i(t_2) - \vec{r}_i(t_1)$. Среднее смещение частиц статистической системы в состоянии равновесия за время $t_2 - t_1$ определяется как среднее по всем частицам системы и по всем временным интервалам, длительность которых равна $t_2 - t_1$:

$$\bar{r} = \frac{1}{N}\sum_{i=1}^{N}\left|\vec{r}_i(t_2) - \vec{r}_i(t_1)\right| = \left\langle \frac{1}{N}\sum_{i=1}^{N}\left|\vec{r}_i(t_2) - \vec{r}_i(t_1)\right| \right\rangle. \qquad (10.35)$$

Так как каждая частица взаимодействует с другими частицами системы направление скорости частиц меняется случайным образом, поэтому среднее смещение частицы оказывается равным нулю. В этих условиях более информативной величиной является величина среднего квадрата смещения $|R(t)|^2$, определяемого формулой

$$|R(t)|^2 = \left\langle \left|\vec{r}_i(t_2) - \vec{r}_i(t_1)\right|^2 \right\rangle, \qquad (10.36)$$

где усреднение проводится по всем частицам и по временным интервалам, длительность которых равна $t = t_2 - t_1$. В состоянии равновесия средний квадрат смещения частицы $|R(t)|^2$ зависит только от разности времен $t_2 - t_1$, но не от выбора начала отсчета.

Для реализации описанной выше вычислительной процедуры нами создан файл `R2.m`, содержащий описание функции, возвращающей значения среднего квадрата смещения, и последовательность команд, которую необходимо выполнить для вычисления значений среднего квадрата смещения, сохраненную нами в файле `Glava10_5.m`. Ниже приведены листинги данных файлов.

```
% листинг файла R2.m
function z=R2(md,Lx,Ly,dt,Tstart,dk)
% функция, возвращающая значения
% среднего квадрата смещения
% md - массив, возвращенный функцией Mol_din
% Lx - ширина МД-ячейки
% Ly - высота МД-ячейки
% dt - шаг интегрирования уравнений движения
% Tstart - момент времени, начиная
% с которого проводится вычисление
% среднего квадрата смещения
% dk - определяющее шаг вычисления
% среднего квадрата смещения

K=length(md);
```

```
Kstart=floor(Tstart/dt);
Nm=floor((K-Kstart)/dk);
N=size(md,1);
C(1,1)=0;
C(1,2)=0;
for nm=1:Nm
  R=0;
  i1=0;
  for k=Kstart:K-dk*nm
    for j=1:N
      Dx=md(j,1,k)-md(j,1,floor(k+dk*nm));
      Dy=md(j,2,k)-md(j,2,floor(k+dk*nm));
     % периодические граничные условия
     if abs(Dx)>Lx/2
        Dx=Dx-sign(Dx)*Lx;
      end;
      if abs(Dy)>Ly/2
        Dy=Dy-sign(Dy)*Ly;
      end;
      R=R+Dx^2+Dy^2;
      i1=i1+1;
    end;
  end;
  C(nm+1,1)=dt*nm*dk;
  C(nm+1,2)=R/(i1*N);
end;
z=C;

% листинг файла Glava10_5.m
% размер начальной ячейки
Nx=5;
Ny=5;

N=Nx*Ny; % число частиц системы
% размер МД-ячейки
Lx=8;
Ly=8;
Vmax=0.2; % максимальное значение начальной скорости
% инициализация, начальная конфигурация
A=Init(Lx,Ly,Nx,Ny,Vmax);

% использование метода МД
Ni=10^4; % число МД-шагов
dt=1*10^-4; % величина МД-шага
md=Mol_din(Lx,Ly,N,Ni,dt,A);
```

```
% вычисление и визуализация
% среднего квадрата смещения
Z=R2(md,Lx,Ly,dt,0.7,50);
plot(Z(:,1),Z(:,2),'k')
```

Результат выполнения описанной последовательности команд представлен на рис. 10.18.

Рис. 10.18. Зависимость среднего квадрата смещения от времени

***Задача* 10.13**

В соответствии с точным решением задачи о движении частицы под действием случайных сил, полученным А. Эйнштейном, в равновесной двумерной системе средний квадрат смещения частицы $|R(t)|^2$ меняется по закону

$$|R(t)|^2 = 4Dt, \tag{10.37}$$

где D — коэффициент самодиффузии.

1. *Для вычисленной зависимости $|R(t)|^2$, используя метод наименьших квадратов, оцените значения коэффициента самодиффузии D.*
2. *Используя зависимости $|\vec{r}_i(t)|^2$, $i = 1, 2, \ldots, N$, получите оценки коэффициентов самодиффузии каждой частицы D_i. Сравните значения коэффициентов самодиффузии D и D_i. Оцените точность оценки коэффициента самодиффузии D.*

Еще одна важная характеристика поведения статистической системы — автокорреляционная функция скорости $Z(t)$, являющаяся количественной характеристикой меры зависимости значений скорости, сдвинутых относительно друг друга на определенный временной интервал.

Для вычисления значений автокорреляционной функции $Z(t)$ следует взять значения скорости каждой частицы, разделенные временным интервалом $t = t_2 - t_1$, перемножить значения скорости, усреднить полученные произведения по временным

интервалам длительностью t и всем частицам рассматриваемой системы. Математическая форма записи данного определения имеет вид

$$Z(t) = \langle \vec{v}_i(t_1)\, \vec{v}_i(t_2) \rangle, \tag{10.38}$$

где $t = t_2 - t_1$.

Коэффициент самодиффузии D и автокорреляционная функция двумерной статистической системы связаны следующим соотношением:

$$D = \frac{1}{2} \int\limits_{-\infty}^{+\infty} Z(t)\, dt. \tag{10.39}$$

Далее представлен листинг файла `CorrV.m`, содержащего описание функции, возвращающей значения среднего квадрата смещения, и последовательность команд, которую необходимо выполнить для вычисления значений среднего квадрата смещения, сохраненную нами в файле `Glava10_6.m`.

```
% листинг файла CorrV.m
function z=CorrV(md,dt,Tstart,dk)
% функция, возвращающая значения
% автокорреляционной функции скорости
% md - массив, возвращенный функцией Mol_din
% dt - шаг интегрирования уравнений движения
% Tstart - момент времени, начиная
% с которого проводится вычисление
% автокорреляционной функции

K=length(md);
Kstart=floor(Tstart/dt);
Nm=floor((K-Kstart)/dk);
N=size(md,1);
for nm=1:Nm
  V=0;
  i1=0;
  for k=Kstart:K-dk*nm
    for j=1:N
       V=V+md(j,3,k)*md(j,3,floor(k+dk*nm))+...
             md(j,4,k)*md(j,4,floor(k+dk*nm));
        i1=i1+1;
    end;
  end;
  C(nm,1)=dt*(nm-1)*dk;
  C(nm,2)=V/(i1*N);
end;
z=C;
```

```
% листинг файла Glava10_6.m
% размер начальной ячейки
Nx=5;
Ny=5;
N=Nx*Ny; % число частиц системы

% размер МД-ячейки
Lx=8;
Ly=8;
Vmax=0.2; % максимальное значение начальной скорости
A=Init(Lx,Ly,Nx,Ny,Vmax); % начальная конфигурация
Ni=10^4; % число МД-шагов
dt=1*10^-4; % величина МД-шага
md=Mol_din(Lx,Ly,N,Ni,dt,A);

% вычисление и визуализация
% автокорреляционной функции скорости
Tstart=0.7;
dk=50;
Z=CorrV(md,dt,Tstart,dk);
plot(Z(:,1),Z(:,2),'k');
```

Результат выполнения описанной последовательности команд представлен на рис. 10.19.

Рис. 10.19. Зависимость автокорреляционной функции скорости частиц статистической системы $Z(t)$ от времени $t = t_2 - t_1$

***Задача* 10.14**

Используя вычисленную зависимость $Z(t)$, оцените значение коэффициента самодиффузии в соответствии с выражением (10.38).

***Задача* 10.15**

Предположите, что зависимость $Z(t)$ от времени описывается функцией вида

$$Z(t) \cong A \exp\left(-t/\tau\right), \tag{10.40}$$

здесь параметр τ называется временем корреляции.

Подставьте теоретическую зависимость $Z(t)$ в формулу (10.40) и определите, как связан коэффициент диффузии D с временем корреляции τ. Используйте данное соотношение между D и τ для вывода выражения для D.

***Задача* 10.16**

Используя вычисленную зависимость $Z(t)$, оцените по методу наименьших квадратов численные значения величин A и τ, входящих в выражение (10.40).

***Задача* 10.17**

Получите оценки коэффициента самодиффузии D, используя результаты задач 10.16 и 10.17. Сравните найденное значение коэффициента самодиффузии с соответствующими значениями, полученными в задачах 10.14, 10.15. С какой точностью совпадают эти оценки?

10.7. МОДЕЛИРОВАНИЕ ФАЗОВЫХ ПЕРЕХОДОВ МЕТОДОМ МОЛЕКУЛЯРНОЙ ДИНАМИКИ

Рис. 10.20. К объяснению причин финитного и инфинитного движений

Анализ потенциала Ленарда–Джонса, представленного на рис. 10.20, позволяет сделать вывод о том, что в одномерном случае частица, находящаяся в этом потенциале, может совершать два вида движений: при $E < 0$ — финитное (между точками остановки X_1, X_2, являющимися корнями уравнения $U(x_{1,2}) = = 0$), при $E \leq 0$ — инфинитное (имеется одна точка остановки x_1). Следовательно, если предположить, что при достижении точки остановки частица получает извне дополнительную кинетическую энергию (например, за счет толчка), то с течением времени полная энергия частицы будет возрастать. Когда ее значение станет положительным, произойдет качественное изменение: движение из финитного станет инфинитным (произойдет «фазовый переход»).

Проведя аналогию между одномерным движением частицы в потенциале Ленарда–Джонса и поведением статистической системы, состоящей из большого количе-

ства частиц, можно предположить возникновение качественных изменений состояния рассматриваемой системы при изменении ее внутренней энергии. Движение частицы, полная энергия которой положительна, становится инфинитным, то есть частица получит возможность перемещаться по всему объему статистической системы. Если движение всех частиц системы, изначально совершавших финитное движение (кристаллическое состояние), становится инфинитным, можно говорить о фазовом переходе (система переходит из кристаллического состояния в жидкость).

Полная энергия статистической системы есть сумма кинетических энергий всех частиц, и потенциальной энергии, вычисляемой суммированием энергий парного взаимодействия частиц (10.1). Следовательно, ее величина зависит от двух параметров, задаваемых в момент времени $t = 0$ величины максимальной скорости и расстояния между частицами, то есть начальной плотности системы. Для исследования зависимости полной энергии системы от указанных величин, можно использовать функцию `InitE`, описание которой сохранено нами в файле `InitE.m`.

```
% листинг файла InitE.m
function [z,E]=InitE(Lx,Ly,Nx,Ny,Vmax,k)
% функция, возвращающая начальную конфигурацию системы
% Lx - ширина МД-ячейки
% Ly - высота МД-ячейки
% Nx - ширина начальной ячейки
% Ny - высота начальной ячейки
% Vmax - максимальное значение скорости
% k - коэффициент, определяющий начальную плотность системы

Pos_row=Ly/(Ny+1);
Pos_col=Lx/(Nx+1);
N=Nx*Ny;
i=1;
for Rows=1:Ny
  for Col=1:Nx
    x(i)=Pos_col*Col/2*k;
    y(i)=Pos_row*Rows/2*k;
    Vx(i)=Vmax*(2*rand(1)-1);
    Vy(i)=Vmax*(2*rand(1)-1);
    i=i+1;
  end;
end;
Vx_full=mean(Vx);
Vy_full=mean(Vy);
i=1:N;
Vx(i)=Vx(i)-Vx_full;
Vy(i)=Vy(i)-Vy_full;
Ep=0;
for i=1:N-1
  for j=i+1:N
```

```
    Dx=x(i)-x(j);
    if abs(Dx)>Lx/2
      Dx=Dx-sign(Dx)*Lx;
    end;
    Dy=y(i)-y(j);
    if abs(Dy)>Ly/2
      Dy=Dy-sign(Dy)*Ly;
    end;
    r=(Dx^2+Dy^2).^0.5;
    Ep=Ep+4*(1/r.^12-1/r.^6);
  end;
end;
z=cat(2,x',y',Vx',Vy');
Ek=0;
for i=1:N
  Ek=Ek+0.5*(Vx(i)^2+Vy(i)^2);
end;
E=Ep+Ek;
```

***Задача* 10.18**

Исследуйте зависимость полной энергии системы в момент времени $t = 0$ от плотности частиц и суммарной кинетической энергии.

***Задача* 10.19**

Исследуйте поведение статистической системы, имеющей следующие параметры: $Nx = 5$, $Ny = 5$, $Lx = 8$, $Ly = 8$, $V\max = 2$, $k = 1,6$. Выберите шаг интегрирования $d\tau$, равным $5 \cdot 10^{-4}$, и найдите решение уравнений движения на протяжении по крайней мере $2 \cdot 10^4$ шагов по времени.

1. *Чему равно значение полной энергии системы в момент времени $t = 0$.*
2. *Проанализируйте форму мгновенных конфигураций статистической системы в различные моменты времени. Как ведет себя система при данных начальных условиях?*
3. *Проанализируйте зависимости радиусов-векторов частиц от времени.*
4. *Оцените время достижения системой равновесного состояния. Какой критерий достижения равновесия вы использовали?*
5. *Постройте зависимости $|R(t)|^2$ и $Z(t)$ в равновесном состоянии.*
6. *Сравните зависимости $|R(t)|^2$, $Z(t)$ с аналогичными зависимостями, полученными в предыдущем разделе.*

***Задача* 10.20**

Исследуйте поведение статистической системы при увеличении температуры, достигаемом последовательным увеличением значения $V\max$, ответив на вопросы, поставленные в задаче 10.19. Удалось ли вам пронаблюдать фазовый переход кристаллическое состояние — жидкость – газ?

***Задача* 10.21**

Решите задачи 10.19, 10.20 для статистической системы, имеющей следующие параметры: $Nx = 7$, $Ny = 7$, $Lx = 20$, $Ly = 20$. *При каких значениях начальной плотности и максимальной скорости* $V\max$ *рассматриваемая система будет находиться в кристаллическом состоянии? Как протекает фазовый переход в рассматриваемой системе?*

10.8. ОСНОВНЫЕ ИТОГИ

В данной главе изложены основные сведения о методе молекулярной динамики. В том числе показано, что его использование даже для двумерной системы с небольшим количеством частиц позволяет выявить на качественном уровне ряд основных свойств статистических систем, а также ввести некоторые понятия кинетической теории переноса в жидкостях и газах. Для получения количественных результатов требуется моделировать трехмерные системы, состоящие из большого числа частиц, что с неизбежностью приводит к увеличению времени счета. Наибольшие временные затраты приходятся на формирование равновесного состояния и расчет сил и энергии. Для сокращения временных затрат используются различные подходы. Их подробное обсуждение, выходящее за рамки нашей книги, заинтересованный читатель может найти в [8, 9]. Там же приведен обширный список литературы, посвященной различным вопросам, связанным с методом МД.

ЛИТЕРАТУРА К ГЛАВЕ 10

1. *Рейф Ф.* Статистическая физика. М.: Наука, 1972.
2. *Матвеев А. Н.* Молекулярная физика. М.: Высшая школа, 1981.
3. *Ландау Л. Д., Лифшиц Е. М.* Статистическая физика. Ч. 1. М.: Наука, 1976.
4. *Климонтович Ю. Л.* Статистическая физика. М.: Наука, 1982.
5. *Румер Ю. Б., Рывкин М. Ш.* Термодинамика, статистическая физика и кинетика. М.: Наука, 1977.
6. *Ландау Л. Д., Лифшиц Е. М.* Квантовая механика. Нерелятивистская теория. М.: Наука, 1974.
7. *Гулд Х., Тобочник Я.* Компьютерное моделирование в физике. М.: Мир, 1990.
8. *Хеерман Д. В.* Методы компьютерного эксперимента в теоретической физике. М.: Мир, 1990.
9. *Лагарьков А. Н., Сергеев В. М.* Метод молекулярной динамики в статистической физике // Успехи физических наук. Т. 125. Вып. 3. 1978.С. 400—448.

Глава 11

МЕТОДЫ МОНТЕ-КАРЛО

11.1. ВВЕДЕНИЕ

Системы с большим числом степеней свободы, примеры которых мы начали рассматривать в предыдущей главе, представляют значительный интерес для физики. Вычисление термодинамических характеристик таких систем зачастую требует вычисления интегралов большой размерности. Например, классическая функция распределения неидеального газа, находящегося при температуре T (измеряемой в энергетических единицах) и состоящего из N частиц с потенциалом попарного взаимодействия $U(|\vec{r}_{ij}|)$, пропорциональна $3N$-мерному интегралу

$$Z = \int d^3r_1 \cdots d^3r_N e^{-\frac{1}{T}\sum\limits_{i<j}^{N} U(|\vec{r}_{ij}|)}. \tag{11.1}$$

За исключением очень малых значений N прямое вычисление интеграла (11.1) из-за большого объема вычислительных операций не представляется возможным. Отмеченное обстоятельство определило необходимость разработки более эффективных методов вычислений многомерных интегралов, к которым относятся методы Монте-Карло. Основная идея метода состоит в замене вычисления подынтегральной функции в каждом узле пространственной сетки, число которых для достижения требуемой точности должно быть велико, ее вычислением только для значений абсцисс, являющихся репрезентативной случайной выборкой. (Ситуация в данном случае аналогична ситуации с предсказанием результатов выборов путем опроса небольшого количества избирателей.) Метод Монте-Карло оказывается весьма эффективным при решении различных типов задач статистической физики и квантовой механики.

11.2. ЧИСЛЕННЫЕ МЕТОДЫ ИНТЕГРИРОВАНИЯ ФУНКЦИЙ, ЗАВИСЯЩИХ ОТ ОДНОЙ ПЕРЕМЕННОЙ

Предваряя обсуждение метода Монте-Карло, ознакомимся с известными численными методами, применяемыми для интегрирования функций, зависящих от одной переменной. Данные методы основаны на геометрической интерпретации определенного интеграла

$$F = \int\limits_a^b f(x)dx, \tag{11.2}$$

как площади фигуры, ограниченной графиком функции $f(x)$ и прямыми $x = a$, $x = b$ (рис. 11.1).

Рис. 11.1. К определению понятия «определенный интеграл»

Разделим отрезок $[a, b]$ на N равных отрезков длиной Δx, где

$$\Delta x = \frac{b-a}{N}. \tag{11.3}$$

Тогда координата правого конца i-го отрезка определяется по формуле

$$x_i = x_0 + i\Delta x, \tag{11.4}$$

где $x_0 = a$, $i = 0, 1, \ldots, N$.

Простейшая оценка площади под кривой $f(x)$ может быть получена как сумма площадей прямоугольника, одна из сторон которого совпадает с $[x_i, x_{i+1}]$ отрезком, а высота равна значению функции в точке x_i (метод левых прямоугольников) (рис. 11.2) или в точке x_{i+1} (метод правых прямоугольников) (рис. 11.3).

Рис. 11.2. Иллюстрация метода левых прямоугольников. Погрешность вычисления значения интеграла на каждом шаге показана закрашенными фигурами

Рис. 11.3. Иллюстрация метода правых прямоугольников. Погрешность вычисления значения интеграла на каждом шаге показана закрашенными фигурами

Значение определенного интеграла вычисляется по формулам

$$F_L = \sum_{i=0}^{N-1} f(x_i)\,\Delta x, \tag{11.5}$$

$$F_R = \sum_{i=1}^{N} f(x_i)\,\Delta x \tag{11.6}$$

для методов левых и правых прямоугольников соответственно.

***Задача* 11.1**

Вычислите значения определенных интегралов на интервале $[0, \pi/2]$, *выбрав в качестве подынтегральных функций функции* $f_1(x) = \sin(x)$, $f_2(x) = \cos(x)$ *методом левых и правых прямоугольников.*

1. *Сравните численные значения определенных интегралов, полученных методом левых и правых прямоугольников.*

2. *Вычислите точные значения определенных интегралов. Сравните точные и численные значения определенных интегралов, оцените погрешность значений определенных интегралов, найденных численно.*

3. *Исследуйте зависимость точности значения определенного интеграла от количества точек N. Постройте графики зависимости погрешности от числа точек N и шага интегрирования* Δx. *Объясните полученные результаты. Используя метод наименьших квадратов, попробуйте подобрать функцию, описывающую зависимость погрешности от числа точек N и шага интегрирования* Δx.

***Задача* 11.2**

Решите задачу 11.1 для интервала интегрирования $[0, \pi]$. *Сравните полученные результаты с результатами задачи 11.1.*

***Задача* 11.3**

Одна из известных модификаций метода прямоугольника заключается в вычислении $f(x)$ *в средней точке каждого отрезка. Получите вычислительную формулу для данной модификации метода прямоугольника. Используя найденную формулу, решите задачи 11.1, 11.2. Сравните погрешности вычисления определенного интеграла методом прямоугольников и методом средней точки.*

***Задача* 11.4**

Значение определенного интеграла зависит от подынтегральной функции и пределов интегрирования. Если в интеграле (11.2) нижний предел остается постоянным, а в качестве верхнего предела выбрана некоторая переменная y, то F является функцией от y (интеграл с переменным верхним пределом)

$$F(y) = \int_a^y f(x)\,dx. \tag{11.7}$$

1. *Для функций $f_1(x)$, $f_2(x)$ из задачи 11.1 вычислите на интервале $[0, \pi]$ значения функций $F_{1,2}(y)$ и постройте их графики.*

2. *Для заданных функций $f_1(x)$, $f_2(x)$ получите аналитические выражения интегралов с переменными верхними пределами $F_{1,2}(y)$, вычислив интеграл (11.7). Постройте графики функций $F_{1,2}(y)$.*

3. *Сравните значения функций $F_{1,2}(y)$, найденные численно, с соответствующими значениями, вычисленными по аналитической формуле. Оцените погрешность численного решения.*

4. *Постройте график зависимости величины погрешности от переменной y. Объясните полученный результат.*

Отметим, что можно повысить точность вычисления определенного интеграла, если заменять реальную функцию на каждом интервале $[x_i, x_{i+1}]$, $i = 0, 1, \ldots, N-1$, отрезком прямой, проходящей через точки с координатами $(x_i, f(x_i))$, $(x_{i+1}, f(x_{i+1}))$ (линейная интерполяция). В данном случае искомый определенный интеграл определяется как сумма площадей всех трапеций:

$$F_N = \sum_{i=0}^{N-1} \frac{1}{2}\left(f(x_{i+1}) + f(x_i)\right)\Delta x = \left[\frac{1}{2}f(x_0) + \sum_{i=1}^{N-1} f(x_i) + \frac{1}{2}f(x_N)\right]\Delta x. \tag{11.8}$$

***Задача* 11.5**

Решите задачи 11.1, 11.3, 11.4, используя для вычисления интеграла формулу трапеций (11.8). Сравните погрешности нахождения определенного интеграла методом прямоугольников и методом трапеций. Сравните погрешности нахождения значений интеграла с переменным верхним пределом методом прямоугольников и методом трапеций.

Более высокая точность вычисления интегралов обеспечивается при использовании параболической интерполяции (полиномом второй степени) по трем соседним точкам:

$$y = ax^2 + bx + c. \tag{11.9}$$

Для нахождения коэффициентов a, b, c полинома, проходящего через точки (x_0, y_0), (x_1, y_1), (x_2, y_2), нужно найти решение следующей системы линейных уравнений:

$$\begin{aligned} y_0 &= ax_0^2 + bx_0 + c, \\ y_1 &= ax_1^2 + bx_1 + c, \\ y_2 &= ax_2^2 + bx_2 + c \end{aligned} \tag{11.10}$$

относительно неизвестных a, b, c.

Решив систему (11.10) относительно неизвестных a, b, c любым известным методом (например, Крамера), подставив найденные выражения в (11.9) и выполнив элементарные преобразования, получаем

$$y = y_0 \frac{(x - x_1)(x - x_2)}{(x_0 - x_1)(x_0 - x_2)} + y_1 \frac{(x - x_0)(x - x_2)}{(x_1 - x_2)(x_1 - x_2)} + y_2 \frac{(x - x_0)(x - x_1)}{(x_2 - x_0)(x_2 - x_1)}. \quad (11.11)$$

Площадь под параболой $y = y(x)$ на интервале $[x_0, x_2]$ находится элементарным интегрированием (11.11):

$$F_0 = \frac{1}{3}(y_0 + 4y_1 + y_2)\Delta x, \quad (11.12)$$

где $\Delta x = x_1 - x_0 = x_2 - x_1$.

Искомый определенный интеграл находится как площадь всех параболических сегментов (формула Симпсона):

$$\begin{aligned} F_N = \frac{1}{3}[&f(x_0) + 4f(x_1) + 2f(x_2) + 4f(x_3) + \ldots \\ &2f(x_{N-2}) + 4f(x_{N-1}) + f(x_N)]\Delta x. \end{aligned} \quad (11.13)$$

Обратите внимание на то обстоятельство, что N в формуле Симпсона должно быть четным числом.

Аналитические оценки погрешности описанных методов интегрирования приведены в Приложении D.

***Задача* 11.6**

Решите задачи 11.1, 11.3, 11.4, используя для вычисления интеграла формулу Симпсона (11.13). Сравните погрешности нахождения определенного интеграла методом трапеций и методом Симпсона. Сравните погрешности нахождения значений интеграла с переменным верхним пределом методом трапеций и методом Симпсона.

11.3. ОСНОВЫ МЕТОДА МОНТЕ-КАРЛО

Проиллюстрируем идеи метода Монте-Карло на примере вычисления определенного интеграла от функции, зависящей от одной переменной. Пусть нам необходимо вычислить интеграл (11.2) от некоторой заданной функции $f(x)$ на интервале $[a, b]$. В предыдущем разделе мы рассмотрели несколько различных формул интегрирования, в которых использовались значения функции $f(x)$, вычисляемые в равноотстоящих точках. Однако можно использовать и другой подход, суть которого легко понять из следующего примера.

Представим себе прямоугольник высотой H и длиной $b - a$ такой, что функция $f(x)$ целиком лежит внутри данного прямоугольника (рис. 11.4). Сгенерируем N пар случайных чисел равномерно распределенных в данном прямоугольнике:

$$a \le x_i \le b, 0 \le y_i \le H. \quad (11.14)$$

Рис. 11.4. К объяснению алгоритма вычисления определенного интеграла методом Монте-Карло

Тогда доля точек (x_i, y_i), удовлетворяющих условию $y_i \leq f(x_i)$, является оценкой отношения интеграла от функции $f(x)$ к площади рассматриваемого прямоугольника. Следовательно, оценка интеграла в данном методе может быть получена по формуле

$$F_N = A\frac{n_s}{N}. \tag{11.15}$$

где n_s — число точек, удовлетворяющих условию $y_i \leq (x_i)$, N — полное количество точек, A — площадь прямоугольника.

В MATLAB описанный алгоритм реализуется следующей последовательностью команд, сохраненную нами в файле `Glava11_1.m`:

```
% листинг файла Glava11_1.m

% координаты нижнего левого угла прямоугольника
Xmin=0;
Ymin=0;
% координаты правого верхнего угла прямоугольника
Xmax=pi/2;
Ymax=1.5;
N=1000; % число точек
        % случайной последовательности

% вычисление последовательности
% случайных чисел, равномерно
% распределенных на интервале [Xmin,Xmax]
for i=1:N
    x(i)=Xmin+(Xmax-Xmin)*rand(1);
end;

% вычисление последовательности
% случайных чисел, равномерно
% распределенных на интервале [Ymin,Ymax]
```

```
for i=1:N
   y(i)=Ymin+(Ymax-Ymin)*rand(1);
end;
f=inline('sin(x)', 'x'); % задание подынтегральной функции

% подсчет точек, лежащих
% ниже графика функции f(x)
>> for i=1:N
         if f(x(i))>=y(i)
            s(i)=1;
         else
            s(i)=0;
         end;
     end;
>>S=sum(s);
>> A=(Xmax-Xmin)*(Ymax-Ymin); % площадь прямоугольника
>> A*S/N %  % вычисление значения интеграла
```

В приведенной выше последовательности команд мы использовали функцию `inline`, которая преобразует первый аргумент функции, являющейся строковым выражением, в описание функции, зависящей от переменных, стоящих на втором и последующих местах в списке формальных параметров функции. Например, для задания функции

`F(x,y) = sin(x+y)`

необходимо выполнить следующую команду:

```
    >> F=inline('sin(x+y) ', 'x', 'y')
F=
     Inline function:
     f(x,y) = sin(x+y)
```

***Задача* 11.7**

Дополните файл, листинг которого приведен выше, блоком команд, позволяющим отображать на одном рисунке график функции и точки, распределенные случайным образом.

***Задача* 11.8**

1. *Сравните значение определенного интеграла, полученное с помощью описанного выше алгоритма, с его точным значением и оцените погрешность вычислений.*

2. *Не менее 10 раз пересчитайте документ, фиксируя при этом получаемые значения определенного интеграла. Оцените среднее значение полученной последовательности значений и ее дисперсию. Сравните полученные результаты с точным значением определенного интеграла.*

3. *Выполните задание 2 для различного числа точек случайной последовательности N. Постройте график зависимости смещения значения определенного интеграла (смещение — разность между точным и вычисленным значениями определенного интеграла) от N. Постройте график зависимости дисперсии от N. Используя метод наименьших квадратов, попробуйте подобрать функцию, описывающую зависимость дисперсии от числа точек N.*

Можно предложить и другой путь вычисления определенного интеграла, рассматривая его как среднее значение функции $f(x)$ на отрезке $[a, b]$:

$$F_N = (b-a)\,\frac{1}{N}\sum_{i=1}^{N} f\left(x_i\right), \tag{11.16}$$

где x_i — последовательность случайных чисел с равномерным законом распределения на отрезке $[a, b]$.

Описанный алгоритм реализуется в MATLAB выполнением следующей последовательности команд, которая сохранена нами в файле `Glava_11_2.m`:

```
% листинг файла Glava_11_2.m

% Задание координат концов отрезка интегрирования
Xmin=0;
Xmax=pi/2;
N=1000; % длина
        % случайной последовательности

% генерация случайной последовательности
for i=1:N
    x(i)=Xmin+(Xmax-Xmin)*rand(1);
end;
f=inline('sin(x)'); % задание подынтегральной функции
% вычисление определенного интеграла
i=1:N;
s(i)=f(x(i));
(Xmax-Xmin)*mean(s)
```

***Задача* 11.9**

Решите задачу 11.9, используя для вычисления значения интеграла формулу (11.16).

Аналитическая оценка погрешности формулы (11.16) может быть получена, если рассматривать последовательность значений $f_i = f(x_i)$. Дисперсия σ^2 последовательности f_i определяется выражением

$$\sigma_f^2 = \left\langle f^2 \right\rangle - \left\langle f \right\rangle^2, \tag{11.17}$$

где

$$\langle f \rangle = \frac{1}{N} \sum_{i=1}^{N} f(x_i) \tag{11.18}$$

и

$$\langle f^2 \rangle = \frac{1}{N} \sum_{i=1}^{N} f(x_i)^2. \tag{11.19}$$

Тогда в соответствии с центральной предельной теоремой теории вероятностей дисперсия значения определенного интеграла определяется выражением

$$\sigma_F^2 \approx \frac{1}{N}\sigma_f^2 = \frac{1}{N}\left[\frac{1}{N}\sum_{i=1}^{N} f(x_i)^2 - \left(\frac{1}{N}\sum_{i=1}^{N} f(x_i)\right)^2\right]. \tag{11.20}$$

Из (11.20) вытекают два следствия, весьма важных для практического вычисления интегралов методом Монте-Карло. Во-первых, ошибка вычисления интеграла в оценке значения интеграла уменьшается как $N^{-1/2}$, то есть очень медленно. (Сравните данный результат с вашими результатами, полученными в задаче 11.9.) Из-за отмеченной особенности значения определенных интегралов от функций одной переменной, вычисленных методом Монте-Карло, оказываются менее точными, чем, например, результаты, получаемые методом трапеций, в которых ошибка убывает как N^{-2} (см. Приложение D). Однако для многомерных интегралов данное преимущество исчезает.

Второй важный момент, вытекающий из (11.20), состоит в том, что точность оценки определенного интеграла зависит от дисперсии функции σ_f, то есть функция по возможности должна изменяться как можно меньше. Для уменьшения дисперсии и, следовательно, повышения точности метода умножим и разделим подынтегральную функцию на положительную весовую функцию $w(x)$, нормированную условием

$$\int_a^b w(x)\,dx = 1. \tag{11.21}$$

В результате интеграл (11.2) можно переписать в виде

$$F = \int_a^b \frac{f(x)}{w(x)} w(x)\,dx. \tag{11.22}$$

Проведем замену переменной x на

$$y(x) = \int_a^x w(\xi)\,d\xi, \tag{11.23}$$

так что

$$\begin{aligned} \frac{dy}{dx} &= w(x), \\ y(x=a) &= 0, \\ y(x=a) &= 1. \end{aligned} \tag{11.24}$$

Тогда интеграл (11.22) становится равным

$$F = \int\limits_a^b \frac{f\left(x\left(y\right)\right)}{w\left(x\left(y\right)\right)} dy. \tag{11.25}$$

Вычисление значения данного интеграла проводится так же, как это было сделано выше, усреднением значений $f\left(x\left(y\right)\right)/w\left(x\left(y\right)\right)$ по случайной выборке точек y, равномерно распределенных на интервале $[a, b]$:

$$F \approx \frac{1}{N} \sum_{i=1}^{N} \frac{f\left(x\left(y_i\right)\right)}{w\left(x\left(y_i\right)\right)}. \tag{11.26}$$

Из (11.25) видно, что если выбрать такую функцию w, которая ведет себя подобно функции f (то есть она велика там, где велика f, и мала там, где мала f), то подынтегральная функция f/w в (11.25) может быть сделана «квазипостоянной». Описанная процедура позволяет уменьшить дисперсию оценки интеграла (11.25) при условии правильного выбора подходящей весовой функции w и возможности обратить соотношение (11.23) для получения зависимости $x(y)$.

***Задача* 11.10**

1. *Вычислите методом Монте-Карло интеграл от функции*

$$f\left(x\right) = \frac{1}{1 + x^2}$$

на интервале $[0, 1]$. *Сравните полученное значение интеграла с его точным значением.*

2. *Выберите в качестве весовой функцию*

$$w\left(x\right) = \frac{1}{3\left(4 - 2x\right)}.$$

Убедитесь в том, что данная функция удовлетворяет условию (11.21).

3. *Постройте графики функций* $w(x)$, $f(x)/w(x)$. *Сравните область значений функций* $f(x)$, $f(x)/w(x)$.

4. *Найдите явное выражение для функции* $y(x)$, *вычислив интеграл (11.23) для выбранной функции* $w(x)$. *Найдите функцию* $x = x(y)$, *обратив функцию* $y = y(x)$. *Получите аналитические выражения для функций* $f(x(y))$, $w(x(y))$. *Сгенерируйте последовательность случайных чисел с равномерным законом распределения на интервале* $[0, 1]$. *Вычислите значение определенного интеграла в соответствии с (11.26). Оцените точность вычисленного значения интеграла, сравнив его с точным значением.*

Для понимания с наиболее общих позиций, почему замена переменных оказывается полезной, необходимо учесть, что равномерное распределение точек по y соответствует распределению точек по x с плотностью вероятности $dy/dx = w(x)$, при этом $w(x)dx$ — вероятность того, что данное число принадлежит отрезку $[x, x + dx]$. Условие нормировки задается выражением (11.20). Это означает, что точки сгущаются около наиболее «важных» значений x, где w (и, как ожидается, и f) велика, и что вычисление подынтегральной функции в «несущественных точках», где w и f малы, составляет малую часть от всего объема вычислений.

Применим описанный подход для генерации чисел с заданным законом распределения $w(x)$ на интервале $[a, b]$. Обозначим через r случайное число, равномерно распределенное на интервале $[0, 1]$ с плотностью вероятности

$$p_u(r) = \begin{cases} 1, & \text{if } r \in [0,1], \\ 0, & \text{if } r \notin [0,1]. \end{cases} \tag{11.27}$$

Наша цель состоит в отыскании соотношения между x и r такого, что если r распределена по закону (11.27), x будет распределена с плотностью $w(x)$.

Для получения данного соотношения сначала вычислим интеграл

$$P(x) = \int_a^x w(\xi)\, d\xi. \tag{11.28}$$

Здесь функция $P(x)$ — интегральная функция распределения, равная вероятности получения числа меньшего или равного x. В геометрической интерпретации $P(x)$ — это часть площади фигуры под кривой плотности вероятности слева от x.

Покажем, что обращение выражения

$$P(x) = r \tag{11.29}$$

приводит к искомому соотношению

$$x = P^{-1}(x). \tag{11.30}$$

Так как переменная r распределена на отрезке $[0, 1]$ равномерно, то функция $P(x)$, связанная с r соотношением (11.29), также распределена равномерно. Вероятность того, что $P(x)$ принадлежит отрезку

$$[P(x), P(x) + dP(x)],$$

равна $dP(x)$.

С другой стороны, в соответствии с (11.29)

$$dP(x) = dr. \tag{11.31}$$

Соотношение между $dP(x)$ и dx найдем, продифференцировав выражение (11.28):

$$\frac{dP(x)}{dx} = p(x). \tag{11.32}$$

Отсюда, используя (11.27), имеем для $0 \le r \le 1$

$$dP(x) = w(x)dx = p_u(r)dr. \tag{11.33}$$

Из выражения (11.33) видно, что переменная x распределена с требуемой плотностью вероятности $p(x)$.

Таким образом, для генерации чисел с заданным законом распределения проводится в соответствии со следующим алгоритмом:

1. Генерация случайного числа r.

2. Нахождение соответствующего значения согласно (11.30).

Применим описанный алгоритм для получения случайных чисел, равномерно распределенных на отрезке $[a, b]$. Требуемая плотность вероятности равна

$$p\left(x\right) = \begin{cases} 1/(b-a), & \text{if } x \in [a, b]\,, \\ 0, & \text{if } x \notin [a, b]\,. \end{cases} \tag{11.34}$$

Функцию интегрального распределения вероятности $P(x)$ на отрезке $[a, b]$ находим, подставив (11.34) в (11.28) и вычислив интеграл. В результате получим

$$P\left(x\right) = \frac{x-a}{b-a}. \tag{11.35}$$

Далее подставляем (11.34) в (11.29)

$$r = \frac{x-a}{b-a}, \tag{11.36}$$

откуда находим искомое соотношение

$$x = P^{-1}\left(r\right) = a + (b-a)\cdot r. \tag{11.37}$$

Несмотря на то что метод неполного интеграла действует безошибочно, для его применения приходится находить обратную функцию (см. (11.30)), которая может быть получена только для сравнительно небольшого класса функций. Кроме того, может оказаться, что интеграл (11.28) не берется аналитически. Предположим, что нам известна таблица значений x_j, при которых неполный интеграл от $w(x)$ принимает ряд значений с равномерным шагом на отрезке $[a, b]$, то есть $y_j \equiv j/M$, $j = 0, 1, \ldots, M$. Тогда

$$y_j \equiv \frac{j}{M} = \int\limits_0^{x_j} w\left(\xi\right) d\xi. \tag{11.38}$$

В результате значения x_j для целых j, выбираемых с равной вероятностью из набора чисел $0, 1, \ldots, M$, будут аппроксимировать требуемое распределение. (Для концевых точек $j = 0$ и $j = M$ требуется отдельное рассмотрение.) Для нахождения x_j, проинтегрируем численно дифференциальное уравнение (11.31), применив простую дискретизацию

$$\frac{y_{j+1} - y_j}{x_{j+1} - x_j} = w\left(x_j\right). \tag{11.39}$$

Так как $y_{j+1} - y_j = 1/M$, получаем удобное рекуррентное соотношение для нахождения x_j:

$$x_{j+1} = x_j + \frac{1}{Mw\left(x_j\right)}. \tag{11.40}$$

Другим удобным методом одно- и многомерных случайных величин с неравномерным законом распределения является метод отбора-отказа (фон Нейман). Его геометрическая интерпретация представлена на рис. 11.5.

Рис. 11.5. Геометрическая интерпретация метода отбора-отказа

Предположим, что необходимо сгенерировать случайную величину x, распределенную на интервале $[0, 1]$ с вероятностью $w(x)$, и пусть $w'(x)$ — положительная функция, удовлетворяющая во всей области интегрирования условию $w'(x) > w(x)$. (Заметим, что в данном случае определенный интеграл от w' будет больше 1.) Бывает удобно (хотя и не всегда полезно) выбрать w' константой, превышающей функции $w'(x)$ в рассматриваемой области. Если теперь сгенерировать точки в двумерной области, равномерно заполняющие площадь под кривой $w'(x)$, а затем отобрать только точки, лежащие под кривой , то последние будут распределены с плотностью вероятности $w(x)$. На практике данный метод реализуется следующей последовательностью действий:

1. Берутся две случайные величины x_i, y_i, определяющие положение точки в двумерном пространстве под функцией $w'(x)$. Значение x_i принимается в том случае, если $y_i < w(x_i)$.

2. Если точка отвергается, то генерируется следующая пара чисел x_i, y_i.

Отметим, что данный метод эффективен только в том случае, когда функция сравнения $w'(x)$ выбирается близкой к $w(x)$. В противном случае на генерацию бесполезных отброшенных точек будет затрачиваться много машинного времени. Для генерации множества точек, равномерно распределенных под кривой $w'(x)$, можно использовать следующий алгоритм:

1. Взять два случайных числа $x_i \in [0, 1]$, $y_i \in [0, \max(w'(x))]$.

2. Принять значение (x_i, y_i), если $y_i \leq w'(x_i)$.

***Задача* 11.11**

Сгенерируйте методом отбора-отказа выборку точек на отрезке $[0,1]$, *распределенную по закону*

$$w(x) = \frac{6}{5}\left(1 - \frac{1}{2}x^2\right),$$

и с помощью данной выборки оцените интеграл от функции

$$f(x) = \frac{1}{1+x^2}$$

на интервале $[0,1]$.

Рассмотренный выше метод вычисления одномерных интегралов легко обобщается для вычисления d-мерных интегралов вида $F = \int f(\vec{x}_i) d^d x$:

$$F = \frac{1}{N}\sum_{i=1}^{N} f(\vec{x}_i), \tag{11.41}$$

где все координаты случайных векторов $\vec{x}_i$ выбираются независимо друг от друга.

В качестве примера расчета многомерных интегралов методом Монте-Карло рассмотрим задачу о нахождении центра масс и момента инерции двумерных твердых тел, для которых известен закон распределения плотности $\rho(x,y)$. Масса dm малого элемента площади $dxdy$ равна

$$dm = \rho(x,y)dxdy, \tag{11.42}$$

а полная масса тела

$$M = \iint \rho(x,y)\,dxdy. \tag{11.43}$$

Пределы интегрирования в (11.43) определяются геометрией тела. Координаты центра масс X, Y определяются в соответствии с формулами

$$X = \frac{1}{M}\iint x\rho(x,y)\,dxdy, \tag{11.44}$$

$$Y = \frac{1}{M}\iint y\rho(x,y)\,dxdy. \tag{11.45}$$

Момент инерции при вращении двумерного тела относительно оси Z равен

$$I_z = \iint \left(x^2+y^2\right)\rho(x,y)\,dxdy. \tag{11.46}$$

Вычисление M, X, Y осуществляется в соответствии с формулами

$$M_N = (x_2 - x_1)(y_2 - y_1)\frac{1}{N}\sum_{i=1}^{N}\rho(x_i,y_i), \tag{11.47}$$

$$X_N = (x_2 - x_1)(y_2 - y_1)\frac{1}{N}\sum_{i=1}^{N} x_i \rho(x_i, y_i), \tag{11.48}$$

$$Y_N = (x_2 - x_1)(y_2 - y_1)\frac{1}{N}\sum_{i=1}^{N} y_i \rho(x_i, y_i), \tag{11.49}$$

$$I_{z,N} = (x_2 - x_1)(y_2 - y_1)\frac{1}{N}\sum_{i=1}^{N} \left(x_i^2 + y_i^2\right) \rho(x_i, y_i), \tag{11.50}$$

где последовательности случайных чисел $x_1 \le x_i \le x_2$, $y_1 \le y_i \le y_2$ генерируются независимо друг от друга.

Рис. 11.6. Диск, рассматриваемый в задаче 11.12

Задача 11.12

1. *Для диска, изображенного на рис. 11.6, вычислите методом Монте-Карло полную массу, центр масс и момент инерции диска относительно оси Z. Примите внешний радиус диска* $r_2 = 5$, *радиус круговой полости* $r_1 = 1,5$. *Центр круговой полости находится в точке с координатами* $(0, 2{,}5)$. *Считать плотность диска однородной и равной* 1.

2. *Вычислите полную массу, центр масс и момент инерции диска относительно оси Z с помощью средств интегрирования MATLAB. Сравните полученные результаты с результатами, полученными методом Монте-Карло.*

3. *Повторите расчеты, выполненные в п. 1, 2, с плотностью*

$$\rho(x, y) = 1 + \frac{1}{2}\left(x^2 + y^2\right),$$

где координаты x, y отсчитываются от центра большого диска.

Рассмотренный метод замены переменной также может быть обобщен и на многомерный случай. При этом для весовой функции $w(\vec{r})$, нормированной на единицу, будет соответствовать новая переменная интегрирования $\vec{y}(\vec{x})$ с якобианом преобразования $\partial\vec{y}/\partial\vec{x}$. В общем случае найти обратную функцию $\vec{x}(\vec{y})$ трудно (если вообще возможно), поэтому удобнее представлять эту замену переменных в многомерном случае как операцию генерирования последовательности $\vec{x}_i(\vec{y}_i)$ с плотностью вероятности $w(\vec{y})$.

Решив задачи 11.11, 11.12, вы убедились, что метод Монте-Карло дает результаты достаточно близкие к результатам, получаемым традиционными методами. Однако для достижения заданной точности вычислений в методе Монте-Карло приходится использовать значительно большое количество частиц. Преимущества метода Монте-Карло проявляются при вычислении многомерных интегралов. Зададимся общим числом точек N, в которых проводится вычисление определенного интеграла, и сравним эффективность детерминированных и случайных методов интегрирования. В традиционном методе, например в многомерном аналоге метода трапеций, при заданном числе точек N проводится разбиение каждого отрезка, параллельного соответствующей координатной оси d-мерного пространства, на $N^{1/d}$ интервалов длиной $h \sim N^{-1/d}$. При этом ошибка интегрирования по ячейке объемом h_d оказывается пропорциональной $O(h^{d+2})$ (см. Приложение D), поэтому суммарная погрешность интегрирования по всем ячейкам в традиционном методе равна

$$NO\left(h^{d+2}\right) = O\left(N^{-2/d}\right). \tag{11.51}$$

Из (11.51) видно, что при больших d эта погрешность убывает очень медленно. С другой стороны, уравнение (11.20) показывает, что ошибка метода Монте-Карло уменьшается как $N^{-1/2}$ и не зависит от d. Если предположить, что оба численных множителя в этих оценках порядка единицы, то легко увидеть, что точность метода Монте-Карло становится более эффективной при $d > 4$. Конечно, точная оценка зависит от используемой традиционной схемы интегрирования или от того, насколько удачно выбрана весовая функция в методе Монте-Карло. Однако основной результат состоит в том, что при больших d скорости уменьшения ошибки с ростом N в детерминированных и стохастических методах существенно различаются.

11.4. АЛГОРИТМ ГЕНЕРАЦИИ СЛУЧАЙНЫХ ЧИСЕЛ С РАВНОМЕРНЫМ ЗАКОНОМ РАСПРЕДЕЛЕНИЯ

В предыдущем разделе мы обсудили вопрос о вычислении определенных интегралов методом Монте-Карло, который состоит из двух основных операций:

1) генерации точек, распределенных в области интегрирования с заданной плотностью вероятности $w(x)$;

2) вычисления значения функции $f(x)/w(x)$ в этих точках.

При этом алгоритм генерации случайных чисел с заданным законом распределения основан на использовании случайных чисел с равномерным законом распределения. Следовательно, точность вычисления интеграла зависит не только от количества

точек, используемых для вычисления значения интеграла, но и от качества работы генератора случайных чисел. В этом разделе рассматриваются основные алгоритмы и характеристики генерации случайных чисел.

Для применения в вычислительных экспериментах генератор должен обладать следующими характеристиками:

- хорошими вычислительными свойствами,
- эффективностью,
- большим периодом,
- воспроизводимостью.

Первыми и наиболее важными из рассматриваемых характеристик являются, конечно, статистические свойства. Известен пример, когда использование плохого генератора случайных чисел в методе молекулярной динамики привело к обнаружению особого поведения системы определенного размера. Результаты независимого исследования, проведенного другими авторами, не подтвердили этой особенности. Это позволило сделать вывод о том, что причиной обнаруженной особенности явилось плохое качество генератора случайных чисел [7].

Также очень важна вычислительная эффективность генератора. Программы для компьютерных экспериментов требуют огромного числа случайных чисел. Для получения порядка 10^{10} чисел за ограниченное время одно случайное число должно вычисляться очень быстро. Кроме того, важно требование на размер необходимой для работы генератора памяти компьютера. Генератор должен быть достаточно быстрым и экономичным по памяти.

Существуют различные способы генерации на компьютере случайных чисел [8]. Далее мы рассмотрим в первую очередь наиболее широко распространенные в настоящее время методы, объединенные общим названием «линейные конгруэнтные генераторы». Они производят определенным способом при задании начального целого числа последовательность случайных чисел с равномерным законом распределения. Несомненным достоинством генераторов данного типа является их воспроизводимость.

Линейный конгруэнтный генератор выдает последовательность произвольных целых чисел, которые можно привести к единичному интервалу $[0, 1]$. Для этого достаточно разделить все члены последовательности на максимально возможное целое число. Ненормированная последовательность целых чисел, однако, предпочтительнее по причине большей вычислительной эффективности алгоритма.

Алгоритм линейного конгруэнтного генератора (Лемер, 1948)

Пусть m, a, c, x_0 — действительные числа, причем $m > x_0$, $m > a$, $m > c$ и $x_0 \geq 0$, $a \geq 0$, $c \geq 0$. Тогда псевдослучайное число последовательности x_i получается из предшествующего ему числа x_{i-1} как

$$x_i = (a \cdot x_{i-1} + c) \ mod\, m.^{1} \tag{11.52}$$

[1] Здесь *mod* — функция деления по модулю, то есть функция получения остатка от деления одного числа на другое. Например 19 *mod* 4 = 3.

При соответствующем выборе чисел m, a, c, x_0 алгоритм Лемера выдает последовательность случайных чисел. При этом генерируемая последовательность имеет повторяющийся цикл или период, не превышающий m чисел. Для получения достоверных результатов при моделировании физических систем необходимо использовать генераторы с максимально большим периодом. Максимальный период достигается при $c \neq 0$. Такой генератор называется смешанным. Однако, как показала практика, смешанные генераторы дают плохие результаты по сравнению с генераторами, у которых $c = 0$ (мультипликативные генераторы). Отметим, что для обеспечения максимального периода мультипликативного конгруэнтного генератора модуль m и множитель a необходимо подбирать специально. Алгоритм линейного конгруэнтного генератора в MATLAB реализуется выполнением следующей последовательности команд, сохраненной нами в файле `Glava11_3.m`:

```
% листинг файла Glava11_3.m

clear all
a=7;
c=7;
m=55;
x(1)=20;
N=100;
for i=1:N
     x(i+1)=mod(a*x(i)+c,m);
end;
i=1:N+1;plot(i,x,'-kS');
axis([0 100 0 60]);
```

Результат выполнения приведенной выше последовательности команд представлен на рис. 11.7.

Рис. 11.7. Случайная последовательность, сгенерированная в соответствии с алгоритмом линейного конгруэнтного генератора

***Задача* 11.13**

1. *Модифицируйте программу так, чтобы в качестве начального значения в алгоритме Лемера использовалось случайное число, выдаваемое генератором случайных чисел с равномерным законом распределения MATLAB. Проанализируйте, как это скажется на свойствах генерируемой последовательности.*

2. *Сгенерируйте две последовательности, у которых начальные значения выбираются случайным образом. Вычислите корреляцию между данными последовательностями. Можно ли считать их независимыми?*

3. *Постройте и сравните гистограммы данных последовательностей. Какой величиной можно охарактеризовать степень неравномерности плотности распределения данных случайных последовательностей?*

11.5. АЛГОРИТМ МЕТРОПОЛИСА

Несмотря на то что методы генерации случайных чисел с заданным распределением, рассмотренные в предыдущем разделе, могут оказаться в ряде случаев весьма эффективными, в общем случае весьма сложно, а в большинстве случаев невозможно провести их обобщение для сложных многомерных весовых функций. Отмеченное обстоятельство определило необходимость разработки других подходов, свободных от указанных недостатков.

Один из общих методов получения произвольного неравномерного распределения вероятности был предложен Метрополисом, Розенблатом и Тейлером в 1953 г. Метод Метрополиса представляет собой частный случай процедуры выборки по значимости, в которой некоторые возможные выборки отбрасываются.

Для простоты рассмотрим одномерный вариант метода Метрополиса. Предположим, что мы хотим сгенерировать случайные переменные с произвольной плотностью вероятности $p(x)$. В методе Метрополиса моделируется «случайное блуждание» точек x_i, распределение которых после большого числа шагов асимптотически приближается к распределению вероятности $p(x)$. Случайное блуждание определяется заданием вероятности перехода $w(x_i \to x_j)$ от одного значения x_i к другому x_j. Можно показать, что достаточным (но необходимым) условием сходимости распределения последовательности $x_0, x_1, x_2, \ldots$ к $p(x)$ является выполнение условия «детального баланса»

$$p\left(x_i\right) w\left(x_i \to x_j\right) = p\left(x_j\right) w\left(x_j \to x_i\right). \tag{11.53}$$

Отметим, что соотношение (11.53) не определяет $w(x_i \to x_j)$ единственным образом. Рассмотрим простейший вариант, удовлетворяющий условию (11.53):

$$w\left(x_i \to x_j\right) = \min\left[1, \frac{p\left(x_j\right)}{p\left(x_i\right)}\right]. \tag{11.54}$$

Предположим, что «пешеход» находится в точке с координатой x_n. Для формирования x_{n+1}:

1. Выбираем случайную координату $x_t = x_n + \delta_n$, где δ_n — случайное число на отрезке $[-\delta, \delta]$.
2. Вычисляем $w = p(x_t)/p(x_n)$.
3. Если $w \geq 1$, принимаем данный переход и полагаем $x_{n+1} = x_t$.
4. Если $w < 1$, генерируем случайное число r.
5. Если $r \leq w$, принимаем данный переход и полагаем $x_{n+1} = x_t$.
6. Если пробный переход не принят, то полагаем $x_{n+1} = x_n$.

При практической реализации метода Метрополиса возникает закономерный вопрос о выборе величины пробного шага δ. Для ответа на этот вопрос предположим, что «пешеход» находится в максимуме w, то есть в точке X_n наиболее вероятного нахождения. Если шаг велик δ, то значение $w(x_i)$ будет значительно меньше, чем значение $w(X_n)$, и большинство пробных шагов будут отвергнуты, то есть выборка окажется неэффективной. Если шаг δ выбран не достаточно большим, то большинство шагов будет принято, но пешеход не сдвинется достаточно далеко, поэтому выборка также окажется неэффективной. На практике хорошим выбором является такая величина пробного шага, при которой принимается около половины всех пробных шагов.

Отметим, что алгоритм Метрополиса не лишен ряда недостатков. Во-первых, последовательные точки выборки $x_0, x_1, \ldots$ не являются независимыми, как показывает анализ способа их получения: точка x_{i+1} вероятнее всего оказывается в точке x_i. Это приводит к тому, что точки, имеющие правильное распределение, не будут статистически независимы. Например, окажется неверной оценка дисперсии (11.20) значения интеграла

$$F = \frac{\int w(x) f(x)\, dx}{\int w(x)\, dx},$$

вычисляемого путем усреднения значений функции $f(x)$ по точкам случайного блуждания. Данную зависимость можно оценить количественно, вычислив автокорреляционную функцию

$$C(k) = \frac{\langle f_i f_{i+k} \rangle - \langle f_i \rangle^2}{\langle f_i^2 \rangle - \langle f_i \rangle^2}, \tag{11.55}$$

где $\langle\ \rangle$ означают усреднение по случайным блужданиям, например

$$\langle f_i f_{i+k} \rangle = \frac{1}{N-k} \sum_{i=1}^{N-k} f(x_i)\, f(x_{i+k}). \tag{11.56}$$

На практике для устранения этого эффекта следует вычислять интеграл и его дисперсию, используя точки траектории случайного блуждания, разделенные таким временным интервалом, чтобы значение автокорреляционной функции не превосходило $0,1$.

Во-вторых, существует влияние стартовой точки на значения генерируемой последовательности, несмотря на то что в методе Метрополиса любое начальное положение является приемлемым и его результаты не должны от него зависеть. Отмеченное обстоятельство приводит к необходимости отбросить несколько начальных шагов, в течение которых система приходит в «равновесное состояние» (термализуется). На практике стартовую точку x_0 выбирают в точке максимума вероятности $w(x)$. Затем производятся несколько термализационных шагов для устранения влияния стартовой точки, и только после этого генерируется выборочная последовательность.

Продемонстрируем реализацию алгоритма Метрополиса в MATLAB на примере генерации последовательности случайных чисел, имеющих гауссово распределение

$$p(x) = \frac{1}{\sqrt{2 \cdot \pi \cdot 5}} e^{-\frac{x^2}{2 \cdot 5}} .$$

Для этого необходимо создать файл `Metropolis.m`, содержащий описание функции `Metropolis`, возвращающей значения случайной последовательности с требуемым законом распределения.

```
% листинг файла Metropolis.m
function z = Metropolis(x,Delta,p)
% функция, возвращающая случайное число,
% вычисленное в соответствии
% с алгоритмом Метрополиса
% x - координата начальной точки
% Delta - максимальное значение пробного шага
% p - имя функции, описывающей
% распределение плотности вероятности

Xt=x+Delta*(2*rand(1)-1);
w=p(Xt)/p(x);
if w>=1
  Xnew=Xt;
end;
if w<1
  r=rand(1);
  if r<=w
    Xnew=Xt;
  else
    Xnew=x;
  end;
end;
z=Xnew;
```

Далее необходимо выполнить следующую последовательность команд, сохраненную нами в файле `Glava11_4.m`:

```
% листинг файла Glava11_4.m

clear all

% задание функции, описывающей
% закон распределения
p=inline('1/((2*pi*5)^0.5)*exp(-x^2/10)','x');
N=2*10^4; % число точек
          % случайной последовательности
x(1)=0; % первый член
        % случайной последовательности

% вычисление членов
% случайной последовательности
for i=2:N
    x(i)=Metropolis(x(i-1),0.5,p);
end;

% визуализация случайной последовательности

figure(1);
i=1:N;
plot(i,x,'k');

% вычисление распределения вероятностей
Ni=50; % число интервалов гистограммы
k=1:Ni;
dx=(max(x)-min(x))/(Ni-1);
Interval(k)=min(x)+dx*(k-1);
s=hist(x,Interval);
for j=1:Ni
    z(j)=min(x)+dx*(j-1);
    Z(j)=N*p(z(j))*dx;
end;

% визуализация экспериментального и
% теоретического распределения
% вероятности
figure;
bar(Interval,s);
colormap white
hold on

plot(z,Z,'k')
hold off
```

Результат выполнения описанной последовательности команд представлен на рис. 11.8, 11.9.

Рис. 11.8. Последовательность случайных чисел, сгенерированная в соответствии с алгоритмом Метрополиса

Рис. 11.9. Теоретическая (сплошная кривая) и экспериментальная распределения вероятностей случайной последовательности, представленной на рис. 11.8

***Задача* 11.14**

Вычислите автокорреляционную функцию $C(k)$ в соответствии с (11.56). Чему равно значение $C(0)$? Определите значение k, при котором $C(k) \leq 0,1$.

***Задача* 11.15**

Модифицируйте функцию Metropolis так, чтобы она возвращала помимо значения координаты общее число переходов при вычислении перемещения на каждом шаге. Подсчитайте отношение длины случайной последовательности к полному числу шагов. Как зависит данная величина от величины пробного шага?

***Задача* 11.16**

Примените алгоритм Метрополиса для генерации одномерного нормального распределения. Для различных значений пробного шага исследуйте долю принятых пробных шагов, корреляционную функцию (и следовательно, приемлемую частоту выборки) и суммарную эффективность расчета. Воспользуйтесь сгенерированными случайными числами для вычисления интеграла

$$\int_{-\infty}^{+\infty} x^2 e^{-\frac{x^2}{2}}\, dx$$

и оцените погрешность полученного значения. Сравните эффективность метода Метрополиса с эффективностью расчета методами, рассмотренными в предыдущем разделе.

***Задача* 11.17**

1. *Используя метод Метрополиса, оцените значение интеграла*

$$\int_0^4 x^2 e^{-x} dx.$$

2. *Постройте график числа посещений «пешеходом» точек* $x_0, x_1, \ldots$ *Равномерно ли выбирается подынтегральная функция? Если нет, то в какой области изменения функции отсчеты производятся чаще?*

3. *Вычислите интеграл*

$$\int_0^4 x^2 e^{-x} dx,$$

используя метод выборки по значимости.

ЛИТЕРАТУРА К ГЛАВЕ 11

1. *Бахвалов Н. С., Лапин А. В., Чижонков Е. В.* Численные методы в задачах и упражнениях. М.: Высшая школа, 2000.

2. *Гулд Х., Тобочник Я.* Компьютерное моделирование в физике. Т. 2. М.: Мир, 1990.

3. *Кунин С.* Вычислительная физика. М.: Мир, 1992.

4. *Тейлор Дж.* Введение в теорию ошибок. М.: Мир, 1985.

5. Метод статистических испытаний / Под ред. Ю. А. Шнейдера. М.: ГИФМ, 1962.

6. *Соболь И. М.* Численные методы Монте-Карло. М.: Наука, 1973.

7. *Хеерман Д. В.* Методы компьютерного эксперимента в теоретической физике. М.: Наука, 1990.

8. *Кнут Д.* Искусство программирования для ЭВМ. Т. 2. Получисленные алгоритмы. М.: Мир, 1977.

Глава 12

СЛУЧАЙНЫЕ БЛУЖДАНИЯ

12.1. ВВЕДЕНИЕ

Впервые задача о случайных блужданиях была сформулирована в 1906 г. Пирсоном. В первоначальной формулировке требовалось определить, на каком расстоянии от начальной точки будет находиться пешеход, сделав N шагов равной длины в произвольных направлениях. Отметим, что со времени постановки задачи в такой формулировке модели случайного блуждания получили широкое распространение в физике, биологии, химии, общественных науках. Например, для описания диффузии молекул газа, движения коллоидных взвесей в жидкости, моделирования длинных полимерных цепочек, решения ряда задач квантовой механики и др. В этой и последующих главах рассматриваются модели, для реализации которых используется метод случайных блужданий.

12.2. ОДНОМЕРНЫЕ СЛУЧАЙНЫЕ БЛУЖДАНИЯ

Рассмотрим идеализированную одномерную задачу о движении пешехода, который в момент времени $t = 0$ находится в точке $x = 0$ (рис. 12.1).

Рис. 12.1. К постановке задачи об одномерном случайном блуждании

Будем считать, что все шаги имеют одинаковую длину l, а направление каждого следующего шага не зависит от направления предыдущего шага. Предположим, что на каждом шаге по времени пешеход делает шаг вправо с вероятностью p или шаг влево с вероятностью $q = 1 - p$. Обозначим общее количество шагов, сделанных пешеходом, вправо n_+, влево n_-, тогда полное число шагов равно $N = n_+ + n_-$. Полное смещение пешехода после N шагов равно $x = (n_+ - n_-)l$, где $-Nl \le x \le Nl$. Зависимость x_i, $i = 0, 1, \ldots, N$, называют реализацией случайного процесса. Совокупность зависимостей x_i, имеющих одинаковую длину и начинающихся из одной точки, называют ансамблем реализаций. Необходимо определить вероятность $P_N(x)$ того, что после N шагов пешеход окажется на расстоянии x от начала отсчета. Знание вероятности позволяет вычислить среднее смещение $\langle x_N \rangle$

$$\langle x_N \rangle = l \sum_{i=-N}^{i=N} i P_N(x) \tag{12.1}$$

и дисперсию $\left\langle \Delta x_N^2 \right\rangle$

$$\left\langle \Delta x_N^2 \right\rangle = \left\langle x_N^2 \right\rangle - \left\langle x_N \right\rangle^2, \tag{12.2}$$

где $\left\langle x_N \right\rangle^2$ — средний квадрат отклонения

$$\left\langle x_N^2 \right\rangle = l^2 \sum_{x=-N}^{N} i^2 P_N(x). \tag{12.3}$$

В методе полного перебора количество и вероятность всех блужданий для заданных N и x вычисляются в явном виде. Начнем рассмотрение метода случайных блужданий со случая, когда $N = 1$. Пешеход с вероятностью p может сместиться на один шаг вправо, затем с вероятностью p сделать следующий шаг вправо или с вероятностью q — влево (рис. 12.2).

Рис. 12.2. Случайные одномерные блуждания пешехода при $N = 1$

Значения $\langle x_1 \rangle$и $\left\langle x_1^2 \right\rangle$, вычисляемые по формулам (12.1) и (12.3), равны

$$\langle x_1 \rangle = l \sum_{i=-1}^{1} i P_1(x_i) = -lq + lp = (p - q)\ l, \tag{12.4}$$

$$\left\langle x_1^2 \right\rangle = l^2 \sum_{i=-1}^{1} i^2 P_1(x_i) = l^2 q + l^2 p = (p + q)\ l^2 = l^2. \tag{12.5}$$

Тогда среднеквадратическое отклонение $\left\langle \Delta x_1^2 \right\rangle$, вычисляемое в соответствии с (12.2), равно

$$\left\langle \Delta x_1^2 \right\rangle = l^2 - [(p - q)\ ]^2 l^2 = (1 - (p - q))\,(1 + (p + q))\ l^2 = 2q \cdot 2p \cdot l^2 = 4pql^2. \tag{12.6}$$

Рассмотрим случай $N = 2$. Пешеход делает шаг из точки $x = 0$ вправо с вероятностью p. Так как направление следующего шага пешехода не зависит от направления предыдущего, то он может оказаться с вероятностью p^2 в точке $x = 2l$ или с вероятностью pq в точке $x = 0$. Вследствие симметрии задачи аналогичный вывод можно сделать для случая, когда пешеход смещается влево от точки $x = 0$. Следовательно, вероятность обнаружить пешехода, сделавшего два шага в точке $x = 0$, равна $2pq$, в точке $x = -2l$ вероятность равна q^2.

Рис. 12.3. Случайные одномерные блуждания пешехода при $N = 2$

Для дальнейшего анализа удобно изобразить рассматриваемые смещения пешехода графически (рис. 12.3).

Среднее значение $\langle x_2 \rangle$ и средний квадрат отклонения $\left\langle x_2^2 \right\rangle$, вычисляемые

в соответствии с (12.1) и (12.3), при $N = 2$ равны

$$\langle x_2 \rangle = l \sum_{i=-2}^{2} i P_2(x_i) = -2lq^2 - lqp + lpq + 2lp^2 = 2\left(p^2 - q^2\right) l = \\ = 2(p-q)(p+q)\, l = 2(p-q)\, l. \tag{12.7}$$

$$\left\langle x_2^2 \right\rangle = l^2 \sum_{i=-2}^{2} i^2 P_2(x_i) = l^2\left(4q^2 + 4p^2\right) = 4\left(q^2 + p^2\right) l^2. \tag{12.8}$$

Подставляя (12.7), (12.8) в (12.2), находим среднеквадратичное отклонение $\left\langle \Delta x_2^2 \right\rangle$:

$$\left\langle \Delta x_2^2 \right\rangle = 8pq\, l^2. \tag{12.9}$$

Рис. 12.4. Случайные одномерные блуждания пешехода при $N = 3$

Применим метод перебора для $N = 3$. Все возможные в данном случае блуждания представлены на рис. 12.4.

Осуществляя перебор всех блужданий, аналогично тому, как это делалось для случаев $N = 1, 2$, получаем

$$\langle x_3 \rangle = l \sum_{i=-3}^{3} x_i P_3(x_i) = \\ = l\left(-3q^3 - 3pq^2 + 3p^2q + 3p^3\right) = \\ = 3l\left(p^2 - q^2\right)(p+q) = \\ = 3(p+q)(p-q) = 3(p-q), \tag{12.10}$$

$$\left\langle x_3^2 \right\rangle = l^2 \sum_{i=-3}^{3} x_i^2 P_3(x_i) = l^2\left(9q^2 + 3pq^2 + 3qp^2 + 9p^2\right) = \\ = l^2\left(12pq + [3(p-q)]^2\right), \tag{12.11}$$

$$\left\langle \Delta x_3^2 \right\rangle = 12pql^2. \tag{12.12}$$

Сравнение выражений (12.4), (12.7), (12.10) показывает, что среднее значение при произвольном N среднее смещение $\langle x_N \rangle$ может быть вычислено по формуле

$$\langle x_N \rangle = (p-q)\, N\, l. \tag{12.13}$$

Сравнивая, аналогично, (12.6), (12.9), (12.12), находим, что при произвольном N дисперсия $\left\langle \Delta x_N^2 \right\rangle$ может быть вычислена по формуле

$$\left\langle \Delta x_N^2 \right\rangle = 4pqN\, l^2. \tag{12.14}$$

***Задача* 12.1**

Найдите все возможные смещения частицы при $N = 4$. Вычислите методом перебора $\langle x_N \rangle$, $\langle \Delta x_N^2 \rangle$. Сравните полученные результаты с соответствующими значениями, вычисленными по формулам (12.13), (12.14).

Так как полное число возможных N-шаговых блужданий в одномерном случае составляет 2^n, то для реализации метода полного перебора целесообразно использовать компьютер. (Отметим, что даже в этом случае метод полного перебора ограничен малыми значениями N.) Для реализации данного метода на компьютере необходимо знать аналитическое выражение функции, описывающей вероятность $P_N(x)$, или рекуррентное соотношение, позволяющее вычислять вероятность $P_N(x)$ при N-шаговых по значению вероятности $P_{N-1}(x)$при $(N-1)$-шаговых блужданиях.

Для нахождения рекуррентного соотношения составим таблицу, в каждую ячейку которой поместим число посещений пешеходом соответствующей точки при N-шаговом блуждании ($N = 1, 2, \ldots, 5$) (табл. 12.1).

Таблица 12.1. Число посещений пешеходом точек с заданными координатами

N	$-5l$	$-4l$	$-3l$	$-2l$	$-l$	0	l	$2l$	$3l$	$4l$	$5l$
1					l	0	l				
2				l	0	$2l$	0	l			
3			1	0	3	0	0	3	0		
4		l	0	$4l$	0	$6l$	0	$4l$	0	l	
5	l	0	$5l$	0	$10l$	0	$10l$	0	$5l$	0	l

Анализ данных, представленных в табл. 12.1, показывает, что

$$\begin{gathered} P_2(\pm 2l) = P_1(\pm l)\,,\ P_3(\pm 3l) = P_2(\pm 2l)\,, \\ P_4(\pm 4l) = P_3(\pm 3l)\,, P_5(\pm 5l) = P_4(\pm 4l)\,, \\ P_2(0) = P_1(-l) + P_1(l)\,, \end{gathered} \tag{12.15}$$

$$\begin{gathered} P_3(-l) = P_2(-2l) + P_2(0)\,, P_3(l) = P_2(0) + P_2(2l)\,, \\ P_4(-2l) = P_3(-3l) + P_3(-l)\,, \\ P_4(0) = P_3(-l) + P_3(l)\,, \\ P_4(2l) = P_3(l) + P_3(3l)\,, \\ P_5(-3l) = P_4(-4l) + P_4(-2l)\,, P_5(-l) = P_4(-2l) + P_4(0)\,, \\ P_5(l) = P_4(0) + P_4(l)\,, P_5(3l) = P_4(2l) + P_4(4l)\,. \end{gathered} \tag{12.16}$$

Из (12.15), (12.16) видно, что число посещений пешеходом точки x_i при N-шаговом блуждании связано с числом посещений точек $x_{i\pm 1}$ при $(N-1)$-шаговом блуждании следующим образом:

$$P_N(x_{\pm N}) = P_{N-1}(x_{\pm N-1})\,, \tag{12.17}$$

$$P_N(x_i) = P_N(x_{i-1}) + P(x_{i+1})\,, \text{ где } i = 2, 3, \ldots, N-1. \tag{12.18}$$

Для получения значений вероятностей нахождения пешехода в точках x_i нужно умножить значение каждой ячейки из табл. 12.1 на значения соответствующей ячейки из табл. 12.2.

Таблица 12.2. Значения коэффициентов, используемых для вычисления вероятностей посещения пешеходом точек с заданными координатами

N	$-5l$	$-4l$	$-3l$	$-2l$	$-l$	0	l	$2l$	$3l$	$4l$	$5l$
1					q		p				
2				q^2		pq		p^2			
3			q^3		pq^2		p^2q		p^3		
4		q^4		pq^3		p^2q^2		p^3q		p^4	
5	q^5		pq^4		p^2q^2		p^3q^2		p^4q		p^5

***Задача* 12.2**

Изобразите графически возможные блуждания пешехода при $N = 5, 6$. Найдите значения числа посещений пешеходом соответствующих точек и сравните их с результатами, получаемыми по формулам (12.17), (12.18).

***Задача* 12.3**

Пользуясь соотношениями (12.17), (12.18), составьте программу для перебора возможных блужданий при заданных значениях N. Вычислите точные значения $P_N(x)$, $\langle x_N \rangle$, $\langle \Delta x_N^2 \rangle$.

***Задача* 12.4**

В одномерном случае оказывается возможным с помощью методов теории вероятностей получить выражение для функции $P_N(x)$ [1]:

$$P_N(x) = \frac{N!}{\left(\frac{N}{2} + \frac{x}{2}\right)!\left(\frac{N}{2} - \frac{x}{2}\right)!} p^{\left(\frac{N}{2} + \frac{x}{2}\right)} q^{\left(\frac{N}{2} - \frac{x}{2}\right)}, \tag{12.19}$$

где $\left(N + \frac{x}{2}\right)$, $\left(N - \frac{x}{2}\right)$ — число шагов влево и вправо соответственно.

Используя формулу (12.19), проверьте полученные вами результаты в задаче 12.3.

В методе Монте-Карло в отличие от метода полного перебора всех N-шаговых блужданий проводится генерация достаточно большого количества реализаций (ансамбля реализаций) (*NTrial* — число реализаций в ансамбле) с искомым законом распределения. При этом длина отдельных реализаций *NStep* должна быть такой, чтобы обеспечить принадлежность каждой реализации выбранных блужданий множеству, содержащему все допустимые блуждания. (Обычно длина реализации варьируется от 100 до 1000 шагов.) Затем из каждой реализации выбираются значения смещения пешехода на последнем шаге. Это позволяет сформировать выборку длиной *NTrial*, статистические свойства которой оказываются близкими к свойствам последовательности, получаемой методом прямого перебора.

Для реализации описанного выше вычислительного алгоритма на первом этапе следует создать файл Walk.m, содержащий описание функции, реализующей метод Монте-Карло.

```
% листинг файла Walk.m
function z = Walk(NStep,NTrial,p)
% функция, реализующая метод Монте-Карло
% NStep - длина единичной реализации
% (число шагов пешехода)
% NTrial - длина случайной последовательности,
% возвращаемой функцией Walk
% p - вероятность перехода вправо

for i=2:NTrial
  x=0;
  for j=1:NStep
    if p>=rand(1)
       x=x+1;
    else
       x=x-1;
    end;
  end;
  X(i)=x;
end;
z=X;
```

Далее необходимо выполнить следующую последовательность команд, сохраненную нами в файле Glava12_1.m:

```
% листинг файла Glava12_1.m
NStep=1000; % длина единичной реализации
NTrial=1000; % число испытаний
p=0.5; % вероятность перехода вправо
% вычисление значений
% случайной последовательности
M=Walk(NStep,NTrial,p);

% визуализация случайной последовательности
i=1:NTrial;
figure;
plot(i,M,'k')

% вычисление и визуализация гистограммы
% случайной  последовательности
Xmin=min(M);
Xmax=max(M);
```

```
Ni=50; % число интервалов гистограммы
k=1:Ni;
dx=(Xmax-Xmin)/(Ni-1); % длина интервала
                       % гистограммы
x(k)=Xmin+dx*(k-1);
s=hist(M,x);

figure;
bar(x,s);
colormap white
```

Результаты выполнения данной последовательности команд представлены на рис. 12.5, 12.6.

Рис. 12.5. Случайная последовательность, возвращенная функцией `Walk`

Рис. 12.6. Гистограмма вычисленная для случайной последовательности, представленной на рис. 12.5

Анализ гистограммы, представленной на рис. 12.6, позволяет высказать гипотезу о том, что значения случайной последовательности распределены по нормальному закону.

Для проверки данной гипотезы будем использовать следующий алгоритм:

1. Вычислить среднее отклонение пешехода от начальной точки $\langle x_N \rangle$.
2. Вычислить среднеквадратичное отклонение пешехода от начальной точки $\sqrt{\langle \Delta x_N^2 \rangle}$.
3. Вычислить теоретические частоты попадания значений случайной последовательности в заданные интервалы в предположении о нормальном законе распределения членов случайной последовательности:

$$Freq_{k1} = \int\limits_{X_{k1}}^{X_{k1+1}} NTrial \cdot normpdf\,(x, X0, \sigma),$$

где

$$normpdf\,(x, X0, \sigma) = \frac{1}{\sqrt{2\pi}\sigma} e^{-(x-X0)^2/2\sigma^2}.$$

4. Вычислить для случайной последовательности критерий χ^2:

$$\chi^2 = \sum_{i=0}^{Ni-1} \frac{(Y_i - Freq_i)^2}{Freq_i}.$$

5. Вычислить теоретическое значение критерия χ^2:

$$qchisq\,(0{,}995, rows\,(Y) - 1 - 3)\,.$$

6. Сравнить полученные критерии χ^2.

Описанный алгоритм реализуется следующей последовательностью команд, сохраненной нами в файле `Glava12_2.m`:

```
% листинг файла Glava12_2.m
NStep=1000; % длина единичной реализации
NTrial=1000; % число испытаний
p=0.5; % вероятность перехода вправо

% вычисление значений
% случайной последовательности
M=Walk(NStep,NTrial,p);

% вычисление и визуализация гистограммы
% случайной последовательности
Xmin=min(M);
Xmax=max(M);
Ni=50; % число интервалов гистограммы
k=1:Ni;
dx=(Xmax-Xmin)/(Ni-1); % длина интервала
                       % гистограммы
x(k)=Xmin+dx*(k-1);
s=hist(M,x);
X0=mean(M);
sigma=std(M);

% вычисление теоретических частот
Nf=20; % число узлов сетки для вычисления
       % интеграла
k=1:Nf;
for i=1:Ni-1
     X1=x(i);X2=x(i+1);
```

```
     dx=(X2-X1)/(Nf-1);
     x1(k)=X1+dx*(k-1);
     y1=NTrial*normpdf(x1,X0,sigma);
     Freq(i)=trapz(x1,y1);
end;

% вычисление экспериментального
% критерия Xi-квадрат
Xi=0;
for i=1:Ni-1
    Xi=Xi+(s(i)-Freq(i)).^2/Freq(i);
end;
```

В результате выполнения команды `Glava12_2` в командном окне MATLAB будут выведены теоретическое и экспериментальные значения критерия χ^2:

```
Xi
    68.3450
chi2inv(0.995,Ni-1-3)
    74.4365
```

Сравнение критерия χ^2 (равного 68,3450), полученного при данном прогоне программы для случайной последовательности, представленной на рис. 12.6, и граничного значения критерия χ^2 (равного 74,4365), определенного теоретически, позволяет сделать вывод об отсутствии оснований для опровержения гипотезы о нормальном законе распределения рассматриваемой последовательности случайных чисел.

Отметим, что для случайной последовательности данных можно использовать не только, в известной мере, интегральную характеристику — гистограмму, но и вычислять точные значения функции распределения. Для этого, как очевидно, следует подсчитать количество посещений пешеходом каждой точки отрезка $[X\min, X\max]$ и отнести их к полному числу шагов, сделанных пешеходом. Для вычисления функции распределения необходимо создать файл `DistrR`, содержащий функцию, возвращающую значения вероятности нахождения пешехода в данной точке.

```
% листинг файла DistrR.m
function z = DistrR(M)
% функция, возвращающая вероятность
% нахождения пешехода в
% данной точке
M1=min(M);
M2=max(M);
k=M1;
K=1;
while k<=M2
  n=0;
  for i=1:length(M)
    if k==M(i)
```

```
      n=n+1;
    end;
  end;
  S(K,1)=k;
  S(K,2)=n/length(M);
  K=K+1;
  k=k+1;
end;
z=S;
```

Далее необходимо выполнить следующую последовательность команд, сохраненную нами в файле `Glava12_3.m`:

```
% листинг файла Glava12_3.m
M1=DistrR(M);
bar(M1(:,1),M1(:,2),0.2);
colormap gray
```

Результат выполнения данной последовательности команд представлен на рис. 12.7.

Рис. 12.7. Вероятность нахождения пешехода последовательности в заданных точках отрезка

***Задача* 12.5**

Сравните значения $\langle x_N \rangle$, $\langle \Delta x_N^2 \rangle$, получаемые методом Монте-Карло с точными значениями, вычисляемыми по формулам (12.13), (12.14). Оцените точность определения $\langle x_N \rangle$, $\langle \Delta x_N^2 \rangle$ методом Монте-Карло. Исследуйте, как зависят погрешности $\langle x_N \rangle$, $\langle \Delta x_N^2 \rangle$ от числа шагов реализации NStep и размера ансамбля реализаций NTrial. Изобразите данные зависимости графически. Попробуйте подобрать функции, аппроксимирующие зависимости $\langle x_N \rangle$, $\langle \Delta x_N^2 \rangle$.

***Задача* 12.6**

Дополните описанный выше файл блоком команд, позволяющим вычислять среднее значение смещения пешехода и его среднеквадратическое отклонение

в соответствии с (12.1)–(12.3) для случайной последовательности, генерируемой методом Монте-Карло. Получите численные значения этих величин.

***Задача* 12.7**

Во многих практических приложениях моделей случайных блужданий величина $\langle \Delta x_N^2 \rangle$ при достаточно больших N удовлетворяет степенному закону, например,

$$\langle \Delta x_N^2 \rangle \sim N^{2\nu}. \tag{12.20}$$

(Соотношение (12.20) является примером асимптотического масштабного закона.) Для одномерных случайных блужданий, как видно из (12.14), $\nu = 1/2$.

1. *Постройте в логарифмическом масштабе график зависимости $\langle \Delta x_N^2 \rangle$ от N и оцените, используя метод наименьших квадратов, показатель степени ν, входящий в формулу (12.20).*

2. *Оцените погрешность найденного значения ν.*

3. *Исследуйте, как зависит погрешность значения ν от числа шагов реализации NStep и размера ансамбля реализаций NTrial. Изобразите данную зависимость графически. Попробуйте подобрать функцию, аппроксимирующую данную зависимость.*

***Задача* 12.8**

Еще одной важной величиной, характеризующей случайные блуждания, является среднее число различных узлов решетки, посещаемых во время N-шагового блуждания — $\langle S_N \rangle$. Вычислите методом Монте-Карло $\langle S_N \rangle$ и определите асимптотическую зависимость $\langle S_N \rangle$ от N.

***Задача* 12.9**

1. *Рассмотрите случай $p = 0,7$, $Nstep = 1000$, $NTrial = 1000$. Вычислите $\langle x_N \rangle$. Как можно интерпретировать $\langle x_N \rangle$ в данном случае?*

2. *Решите задачи 12.5–12.7 для случая $p = 0,7$.*

12.3. МЕТОД СЛУЧАЙНЫХ БЛУЖДАНИЙ НА ПЛОСКОСТИ

Рассмотрим обобщения метода одномерных случайных блужданий для систем большей размерности. Их обсуждение совершенно естественно начать с рассмотрения случайных блужданий в двумерной плоскости. Статический характер задачи случайного блуждания означает, что следует рассматривать либо большое количество случайных блужданий одного пешехода, либо большое число пешеходов, движущихся одновременно. Далее нами используется второй подход.

Рассмотрим группу пешеходов, расположенных в момент времени $t = 0$, случайным образом в круге единичного радиуса с центром в начале координат. Будем считать, что на каждом шаге по времени каждый пешеход движется случайным образом равновероятно в одном из четырех направлений: в положительном направлении оси OX, в отрицательном направлении оси OX, в положительном направлении оси OY, в отрицательном направлении оси OY. При этом направление движения выбирается в соответствии с методом Монте-Карло в зависимости от величины случайного числа и заданного порогового значения.

Для моделирования двумерных случайных блужданий необходимо создать два файла: файл `Init2.m`, содержащий описание функции, возвращающей начальную конфигурацию системы, и файл `Move.m`, содержащий описание функции, возвращающей мгновенные значения координат пешеходов.

```
% листинг файла Init2.m
function z = Init2(NParticle,R)
% функция, возвращающая начальную
% конфигурацию системы
% NParticle - число пешеходов
% R - радиус-вектор максимального смещения
% от начала координат

for i=1:NParticle
   R0=(2*rand(1)-1)*R;
   teta=2*pi*rand(1);
   XY(i,1)=R0*cos(teta);
   XY(i,2)=R0*sin(teta);
end;
z=XY;

% листинг файла Move.m
function z = Move(A,NParticle,NStep)
% функция, возвращающая
% мгновенные значения координат пешеходов
% A - массив, возвращенный функцией Init
% NParticle - число пешеходов
% NStep - число шагов Монте-Карло

for j=1:NParticle
  XY(1,1)=A(j,1);
  XY(1,2)=A(j,2);
  for i=1:NStep-1
    prob=rand(1);
    if prob<=0.25
      XY(i+1,1)=XY(i,1)+1;
      XY(i+1,2)=XY(i,1);
    end;
```

```
    if (prob>0.25)&(prob<=0.5)
      XY(i+1,1)=XY(i,1)-1;
      XY(i+1,2)=XY(i,1);
    end;
    if (prob>0.5)&(prob<=0.75)
      XY(i+1,1)=XY(i,1);
      XY(i+1,2)=XY(i,1)-1;
    end;
    if prob>0.75
      XY(i+1,1)=XY(i,1);
      XY(i+1,2)=XY(i,1)+1;
    end;
  end;
  if j==1
    Z=XY;
  else
    Z=cat(3,Z,XY);
  end;
end;
z=Z;
```

Далее необходимо выполнить следующую последовательность команд, сохраненную нами в файле `Glava12_4.m`:

```
% листинг файла Glava12_4.m
NParticle=2000; % число пешеходов
NStep=400; % число шагов Монте-Карло
R=1; % радиус-вектор
     % максимального смещения
     % от начала координат
A=Init2(NParticle,R); % вычисление
                      % начальной конфигурации

% визуализация начальной
% конфигурации системы
Np=2000;
i=1:Np;
dphi=2*pi/(Np-1);phi(i)=dphi*(i-1);
x=cos(phi);y=sin(phi);

figure;
plot(x,y,'k');
hold on
plot(A(:,1),A(:,2),'ok','Markersize',0.1);
hold off
```

```
% вычисление координат каждого
% пешехода в заданные моменты времени
Am=Move(A,NParticle,NStep);

% визуализация мгновенных
% значений координат
% и радиусов-векторов
% первого и сотого пешеходов
i=1:NStep;
figure;
plot(i,Am(:,1,1),'k',i,Am(:,1,100),':k');

figure;
plot(i,Am(:,2,1),'k',i,Am(:,2,100),':k');

R=(Am(:,1,1).^2+Am(:,2,1).^2).^0.5;
R100=(Am(:,1,100).^2+Am(:,2,100).^2).^0.5;
figure;
plot(i,R,'k',i,R100,':k');
% вычисление усредненных по ансамблю
% пешеходов значений координат
for i=1:NStep
    for j=1:NParticle
        X(j)=Am(i,1,j);
        Y(j)=Am(i,2,j);
    end;
    x1(i)=mean(X);
    y1(i)=mean(Y);
end;

% визуализация мгновенных
% значений координат
% и радиуса-вектора,
% усредненных по ансамблю пешеходов
i=1:NStep;
figure
plot(i,x1,'k')

figure;
plot(i,y1,'k')
i=1:NStep;
Rm=(x1.^2+y1.^2).^0.5;
figure;
plot(i,Rm,'k');
```

```
% вычисление и визуализация квадратов
% координат и квадрата радиуса-вектора,
% усредненных по ансамблю пешеходов
for i=1:NStep
    for j=1:NParticle
         X(j)=Am(i,1,j).^2;
         Y(j)=Am(i,2,j).^2;
    end;
    xm(i)=mean(X);
    ym(i)=mean(Y);
end;

% вычисление и визуализация мгновенных
% значений среднеквадратичного
% отклонения усредненной траектории
i=1:NStep;
figure
plot(i,xm,'k')

figure;
plot(i,ym,'k')

R2=xm+ym-x1.^2-y1.^2;
figure
plot(i,R2,'k')
```

Результаты выполнения описанной последовательности команд представлены на рис. 12.8–12.17.

Рис. 12.8. Начальное распределение пешеходов

Рис. 12.9. Зависимость мгновенных значений x-й координаты первого и сотого пешеходов от номера шага Монте-Карло

Рис. 12.10. Зависимость мгновенных значений y-й координаты первого и сотого пешеходов от номера шага Монте-Карло

Рис. 12.11. Зависимость мгновенных значений радиусов-векторов первой и сотой частиц от номера шага Монте-Карло

Рис. 12.12. Зависимость мгновенных значений x-й координаты, полученной усреднением по ансамблю пешеходов, от номера шага Монте-Карло

Рис. 12.13. Зависимость мгновенных значений y-й координаты, полученной усреднением по ансамблю пешеходов, от номера шага Монте-Карло

Рис. 12.14. Зависимость мгновенных значений радиуса-вектора, полученного усреднением по ансамблю пешеходов, от номера шага Монте-Карло

Рис. 12.15. Зависимость мгновенных значений усредненного по ансамблю пешеходов квадрата смещения вдоль оси OX от номера шага Монте-Карло

Рис. 12.16. Зависимость мгновенных значений усредненного по ансамблю пешеходов квадрата смещения вдоль оси OY от номера шага Монте-Карло

Рис. 12.17. Зависимость мгновенных значений среднеквадратичного отклонения усредненной по ансамблю пешеходов траектории от номера шага Монте-Карло

Анализ зависимости, представленной на рис. 12.17, показывает, что с течением времени происходит линейное увеличение площади окружности, внутри которой находятся все пешеходы, то есть

$$\left\langle \Delta R_i^2 \right\rangle = A \cdot i, \tag{12.21}$$

где $A \sim 1$ — коэффициент пропорциональности.

Из (12.21) видно, что среднеквадратичное смещение $R_i \equiv \sqrt{\langle \Delta R_i^2 \rangle}$ меняется с увеличением числа шагов по закону

$$\sqrt{\langle \Delta R_i^2 \rangle} = \sqrt{A} \cdot i^{\nu}, \tag{12.22}$$

где для рассматриваемой двумерной модели $\nu = 1/2$.

Задача 12.10

Создайте анимационный клип, демонстрирующий в динамике процесс увеличения площади, занимаемой пешеходами.

Задача 12.11

Оцените, используя метод наименьших квадратов, значение коэффициента A в (12.21). Определите погрешность найденного значения A, используя метод оценки погрешностей коэффициентов линейной регрессии, изложенный в [2, глава 8].

Задача 12.12

Предположим, что каждый пешеход расположен случайным образом внутри единичной окружности и имеет случайную начальную скорость, направленную в одном из четырех возможных направлений. При этом каждый пешеход в единицу времени совершает один шаг в направлении, совпадающем с направлением начальной скорости. Проведите исследование движения данной группы пешеходов и постройте зависимости:

1) $\langle x_i \rangle$, $\langle y_i \rangle$;

2) $\langle x_i^2 \rangle$, $\langle y_i^2 \rangle$;

3) $\langle \Delta x_i^2 \rangle$,$\langle \Delta y_i^2 \rangle$;

4) $\langle R_i^2 \rangle$.

Сравните результаты, полученные для модели случайной скорости, с аналогичными результатами, полученными для модели случайного блуждания. Имеются ли качественные различия в поведении рассматриваемых моделей?

Отметим, что процесс движения пешеходов подобен процессу движения молекулы в разреженном газе. Действительно, предположим, что молекула между столкновениями с другими молекулами пробегает расстояние l. Если допустить, что последовательные смещения молекулы между столкновениями статистически независимы, то ее движение будет аналогично рассмотренному выше движению пешеходов. Так как движение молекулы в разреженном газе является диффузионным процессом, возникает возможность установить связь случайного процесса с диффузией. Зависимость среднеквадратического смещения $\left\langle \Delta R(t)^2 \right\rangle$ диффундирующей частицы от времени описывается известным соотношением Эйнштейна

$$\left\langle \Delta R(t)^2 \right\rangle = 2dD\,t, \tag{12.23}$$

где d — размерность пространства, D — коэффициент самодиффузии молекулы.

Если отождествить длину свободного пробега l с длиной шага пешехода и принять время между столкновениями равным τ, то $i = t/\tau$. Следовательно, формулу (12.22) можно переписать в виде

$$\left\langle \Delta R(t)^2 \right\rangle = 2dD\tau i, \tag{12.24}$$

Сравнивая (12.21), (12.23) и учитывая, что в рассматриваемом случае $d = 2$, $\tau = 1$, $l = 1$, находим выражение для коэффициента диффузии пешехода в двумерном случае

$$D = \frac{A}{4} \approx \frac{1}{4}. \tag{12.25}$$

В общем случае для оценки коэффициента диффузии следует использовать формулу (12.14), которая с учетом сделанных выше предположений, принимает вид

$$\left\langle \Delta x^2 \right\rangle = 4pql^2 \left(\frac{t}{\tau}\right). \tag{12.26}$$

Сравнивая (12.26) с (12.23), получаем

$$D = \frac{2pql^2}{d\tau}. \tag{12.27}$$

Для случайно блуждающего пешехода в двумерном случае при $p = q = 1/2$

$$D = \frac{l^2}{4\tau}. \tag{12.28}$$

Сравнение (12.25) и (12.28) показывает, что значение коэффициентов самодиффузии D, получаемых в численных экспериментах с моделями, в которых $l \neq 1$, $\tau \neq 1$ могут быть вычислены умножением значения соответствующего коэффициента, найденного для модели, в которой $l = 1$, $\tau = 1$ на коэффициент l^2/τ.

***Задача* 12.13**

Перечислите все случайные блуждания на квадратной решетке с $N = 4$ и получите точные значения для $\langle x_N \rangle$, $\langle y_N \rangle$ и $\langle \Delta R_N^2 \rangle$. Сравните результаты аналитических расчетов с результатами, полученными методом Монте-Карло.

Отметим, что рассмотренная выше квадратная решетка, в которой число ближайших соседей равно четырем, не является единственно возможной формой решетки. На практике для моделирования двумерных плоских тел применяется треугольная решетка (рис. 12.18), период которой далее, если это не оговорено особо, будем считать равным единице.

Рис. 12.18. Возможные направления смещения пешехода при блуждании по треугольной решетке

***Задача* 12.14**

Исследуйте особенности движения группы пешеходов, распределенных в момент времени $t = 0$ случайным образом в круге единичного радиуса, на треугольной решетке.

1. *Проведите исследование движения данной группы пешеходов и постройте зависимости $\langle x_i \rangle$, $\langle y_i \rangle$, $\langle x_i^2 \rangle$, $\langle y_i^2 \rangle$, $\langle \Delta x_i^2 \rangle$, $\langle \Delta y_i^2 \rangle$, $\langle \Delta R_i^2 \rangle$.*

2. *Сравните результаты, полученные для треугольной решетки, с аналогичными результатами, полученными для случайного блуждания на квадратной решетке. Имеются ли качественные различия в поведении рассматриваемых моделей? По какому закону меняется $\langle \Delta R_i^2 \rangle$ и чему равняется v в (12.22)? Зависит ли от симметрии решетки коэффициент самодиффузии D?*

***Задача* 12.15**

Рассмотрите падение дождевой капли, начинающей движение из точки с координатами $(0, h)$, под действием случайных порывов ветра. Используйте модель случайных блужданий на квадратной решетке, считая, что вероятность шага вниз p_d больше вероятности шага вверх p_u, а вероятность смещения капли влево равна $p_{\leftarrow}$, вправо — $p_{\rightarrow}$. Так как $p_d > p_u$, то можно ожидать, что после некоторого числа шагов капля достигнет земной поверхности. Выберите вероятности переходов равными $p_d = 0,5$, $p_u = 0,1$, $p_{\leftarrow} = p_{\rightarrow} = 0,2$. Проведите моделирование движения капли методом Монте-Карло.

1. *Постройте траекторию движения капли, зависимости $\langle x_i \rangle$, $\langle y_i \rangle$, $\langle x_i^2 \rangle$, $\langle y_i^2 \rangle$, $\langle \Delta x_i^2 \rangle$, $\langle \Delta y_i^2 \rangle$, где $\langle\ \rangle$ обозначают усреднение по ансамблю реализаций.*

2. *Определите среднее время τ, за которое капля достигнет поверхности земли и функциональную зависимость $\tau = \tau(h)$. Как зависит $\langle \Delta x_i^2 \rangle$ от h и τ? Чему равняется скорость падения капли?*

   Указание: *для моделирования движения пешехода в горизонтальном направлении используйте случайные числа с равномерным законом распределения на отрезке $[-1, 1]$.*

***Задача* 12.16**

Рассмотрите двумерную модель случайных блужданий, в которой в единицу времени пешеход совершает шаг единичной длины в случайном направлении. Создайте, используя документ, описанный в разделе 12.2, программу для вычисления методом Монте-Карло вероятности $P_N(r)dr$ нахождения пешехода в интервале от r до $r+dr$, где r — расстояние, отсчитываемое от начала координат, N — число шагов, совершаемых пешеходом.

1. *Покажите, что для достаточно больших значений N функции распределения пешеходов $P_N(x)$, $P_N(y)$ вдоль осей oX, oY, соответственно, аппроксимируются нормальным законом распределения. Постройте зависимости $\langle x_N \rangle$, $\langle x_N^2 \rangle$, $\langle \Delta x_N^2 \rangle$, $\langle y_N \rangle$, $\langle y_N^2 \rangle$, $\langle \Delta y_N^2 \rangle$.*

2. *Предположив, что*
$$\langle \Delta x_N^2 \rangle, \langle \Delta y_N^2 \rangle \sim aN^{2v}, \tag{12.29}$$
   оцените, используя метод наименьших квадратов, значения коэффициент a и показателя степени ν, входящих в (12.29).

   Сравните значение ν для одномерных и двумерных случайных блужданий.

3. *Постройте зависимости $\langle R_N \rangle$, $\langle R_N^2 \rangle$, $\langle \Delta R_N^2 \rangle$. Можно ли аппроксимировать зависимость $\langle R_N^2 \rangle$ зависимостью вида (12.28)?*

4. *Постройте функцию, описывающую вероятность $P_N(r)$ нахождения пешехода в интервале от r до $t + dr$, где r — расстояние, отсчитываемое*

от начала координат, N — число шагов, совершаемых пешеходом. Используя критерий χ^2, проверьте гипотезу о возможности аппроксимации функции $P_N(r)$ гауссовой кривой.

5. *Перейдите к полярным координатам*

$$r = \sqrt{x^2 + y^2},$$

$$\varphi = arctg\,(y/x)$$

и покажите, что распределение пешеходов по углу φ близко к равномерному.

Результаты исследования модели двумерных блужданий, использованной в задаче 12.16, показывают, что функции распределения вероятностей пешеходов вдоль соответствующих координатных осей могут быть аппроксимированы нормальным законом распределения. Типичные результаты, полученные при решении задачи 12.16, представлены на рис. 12.19–12.24. (Здесь нами использовалась выборка, сформированная по $NTrial$ ($NTrial = 1000$) независимым случайным реализациям длиной $NStep$ ($NStep$=4000).)

В то же время функция распределения радиусов-векторов точек случайной последовательности, как видно из рис. 12.24 и как следует из результатов задачи 12.16 (ч. 3), не описывается нормальным законом распределения.

Рис. 12.19. X-е координаты точек двумерной случайной последовательности

Рис. 12.20. Y-е координаты точек двумерной случайной последовательности

Рис. 12.21. Радиусы-векторы точек двумерной случайной последовательности

Рис. 12.22. Теоретическая (сплошная) и экспериментальная (столбчатая диаграмма) функции распределения X-х координат случайной последовательности

Рис. 12.23. Теоретическая (сплошная) и экспериментальная (столбчатая диаграмма) функции распределения Y-х координат случайной последовательности

Рис. 12.24. Экспериментальная функция распределения радиусов-векторов точек случайной последовательности

Для объяснения полученного результата найдем функцию распределения вероятности двумерных случайных блужданий. Так как движение пешехода вдоль осей OX и OY независимы, вероятность обнаружить пешехода в окрестности точки с координатами (x, y) равна произведению функций, описывающих функции распределения вероятностей координат вдоль соответствующих координатных осей:

$$P(x, y) = P_x(x) \cdot P_y(y)\, dxdy, \tag{12.30}$$

где

$$P_x(x) = \frac{A_x}{\sqrt{2\pi}\sigma_x} e^{-(x-\langle x\rangle)^2/2\sigma_x^2}, \tag{12.31}$$

$$P_y(y) = \frac{A_y}{\sqrt{2\pi}\sigma_y} e^{-(y-\langle y\rangle)^2/2\sigma_y^2}, \tag{12.32}$$

здесь A_x, A_y — нормировочные постоянные.

Подставляя (12.31), (12.32) в (12.30) и считая, что $\sigma_x = \sigma_y = \sigma$, получим

$$P(x, y) = \frac{A}{2\pi\sigma^2} e^{-\left(x^2+y^2\right)/2\sigma^2} dxdy, \tag{12.33}$$

где A — нормировочный коэффициент, определяемый из условия

$$\iint\limits_s P(x, y)\, dxdy = N, \tag{12.34}$$

здесь s — область, занятая пешеходами, N — число пешеходов.

Переходя в (12.33) к полярным координатам

$$x = r\cos\varphi, y = r\sin\varphi,$$

получаем

$$P(r, \varphi) = \frac{A}{2\pi\sigma^2} e^{-r^2/2\sigma^2} rdrd\varphi. \tag{12.35}$$

Отметим, что выражение, стоящее в правой части (12.35), не зависит от переменной φ, что свидетельствует о равномерности распределения пешеходов по углу.

Проинтегрировав (12.35) по углу, изменяющемуся от 0 до 2π, получаем вероятность нахождения частицы в интервале $[r, r + dr]$:

$$P(r) = \frac{A}{\sigma^2} e^{-r^2/2\sigma^2} r dr. \tag{12.36}$$

Значение нормировочной постоянной A найдем из условия (12.34)

$$\int_0^R \frac{A}{\sigma^2} e^{-r^2/2\sigma^2} r dr = N, \tag{12.37}$$

где R — радиус окружности, занимаемой пешеходами.

Вычислив в (12.37) интеграл, получаем выражение для нормировочного коэффициента A:

$$A = \frac{N}{1 - e^{-R^2/2\sigma^2}}. \tag{12.38}$$

Подставляя (12.38) в (12.36), окончательно найдем

$$P(r) = \frac{N}{\sigma^2} \frac{r}{1 - e^{-R^2/2\sigma^2}} e^{-r^2/2\sigma^2} dr. \tag{12.39}$$

Результат вычисления вероятности по формуле (12.39) для случайной последовательности, мгновенные значения радиуса-вектора которой представлены на рис. 12.21, показаны на рис. 12.25.

Рис. 12.25. Теоретическая (сплошная) и экспериментальная (столбчатая диаграмма) функции распределения радиусов-векторов точек случайной последовательности

Задача 12.17

Дополните созданный вами документ для решения задачи 12.16 фрагментом, позволяющим вычислять теоретическую функцию распределения в соответствии с (12.39).

Для описания рассеяния частицы на неподвижных рассеивающих центрах с анизотропным сечением рассеивания используют модель, в которой считается, что пешеход на N-м шаге с вероятностью p смещается в направлении перехода на $(N - 1)$-м

шаге и с вероятностью $1 - p$ в противоположном направлении (персистентное случайное блуждание). Алгоритм моделирования персистентного случайного блуждания методом Монте-Карло реализуется следующей последовательностью действий:

1. Задать длину случайной последовательности (число шагов пешехода) $NStep$ и размер ансамбля случайных реализаций $NTrial$.
2. Задать начальную координату пешехода и направление первого шага.
3. Выбрать пороговое значение вероятности перехода p.
4. Сгенерировать случайное число q с равномерным законом распределения на интервале $[0, 1]$.
5. Если $q \geq p$, осуществить переход в направлении, совпадающем с направлением перехода на предыдущем шаге.
6. Если $q < p$, осуществить переход в направлении, противоположном направлению перехода на предыдущем шаге.
7. После выполнения заданного числа шагов $NStep$ вернуть пешехода в начальную точку, иначе повторять п. 4—6, пока число шагов меньше $NStep$.
8. Сгенерировать новую случайную последовательность, выполняя п. 5—7.
9. Если число случайных последовательностей меньше $NTrial$, повторить п. 5—8.

Для моделирования персистентных случайных блужданий в MATLAB в соответствии с описанным алгоритмом создадим файл `Walk2.m`, содержащий описание функции, возвращающей мгновенные значения координат пешеходов на каждом шаге Монте-Карло.

```
% листинг файла Walk2.m
function z = Walk2(NStep,NTrial,p,x0,dx)
% функция, возвращающая мгновенные
% значения координат пешеходов
% NStep - число шагов Монте-Карло
% NTrial - число пешеходов
% p - пороговое значение вероятности перехода
% x0 - координата пешехода в момент времени t=0
% dx - направление смещения пешехода на первом шаге

for i=1:NTrial
  x=x0;
  d=dx;
  K=1;
  for j=1:NStep
    if p<rand(1)
      d=-d;
```

```
    end;
    x=x+d;
    Z(K)=x;
    K=K+1;
  end;
  if i==1
   X=Z';
  else
    X=cat(2,X,Z');
  end;
end;
z=X;
```

Далее необходимо выполнить следующую последовательность команд, сохраненную нами в файле `Glava12_5.m`:

```
% листинг файла Glava12_5.m

NStep=1000; % длина единичной реализации
NTrial=1000; % число испытаний
p=0.5; % вероятность перехода вправо
x0=0;
dx=1;

% вычисление случайной последовательности
M=Walk2(NStep,NTrial,p,x0,dx);

% вычисление мгновенных значений
% среднего по ансамблю пешеходов
% смещения
for i=1:NStep
    for j=1:NTrial
        X(j)=M(i,j);
    end;
    x(i)=mean(X);
end;

% вычисление мгновенных значений
% среднего по ансамблю пешеходов
% квадрата смещения
for i=1:NStep
     for j=1:NTrial
         X2(j)=M(i,j).^2;
     end;
     x2(i)=mean(X2);
 end;
```

```
% вычисление  и визуализация мгновенных
% значений среднеквадратичного
% отклонения пешехода от  начальной точки,
% усредненных по ансамблю
% реализаций (рис. 12.26)
i=1:NStep;
Xs(i)=x2(i)-x(i).^2;
plot(i,Xs,'k')
```

Результат выполнения команды `Glava12_5` представлен на рис. 12.26.

Рис. 12.26. Зависимость мгновенных значений среднеквадратичного отклонения пешехода, усредненного по ансамблю реализаций, от номера шага Монте-Карло

***Задача* 12.18**

Используя метод наименьших квадратов, оцените по зависимости $\langle \Delta x_j^2 \rangle$, $j = 0, 1, \ldots, NStep - 1$, *значение* ν *в (12.20) и значение коэффициента самодиффузии* D.

***Задача* 12.19**

Рассмотрите, используя описанный выше документ, два случая: $p = 0,25$ *и* $p = 0,75$. *Используя метод наименьших квадратов, оцените по зависимости* $\langle \Delta x_j^2 \rangle$, $j = 0, \ldots, N - 1$ *значение* ν *в (12.22) и значение коэффициента самодиффузии* D.

Дайте физическое объяснение, почему значения коэффициента диффузии зависят от порогового значения вероятности перехода p. *Постройте зависимость коэффициента самодиффузии от порогового значения вероятности* p.

***Задача* 12.20**

Постройте зависимость $\sqrt{\langle x_j^2 \rangle}$, $j = 0, \ldots, N - 1$. *Постройте график зависимости скорости движения пешехода от времени. Оцените предельную скорость движения пешехода. Как зависит предельная скорость движения пешехода от порогового значения вероятности* p?

***Задача* 12.21**

Для описания особенностей процесса диффузии в хроматографической колонке используется следующая модель случайных блужданий: пешеход может с вероятностью p остаться на месте или с вероятностью $1 - p$ *сместиться на один шаг. Модифицируйте функцию* `Walk2` *так, чтобы данная функция возвращала ансамбль случайных последовательностей, сгенерированных по заданному правилу. Определите коэффициенты самодиффузии D при различных значениях пороговой вероятности p. Отобразите зависимость графически.*

В физике твердого тела и конденсированных сред важную роль сыграли модели случайного блуждания при наличии ловушек. Рассмотрим более подробно одну из таких моделей, используемых для изучения особенностей переноса энергии в твердом теле. Твердое тело в данной модели представляется в виде решетки, состоящей из двух типов узлов (подрешеток): узлов-хозяев и узлов-ловушек. Падающий фотон поглощается узлом-хозяином (атомом), в результате чего происходит его возбуждение. Затем энергия возбуждения (экситон) случайным образом передается ближайшим соседям и возбужденный атом возвращается в основное состояние. Блуждание экситона по решетке продолжается до тех пор, пока он не встретится с узлом-ловушкой, при котором энергия поглощается и происходит некоторый физический процесс, например химическая реакция.

Рассмотрим один из наиболее простых вариантов описанной модели переноса энергии: экситон распространяется в одномерной решетке, содержащей периодическую подрешетку, содержащую узлы-ловушки. Отметим, что если расстояние между узлами-ловушками одинаково, блуждание по бесконечной решетке можно заменить случайным блужданием по окружности (рис. 12.27).

Рис. 12.27. Эквивалентная замена случайного блуждания на одномерной решетке, содержащей подрешетку узлов-ловушек с периодом, равным 4, блужданием на окружности с $N = 3$ узлами-хозяевами (закрашенный кружок) и одним узлом-ловушкой (незакрашенный кружок)

Для оценки зависимости среднего времени жизни экситона τ_l (время, в течение которого возбуждение распространяется из начальной точки до узла-ловушки) используем метод Монте-Карло. Будем считать, что в одномерной решетке имеются узлы-ловушки, расположенные в точках $x = 0$, $x = a$ $(a > 0)$. Пешеход начинает движение из точки $x_0 \in [0, a]$ и с равной вероятностью за время $\tau = 1$ переходит в один из ближайших соседних узлов решетки. Для нахождения вида функциональной зависимости $\tau_l = \tau_l(a, x_0)$ будем использовать следующий вычислительный алгоритм:

1. Задать расстояние между узлами-ловушками a.

2. Задать число точек Ns, в которых вычисляются значения функции $\tau_l = \tau_l(a, x_0)$.

3. Провести разбиение отрезка $[0, a]$ на Ns интервалов ($x_{0,i} = (a/Ns) \cdot i$, где $i = 0, 1, \ldots, Ns$).

4. Вычислить $NTrial$ случайных реализаций, состоящих из $NStep$ шагов, в каждой точке x_{0i} — узлах разбиения отрезка $[0, a]$.

5. Определить время достижения узла-ловушки пешеходом для каждой случайной последовательности $\tau_{l_{k,i}} = \min(x_{0i}, a - x_{0i})$, где $k = 1, 2, \ldots, NTrial$.

6. Вычислить среднее время жизни возмущения, стартующего из точки x_{0i}:

$$\langle \tau_l \rangle_i = \frac{1}{NTrial} \sum_{k=1}^{NTrial} \tau_{l_{k,i}}.$$

Для реализации описанного алгоритма необходимо создать файл `Time2.m`, содержащий описание функции, возвращающей два вектора:

1) вектор, содержащий значения стартовой точки;
2) вектор, содержащий усредненное по ансамблю реализаций время достижения узла-ловушки.

```
% листинг файла Time2.m
function [X,t] = Time2(NTrial,p,Ns,a)
% функция, возвращающая значения
% зависимость времени жизни частицы,
% совершающей блуждания на решетке
% с ловушками от времени
% NTrial - число испытаний
% p - граничное значение вероятности перехода
% Ns - число узлов разбиения отрезка
% a - расстояние между узлами сетки

dx=a/(Ns+1); % шаг изменения координат
             % стартовых точек
for m=1:Ns
  i=1;
  while i<=NTrial
    j=0;
    x=dx*m;
   % перемещение i-го пешехода
   % до достижения узла-ловушки
    while (x<a)&(x>0)
      if p>rand(1)
        x=x+1;
      else
        x=x-1;
      end;
```

```
      j=j+1;
    end;
    Ns(i)=j; % время жизни i-го пешехода
    i=i+1;
  end;
  X(m)=dx*m;
  t(m)=mean(Ns); % среднее время жизни
                 % для данного значения
                 % начальной точки
end;
```

Далее необходимо выполнить следующую последовательность команд, сохраненную нами в файле `Glava12_6.m`:

```
% листинг файла Glava12_6.m
clear all
NTrial=2000; % число испытаний
Ns=6; % число узлов разбиения отрезка
a=30; % расстояние между узлами сетки

% вычисление зависимости
% времени жизни частицы,
% совершающей блуждания на решетке
% с ловушками от времени
[X T]=Time2(NTrial,0.5,Ns,a);

plot(X,T,'Sk') % визуализация зависимости
               % среднего времени жизни частицы
```

Результат выполнения описанной последовательности команд представлен на рис. 12.28.

Рис. 12.28. Зависимость $\tau_l = \tau_l(30, x_0)$

***Задача* 12.22**

Проверьте, используя метод наименьших квадратов, что зависимость $\tau_l = \tau_l(30, x_0)$ *описывается формулой*

$$\tau = (2D)^{-1} x_0 \left(a - x_0\right), \tag{12.40}$$

где D *— коэффициент самодиффузии в отсутствии узлов-ловушек.*

***Задача* 12.23**

Структуре реальных твердых тел присуще наличие разнообразных дефектов, простейшим из которых является отсутствие атома в одном из узлов решетки или наличие дополнительного атома на поверхности. При конечной температуре кристалла вакансия диффундирует, меняясь местами с соседними атомами случайным образом.

Предположив, что в момент времени $t = 0$ *вакансия расположена в центре окружности радиуса* r*, проведите моделирование движения вакансии и определите время, за которое вакансия достигает поверхности металла, находящейся на расстоянии* r*.*

Постройте распределение вероятности для времени достижения вакансией границы. Какой функцией можно аппроксимировать данное распределение вероятности?

***Задача* 12.24**

Рассмотрите одномерную решетку с отражающими узлами, расположенными в точках $x = \pm a$*. Наличие отражающих узлов, означает, что пешеход, начавший случайное движение в момент времени* $t = 0$ *из точки* $x_0 \in]-a, a[$*, попав на* n*-м шаге в точки* $x_n = \pm a$*, на* $n+1$*-шаге окажется в точках* $x = \pm a \pm 1$*.*

1. *Исследуйте случайные блуждания пешехода, используя метод Монте-Карло, и определите функцию распределения вероятности* $P_{N,x_0}(x)$*.*

2. *Сравните функции распределения вероятностей при наличии и отсутствии отражающих стенок.*

3. *Начиная с каких значений* N *различия в рассматриваемых распределениях становятся существенными?*

12.4. МОДЕЛИРОВАНИЕ ДВИЖЕНИЯ РЕШЕТОЧНОГО ГАЗА

В модели решеточного газа рассматривается случайное движение невзаимодействующих друг с другом частиц (пешеходов) на двумерной или трехмерной решетках. Условие отсутствия взаимодействия частиц означает невозможность попадания двух частиц в один узел. (Данная модель используется в физике твердого тела, так как в твердых телах (в первую очередь в металлах) диффузия обусловлена температурными

вакансиями, концентрация которых зависит от температуры.) Основной характеристикой, представляющей интерес с физической точки зрения, в данной задаче является коэффициент диффузии D. Метод Монте-Карло в данной задаче реализуется в соответствии со следующим алгоритмом:

1. Задать числа узлов решетки L.

2. Задать число частиц N ($N \leq L^2$).

3. Задать начальное распределение частиц.

4. Перемещать частицы случайным образом в один из свободных соседних узлов. (Если узел оказывается занятым, то частица остается в текущем узле.)

Для реализации описанного алгоритма в MATLAB необходимо создать файлы: `InitL.m`, содержащий описание функции, возвращающей начальную конфигурацию системы, и `MoveL.m`, содержащий описание функции, возвращающей мгновенные (на каждом шаге Монте-Карло) значения координат частиц системы.

```
% листинг файла InitL.m
function [x,y,Site] = InitL(L,NParticle)
% функция, возвращающая
% начальную конфигурацию системы

x=zeros(NParticle,1);
y=zeros(NParticle,1);
Site=zeros(NParticle,NParticle);

i=1;
while i<=NParticle
     % случайный выбор номера узла
     Xadd=floor((L+1)*rand(1)+1);
     Yadd=floor((L+1)*rand(1)+1);
     if Xadd==0
        Xadd=1;
     end;
     if Yadd==0
        Yadd=1;
     end;
     if Site(Yadd,Xadd)==0 % если узел свободен,
                           % помещаем в него частицу
        Site(Xadd,Yadd)=10^307;
         x(i)=Xadd;
         y(i)=Yadd;
         i=i+1;
      end;
   end;
```

```
% листинг файла MoveL.m
function z = MoveL(L,NParticle,NTrial,x,y,Site);
% функция, возвращающая
% мгновенные значения координат частиц

X=x;
Y=y;
Site1=Site;
z=cat(2,x,y);
for i=1:NTrial-1
  for j=1:NParticle
    ITrial=floor(NParticle*rand(1)+1); % номер перемещаемой
                                       % частицы
    if ITrial==0
      ITrial=1;
    end;
    XTrial=X(ITrial);
    YTrial=Y(ITrial);

    % перемещение в случайном направлении
    % с учетом периодических граничных условий
    Direction=floor(4*rand(1)+1);
    if Direction==1
      XTrial=XTrial+1;
      if XTrial>L
        XTrial=XTrial-L;
      end;
    end;
    if Direction==2
      XTrial=XTrial-1;
      if XTrial<1
        XTrial=L-XTrial;
      end;
    end;
    if Direction==3
      YTrial=YTrial+1;
      if YTrial>L
        YTrial=YTrial-L;
      end;
    end;
    if Direction==4
      YTrial=YTrial-1;
      if YTrial<1
        YTrial=L-YTrial;
      end;
```

```
    end;
    if Site1(YTrial,XTrial)==0 % узел свободен
      Site1(Y(ITrial),X(ITrial))=0;
      X(ITrial)=XTrial;
      Y(ITrial)=YTrial;
      Site1(YTrial,XTrial)=10^307; % новый узел занят
    end;
  end;
  z1=cat(2,X,Y);
  z=cat(3,z,z1);
end;
```

Далее необходимо выполнить следующую последовательность команд, сохраненную нами в файле `Glava12_7.m`:

```
% листинг файла Glava12_7.m
L=50; % размер ячейки
NParticle=2000; % число пешеходов
NTrial=100; % число испытаний

[Xi Yi Site] = InitL(L,NParticle); % вычисление
                                   % начальной конфигурации
figure(1);plot(Xi,Yi,'.k') % визуализация
                           % начальной конфигурации

C=MoveL(L,NParticle,NTrial,Xi,Yi,Site); % вычисление координат
                                        % пешеходов
                                        % на каждом шаге Монте-Карло
% вычисление мгновенных значений
% среднеквадратического отклонения,
% усредненных по ансамблю пешеходов
   for i=1:NTrial
        R2=0;
        X=C(:,1,i);
        Y=C(:,2,i);
        for j=1:NParticle
           dx=X(j)-Xi(j);
           dy=Y(j)-Yi(j);
           if abs(dx)>L/2
             dx=dx-dx/abs(dx)*L;
           end;
           if abs(dy)>L/2
             dy=dy-dy/abs(dy)*L;
           end;
           R2=R2+dx.^2+dy.^2;
        end;
```

```
    dR(i)=R2/NParticle;
  end;
i=1:NTrial;plot(i-1,dR)
```

Результаты выполнения приведенной выше последовательности команд представлены на рис. 12.29, 12.30.

Рис. 12.29. Расположение пешеходов в начальный момент времени

Рис. 12.30. Зависимость мгновенных значений среднеквадратического отклонения от равновесного положения, вычисленных усреднением по ансамблю пешеходов, от номера шага Монте-Карло

Сделаем два важных замечания, относящихся к рассматриваемой модели и последовательности выполняемых команд для ее реализации:

1. Для описания движения частиц на решетке конечных размеров используются периодические граничные условия, подобные граничным условиям, использованным в методе молекулярной динамики (см. главу 10.)

2. «Время» в рассматриваемой модели измеряется в единицах, равных длительности одного шага метода Монте-Карло на частицу, то есть за один шаг метода Монте-Карло на частицу каждая частица совершает в среднем один переход. Коэффициент диффузии D может быть оценен как предел выражения

$$D\left(t\right)=\frac{1}{2}\left\langle \Delta R(t)^{2}\right\rangle \tag{12.41}$$

при $t\to\infty$. Как видно из рис. 12.29, коэффициент диффузии может быть оценен как угол наклона прямой, аппроксимирующей зависимость $\Delta R(t)^2$ методом наименьших квадратов.

***Задача* 12.25**

Вычислите коэффициент диффузии решеточного газа на квадратной решетке D для значений плотности $\rho = 0,1,\ 0,2,\ 0,3,\ 0,4,\ 0,5,\ 0,6,\ 0,7,\ 0,8$

($\rho = NParticle/L^2$). Постройте график зависимости $D = D(\rho)$. Попробуйте подобрать аналитическую функцию, аппроксимирующую данную зависимость. Объясните, почему $D = D(\rho)$ является монотонно убывающей функцией плотности?

***Задача* 12.26**

Для заданного значения плотности решеточного газа вычислите несколько зависимостей $\langle \Delta R_i^2 \rangle$, определите параметры прямых аппроксимирующих данные зависимости, вычислите сумму квадратов разностей между точными значениями $\langle \Delta R_i^2 \rangle$ и соответствующими аппроксимированными значениями (сумма квадратов отклонений). Как зависит сумма квадратов отклонений от числа частиц решеточного газа N.

***Задача* 12.27**

Предложите алгоритм и вычислите зависимость вероятности перехода частицы в момент времени t в некоторую свободную ячейку и ее возвращения в момент времени $t + 1$ назад в ту же ячейку от плотности решеточного газа. Определите качественно зависимость между значением коэффициента диффузии D и данной вероятностью.

***Задача* 12.28**

Рассмотрите модель случайных блужданий частиц на одномерной решетке, в которой исключается попадание двух частиц в один узел. (Это ограничение означает, что частицы не могут проходить друг через друга.) Вычислите зависимость $\langle \Delta x^2 \rangle$ от времени. Сравните зависимость $\langle \Delta x_i^2 \rangle$ с аналогичными зависимостями, получаемыми в модели одномерных случайных блужданий. Можно ли говорить о процессе диффузии в данном случае? Какова в рассматриваемой модели зависимость $\langle \Delta x^2 \rangle$ от t?

12.5. НЕПРЕРЫВНАЯ МОДЕЛЬ СЛУЧАЙНЫХ БЛУЖДАНИЙ

Рассмотрим переход от дискретной модели случайных блужданий, рассмотренной нами в разделе 12.2, к непрерывной. Если предположить, что направления шагов пешехода равновероятны, то дискретное случайное блуждание можно переписать в виде простого «порождающего» уравнения:

$$P_n(i) = \frac{1}{2} P_{n-1}(i+1) + \frac{1}{2} P_{n-1}(i-1), \tag{12.42}$$

где $P_n(i)$ — вероятность попадания пешехода после n шагов в i-й узел.

Для получения непрерывного аналога (12.42) — дифференциального уравнения для плотности вероятности $P(x,t)$ — положим $t = n\tau$, $x = ia$, $P_n(i) = aP(x,t)$, где τ — временной интервал между двумя последовательными шагами, a — период решетки. Используя выбранные обозначения, перепишем (12.42) в эквивалентной форме:

$$P(x,t) = \frac{1}{2} P(x+a, t-\tau) + \frac{1}{2} P(x-a, t-\tau). \tag{12.43}$$

Вычтя из обеих частей (12.43) $P(x, t-\tau)$ и разделив на τ, приводим (12.43) к виду

$$\begin{aligned}&\frac{1}{\tau}\left[P(x,t)-P(x,t-\tau)\right]=\\&=\frac{a^2}{2\tau}\left[P(x+a,t-\tau)-2P(x,t-\tau)+P(x-a,t-\tau)\right]\frac{1}{a^2}.\end{aligned} \tag{12.44}$$

Считая a, τ малыми и перейдя к пределу $a \to 0$, $\tau \to 0$ при конечном отношении $D = a^2/2\tau$, получаем одномерное уравнение диффузии

$$\frac{\partial P(x,t)}{\partial t} = D\frac{\partial^2 P(x,t)}{\partial x^2}. \tag{12.45}$$

Поступая аналогично, можно получить уравнение диффузии в трехмерном пространстве

$$\frac{\partial P(x,y,z,t)}{\partial t} = D\nabla^2 P(x,y,z,t). \tag{12.46}$$

где $\nabla^2 = \partial^2/\partial x^2 + \partial^2/\partial y^2 + \partial^2/\partial z^2$ — оператор Лапласа.

Уравнение (12.45) называется уравнением диффузии или уравнением Фоккера–Планка. Оно используется для описания динамики молекул газа и жидкости. Можно показать, что решением уравнения (12.45) для свободного пространства является функция

$$P(x,t) = \frac{1}{\sqrt{2\pi Dt}}e^{-\frac{x^2}{4Dt}}, \tag{12.47}$$

где $P(x,t)$ — плотность вероятности обнаружить частицу в момент времени t в точке x, которая при $t = 0$ находилась в точке $x = 0$.

Функция, стоящая в правой части (12.46), для заданных моментов времени описывает нормальное распределение.

***Задача* 12.29**

Убедитесь в том, что:

1) *функция $P(x,t)$, задаваемая выражением (12.47), является решением уравнения (12.45);*

2)
$$\langle x(t)\rangle = \int\limits_{-\infty}^{\infty} xP(x,t)\,dx = 0; \tag{12.48}$$

3)
$$\left\langle x^2(t)\right\rangle = \int\limits_{-\infty}^{\infty} x^2P(x,t)\,dx = 2Dt. \tag{12.49}$$

Анализ (12.49) показывает, что $\left\langle x^2(t)\right\rangle \sim t$, как и для случая одномерных случайных блужданий на дискретной решетке, поэтому с течением времени его ширина меняется пропорционально $t^{1/2}$. Обобщение (12.49) на d-мерный случай дает

$$\left\langle R^2(t)\right\rangle = 2dDt,$$

где $\langle R^2(t)\rangle$ — квадрат смещения частицы.

Уравнения (12.45), (12.46) относятся к уравнениям параболического типа. Численные (детерминированные) методы их решения были рассмотрены нами выше в разделе 8.7 при обсуждении колебательной химической реакции. Отметим, что для решения данных уравнений можно использовать не только детерминированные, но и случайные методы, основанные на методе Монте-Карло. Для этого следует перейти от непрерывных уравнений (12.45), (12.46) к их дискретным аналогам и использовать случайные блуждания на дискретной пространственной сетке с дискретными временными шагами. Так как асимптотическое поведение решений уравнения в частных производных и модели случайного блуждания оказываются одинаковыми, то в данном случае метод Монте-Карло является методом численного анализа. Если целью исследования является изучение модели случайного блуждания на решетке, то метод случайного блуждания является методом моделирования. Следовательно, в данном случае различие между моделированием и численным анализом определяется только точкой зрения исследователя.

***Задача* 12.30**

Обобщите описанные в разделах 12.2, 12.3 методы случайных блужданий на случай трехмерной решетки.

1. *Вычислите зависимости* $\langle R_N\rangle$, $\langle R_N^2\rangle$, $\langle \Delta R_N^2\rangle$ *для простой кубической решетки* ($z = 6$) *и оцените значение* ν. *Зависит ли значение* ν *от размерности пространства?*

2. *Вычислите значение коэффициента диффузии для трехмерных блужданий. Согласуется ли полученное вами значение с формулой (12.41)?*

ЛИТЕРАТУРА К ГЛАВЕ 12

1. *Рейф Ф.* Статистическая физика. М.: Наука, 1972.

2. *Тейлор Дж.* Введение в теорию ошибок. М.: Мир, 1985.

3. *Гулд Х., Тобочник Я.* Компьютерное моделирование в физике. М.: Мир, 1990. Ч. 2.

4. *Гросберг А. Ю., Хохлов А. Р.* Физика в мире полимеров. М.: Наука, 1989.

5. *Гросберг А. Ю., Хохлов А. Р.* Статистическая физика макромолекул. М.: Наука, 1989.

6. *Соболь И. М.* Численные методы Монте-Карло. М.: Наука, 1973.

7. *Кунин С.* Вычислительная физика. М.: Мир, 1992.

8. *Пресс В. П., Фланнери Б. П., Тьюкольски С. А., Веттерлинг В. Т.* Численные рецепты. М.: Мир, 1990.

Глава 13

МОДЕЛИРОВАНИЕ СТАТИСТИЧЕСКОЙ СИСТЕМЫ В ПРОЦЕССЕ РЕЛАКСАЦИИ И СОСТОЯНИИ РАВНОВЕСИЯ

13.1. ВВЕДЕНИЕ

В предыдущих главах мы смогли убедиться в эффективности компьютерного моделирования при изучении поведения систем, состоящих из большого количества частиц (систем с многими степенями свободы). Оказывается, что в модельных экспериментах даже с относительно небольшим числом частиц (по сравнению с реальными системами, содержащими порядка 10^{20}—10^{25} взаимодействующих друг с другом атомов и/или молекул) можно выявить наиболее существенные особенности поведения статистических систем, понять механизмы, приводящие к такому поведению. Отметим важную особенность рассматриваемых систем, состоящую в стремлении с течением времени к беспорядку. В соответствии с современными представлениями именно данное свойство макроскопических систем определяет направление или «вектор времени».

В этой главе мы продолжаем рассмотрение свойств систем со многими степенями свободы, состояние которых может быть описано как путем задания координат, скоростей каждой частицы системы и законов их взаимодействия (описание микроскопического состояния), так и путем задания некоторых усредненных параметров системы, например средней температуры макросистемы, средней концентрации молекул и так далее (макроскопическое описание). Для демонстрации понятий порядок и беспорядок, направление времени, микроскопическое и макроскопическое описание системы нами используется модель, состоящая из N тождественных частиц, взаимодействием которых друг с другом можно пренебречь (идеальный газ). Будем также предполагать, что газ заключен в изолированный сосуд (то есть не подвержен влиянию внешней системы или силы) и пребывает в покое в течение долгого времени. Адекватным аналогом модели идеального газа является разреженный реальный газ, в котором время одного пролета между стенками сосуда много меньше, чем время между столкновениями двух молекул. Выбранная модель позволяет использовать для описания изменения параметров микроскопических состояний системы, а через них и изменения макроскопических параметров, вероятностные методы.

13.2. МОДЕЛИРОВАНИЕ ПРОЦЕССА РЕЛАКСАЦИИ СТАТИСТИЧЕСКОЙ СИСТЕМЫ

Рассмотрим сосуд, в котором находится идеальный газ, разделенный перегородкой на две равные части, одна из которых в начальный момент времени пуста (рис. 13.1). В левой половине сосуда находится N молекул. После мгновенного удаления подвижной заслонки молекулы начинают переходить из заполненной части сосуда в пустую. Поскольку движение каждой частицы происходит независимо от всех остальных, вероятность прохождения через отверстие для всех частиц одинакова. Будем полагать, что в единицу времени через отверстие проходит одна частица.

Рис. 13.1. К моделированию процесса релаксации статистической системы

Для идентификации микро- и макросостояний рассматриваемой системы предположим, что частицы изначально пронумерованы от 1 до N. Микросостояние будем описывать номерами частиц, находящимися в левой половине ящика, поскольку знание номеров этих частиц позволяет однозначно определить номера частиц, находящихся в правой половине ящика. В качестве макроскопической характеристики рассматриваемой системы будем использовать число молекул, находящихся в левой половине ящика. Основная цель исследования состоит в определении численности частиц в левой n и правой n' половинах ящика $(n+n' = N)$ в заданные моменты времени. Очевидно, что для получения однозначного ответа на данный вопрос требуется знание функции $P_N(n,t)$, определяющей вероятность нахождения в момент времени t в левой половине ящика n частиц.

Один из методов нахождения функции $P_N(n,t)$ — метод точного перебора, уже рассматривавшийся нами в главе 12 при обсуждении метода случайных блужданий. Рассмотрим метод перебора для случая, когда при $t = 0$ $n = 6$ и $n' = 0$.

При $t = 1$ возможна единственная комбинация: $n = 5$, $n' = 1$, откуда следует, что $P(n = 5, t = 1) = 1$.

При $t = 2$ возможны комбинации, когда одна из пяти частиц, оставшихся в левой половине ящика, смещается направо или же частица, находящаяся в правой половине ящика, возвращается в левую половину. Так как первая комбинация может реализоваться пятью различными способами, вероятность данного события $P(n = 4, t = 2) = 5/6$. Вторая комбинация реализуется единственным способом, поэтому вероятность данного события $P(n = 6, t = 2) = 1/6$. Следовательно, при $t = 2$ среднее число частиц в левой половине ящика $\langle n \rangle$ составляет

$$\langle n \rangle = 4P\,(n = 4, t = 2) + 6P\,(n = 6, t = 2) = 4\frac{2}{6}.$$

На следующем шаге по времени имеем

$$P\,(n = 3, t = 3) = (4/6)\,P\,(n = 4, t = 2) = 20/36,$$

что отвечает перемещению одной из четырех оставшихся частиц вправо. Рассуждая аналогично, найдем

$$P\left(n=5,t=3\right)=\left(1\right)P\left(n=6,t=2\right)+\left(2/6\right)P\left(n=4,t=2\right)=16/36.$$

Следовательно, при $t=3$ среднее число частиц в правой половине ящика составляет

$$\langle n\rangle=5P\left(n=5,t=3\right)+3P\left(n=3,t=3\right)=3\frac{8}{9}.$$

Так как полное число частиц мало, перебор удается продолжить еще несколько временных шагов. Однако, как легко видеть, количество комбинаций быстро увеличивается с ростом t. При достаточно больших N и t число возможных комбинаций становится столь велико, что не хватит мощности даже современных компьютеров.

В этих условиях, как и в задаче о случайных блужданиях, весьма эффективным оказывается метод Монте-Карло. Напомним, что в методе Монте-Карло генерируется репрезентативная выборка случайных перемещений, являющаяся подмножеством множества всех возможных перемещений. При этом чем больше окажется объем выборки, тем более высокая точность будет у получаемых результатов.

Для реализации метода Монте-Карло необходимо задать вероятность перехода частицы из левой половины ящика в правую. Так как любая частица с равной вероятностью может пройти через отверстие, вероятность перемещения в единицу времени частицы из левой половины ящика в правую пропорциональна числу частиц в левой половине в данный момент времени, деленному на полное число частиц n/N. Используя данный факт, можно моделировать временную эволюцию модели в соответствии со следующим алгоритмом, который реализуется следующей последовательностью действий:

1. Задание числа шагов *T*.
2. Инициализация счетчика $j=0$.
3. Генерация случайного числа *Prob* в интервале от 0 до 1.
4. Сравнение *Prob* с текущим значением n/N в левой половине сосуда.
5. Перемещение частицы слева направо, если $Prob\leq n/N$, и перемещение частицы справа налево при $Prob>n/N$.
6. Повторение п. 3—5, пока $j\geq T$.

Для реализации описанного алгоритма в MATLAB необходимо создать два файла:

1) файл `ParticleAtTheLeft.m`, содержащий описание функции, возвращающей единичную реализацию зависимости числа частиц в левой половине ящика $n=n(t)$;
2) Файл `Ensemble.m`, содержащий описание функции, возвращающей ансамбль реализаций зависимостей $n=n(t)$.

```
% листинг файла ParticleAtTheLeft.m
function z=ParticleAtTheLeft(NParticle,NStep)
% функция, возвращающая
% единичную реализацию зависимости
```

```
% NParticle - число частиц
            % в левой половине ящика
            % в момент времени t=0
% NStep - длина единичной реализации

LeftN(1)=NParticle;
for i=2:NStep
  Prob=LeftN(i-1)/NParticle;
  if rand(1)<=Prob
    LeftN(i)=LeftN(i-1)-1;
  else
    LeftN(i)=LeftN(i-1)+1;
  end;
end;
z=LeftN;

% листинг файла Ensemble.m
function z = Ensemble(NTrial,NParticle,NStep)
% NTrial - объем ансамбля реализаций
% NParticle - число частиц
            % в левой половине ящика
            % в момент времени t=0
% NStep - длина единичной реализации
for i=1:NTrial
  Z=ParticleAtTheLeft(NParticle,NStep);
  if i==1
    z=Z;
  else
    z=cat(3,z,Z);
  end;
end;
```

Далее необходимо выполнить следующую последовательность команд, сохраненную нами в файле `Glava13_1.m`:

```
% листинг файла Glava13_1.m
NParticle=100; % число частиц системы
NStep=1000; % длина единичной реализации
NTrial=100; % объем ансамбля реализаций

% вычисление
% ансамбля реализаций
E=Ensemble(100,NParticle,NStep);

% визуализация выбранных из ансамбля
% реализаций зависимостей
```

```
i=1:NStep;
figure(1);
plot(i,E(1,:,1),'k')

figure(2);
plot(i,E(1,:,50),'k')

figure(3);
plot(i,E(1,:,100),'k')

% вычисление средней
% по ансамблю
% реализаций зависимости
for i=1:NStep
    for j=1:NTrial
        s(j)=E(1,i,j);
    end;
    Em(i)=mean(s);
 end;

% визуализация средней
% по ансамблю
% реализаций зависимости
i=1:NStep;
figure(4);
plot(i,Em);
```

Результаты выполнения описанной последовательности команд представлены на рис. 13.2–13.5.

Рис. 13.2. Зависимость числа частиц в левой половине ящика от времени. (Ансамбль № 1)

Рис. 13.3. Зависимость числа частиц в левой половине ящика от времени. (Ансамбль № 50)

Рис. 13.4. Зависимость числа частиц в левой половине ящика от времени. (Ансамбль № 100)

Рис. 13.5. Усредненная по ансамблю реализаций зависимость числа частиц в левой половине ящика от времени

Анализ зависимости $n = n(t)$, представленной на рис. 13.5, показывает, что с течением времени рассматриваемая система стремится к состоянию, в котором в обеих половинах ящика находится примерно одинаковое число частиц. Если пренебречь малыми случайными флуктуациями, то с макроскопической точки зрения (когда нас интересуют не номера частиц, находящихся в левой половине системы, но их количество) можно считать, что начиная с некоторого момента времени параметры системы перестают зависеть от времени. Данное состояние системы называется равновесным. Время, необходимое системе для достижения равновесного состояния, называют временем релаксации τ_{rel}.

Для оценки τ_{rel} аппроксимируем зависимость $n = n(t)$ аналитической функцией вида $ae^{bt} + c$, дополнив описываемый файл, позволяющий оценивать неизвестные коэффициенты функции, используя метод наименьших квадратов. Для аппроксимации нахождения коэффициентов аппроксимирующей функции в MATLAB необходимо выполнить следующую последовательность команд:

```
% листинг файла Glava13_2.m

NParticle=100; % число частиц системы
NStep=1000; % длина единичной реализации
NTrial=100; % объем ансамбля реализаций

% вычисление ансамбля реализаций
E=Ensemble(NTrial,NParticle,NStep);

% вычисление средней
% по ансамблю реализаций зависимости
for i=1:NStep
      for j=1:NTrial
```

```
          s(j)=E(1,i,j);
      end;
      Em(i)=mean(s);
 end;
% нахождение функции, аппроксимирующей зависимость
F10=inline('exp(u(1)+u(2).*z)+u(3)','u','z') % задание
                                             % аппроксимирующей
                                             % функции
i=1:NStep;
x(i)=i;
beta=nlinfit(x,Em,F10,[5 -0.1 1]) % функция, возвращающая решение
                                  % системы уравнений метода
                                  % наименьших квадратов
                                  % [5 -0.1 1] - начальные значения
                                  % искомых коэффициентов beta

% визуализация зависимости,
% усредненной по ансамблю
% реализаций, и аппроксимирующей ее функции
plot(x,Em,'k',x,F10(beta,x),'k')
```

Результат выполнения описанной последовательности команд представлен на рис. 13.6.

Рис. 13.6. Зависимость $n = n(t)$ и аппроксимирующая ее функция

Время релаксации изучаемой системы к состоянию равновесия может быть определено из очевидного равенства

$$N_{left} := ae^{b\tau_{rel}} + c, \tag{13.1}$$

откуда

$$\tau_{rel} := \frac{1}{b} \ln \left(\frac{|\Delta N|}{a} \right), \tag{13.2}$$

где $\Delta N = \left| \frac{N}{2} - c \right|$, N — число точек зависимости $n = n(t)$.

В обозначениях, использованных в файлах MATLAB, формула (13.2) принимает следующий вид:

$$\tau_{rel} := \frac{1}{b} \ln \left(\frac{|\Delta N|}{a} \right), \tag{13.3}$$

где

$$\Delta N := \left| \frac{NParticle}{2} - beta(3) \right|. \tag{13.4}$$

Таким образом, для вычисления оценки времени релаксации необходимо дополнить описываемый документ следующими командами:

```
dN=abs(NParticle/2-beta(3));
tau=floor(1/beta(2)*log(dN/beta(1)))
```

Выполнив приведенную выше последовательность команд, получаем

```
tau = 187
```

***Задача* 13.1**

1. *Дополните, описанную выше последовательность команд, блоком команд, позволяющим для заданного набора числа частиц рассматриваемой системы проводить автоматические вычисления времени релаксации.*

2. *Постройте график зависимости* $\tau_{rel} = \tau_{rel}\,(NParticle)$.

3. *Какой функцией можно аппроксимировать данную зависимость? Объясните полученный результат с физической точки зрения.*

Мерой равновесных флуктуаций служит дисперсия σ_N числа частиц в левой половине ящика, определяемая равенством

$$\sigma^2_{N_{left}} = \left\langle {N_{left}}^2 \right\rangle - \left\langle N_{left} \right\rangle^2, \tag{13.5}$$

здесь $\langle\ \rangle$ означают усреднение по времени, проводимое после достижения системой состояния равновесия.

Для автоматизации вычислений величины относительной флуктуации

$$\left\langle \sigma_{N_{left}} / N_{left} \right\rangle$$

при различном числе частиц системы необходимо создать файл `Fluct.m`, содержащий описание функции, возвращающей значения относительных флуктуаций числа частиц, находящихся в левой половине ящика, для заданных значений числа частиц системы.

```
% листинг файла Fluct.m
function Z = Fluct(Vect_Particle,NTrial,NStep)
% функция, возвращающая значения
% относительных флуктуаций
% числа частиц, находящихся в левой половине
% ящика, для заданных в векторе Vect_Particle
% значений числа частиц системы
% Vect_Particle - вектор, содержащий
% заданные значения числа частиц системы
% NTrial - число испытаний
% NStep - длина единичной реализации
M=length(Vect_Particle);

% задание вида функции,
% аппроксимирующей зависимость
n=1:NStep;
F10=inline('exp(u(1)+u(2).*z)+u(3)','u','z');
x(n)=n;

for m=1:M
   % вычисление мгновенных значений
   % зависимости для системы, состоящей из
   % Vect_Particle(m) частиц
    E=Ensemble(NTrial,Vect_Particle(m),NStep);

    % вычисление мгновенных значений
    % зависимости  n=n(t), усредненной по
    % ансамблю реализаций
    for i=1:NStep
        for j=1:Vect_Particle(m)
            s(j)=E(1,i,j);
        end;
        Em(i)=mean(s);
    end;

   % оценка времени релаксации системы,
   % состоящей из Vect_Particle(m) частиц
   beta=nlinfit(x,Em,F10,[5 -0.1 1]);
   dN=abs(Vect_Particle(m)/2-beta(3));
   tau=floor(1/beta(2)*log(dN/beta(1)));

   % вычисление мгновенных значений числа
   % частиц, находящихся в левой
   % половине ящика в равновесном состоянии
   k=1;
```

```
   for j=tau:NStep
         z(k)=Em(j)-Vect_Particle(m)/2;
         z1(k)=Em(j);
         k=k+1;
   end;

   % относительные флуктуации числа частиц,
   % находящихся в левой половине ящика, в
   % равновесном состоянии от времени
    Z(m)=std(z)/mean(z1);
end;
```

Далее необходимо выполнить следующую последовательность команд, сохраненную нами в файле `Glava13_3.m`:

```
% листинг файла Glava13_3.m

i=1:13;
V=10+3*(i-1);
NTrial=100; % число испытаний
NStep=1000; % длина единичной
            % последовательности

% вычисление и визуализация зависимости
% величины относительных флуктуаций
% от числа частиц статистической системы
A=Fluct(V,NTrial,NStep);
plot(V,A,'kx'); axis([9 49 0 0.1])
```

Результат выполнения описанной последовательности команд представлен на рис. 13.7.

Рис. 13.7. Зависимость величины относительных флуктуаций от числа частиц системы

Анализ зависимости, представленной на рис. 13.7, позволяет предположить, что величина относительных флуктуаций числа частиц, находящихся в левой половине ящика, зависит от числа частиц системы как $1/\sqrt{N}$. Для проверки гипотезы аппроксимируем данную зависимость функцией вида $a/\sqrt{b \cdot x + c}$.

Для нахождения коэффициентов аппроксимирующей функции необходимо выполнить последовательность команд, сохраненную нами в файле `Glava13_4.m`:

```
% листинг файла Glava13_4.m
i=1:13;
V=10+3*(i-1);
NTrial=100; % число испытаний
NStep=1000; % длина единичной
            % последовательности

% вычисление и визуализация зависимости
% величины относительных
% флуктуаций от числа частиц
% статистической системы
A=Fluct(V,NTrial,NStep);
plot(V,A,'kx');
axis([9 49 0 0.1])

% вычисление коэффициентов функции,
% аппроксимирующей зависимость
% относительной величины флуктуаций
% от числа частиц системы
F11=inline('u(1)./(u(2).*z+u(3)).^0.5','u','z');
beta=nlinfit(V,A,F11,[1 3 -1]);

% визуализация аппроксимирующей функции
Nx=500;
j=1:Nx;
s(j)=min(V)+(max(V)-min(V))/Nx*j;
plot(V,A,'kx',s,F11(beta,s),'k');axis([9 49 0 0.1])
```

Результаты выполнения описанной последовательности команд представлены на рис. 13.8.

***Задача* 13.2**

Аппроксимируйте зависимость $\langle\sigma_{N_{left}}\rangle/\langle N_{left}\rangle$ функциями другого вида. Для каждой функции вычислите сумму квадратов отклонений между точными значениями зависимости $\langle\sigma_{N_{left}}\rangle/\langle N_{left}\rangle$ и значениями аппроксимирующей функции в соответствующих точках. Сравните данные значения и проверьте, что для функции вида $a/\sqrt{bx+c}$ данная величина оказывается наименьшей.

Рис. 13.8. Зависимость величины относительных флуктуаций от числа частиц системы

13.3. ЭНТРОПИЯ

Результаты, полученные в вычислительных экспериментах, описанных в предыдущем разделе, демонстрируют фундаментальное свойство систем, состоящих из большого числа частиц. Оказывается, что любая изолированная статистическая система независимо от начального состояния стремится с течением времени прийти в состояние равновесия. В этом состоянии все макроскопические величины, характеризующие термодинамическое равновесное состояние системы, в пренебрежении малыми флуктуациями не зависят от времени.

Для изучения основных свойств статистических систем мы использовали модель, состоящую из относительно небольшого числа частиц, движение которых подчинялось вероятностным законам. Для выбранной модели оказывается возможным подсчитать для каждого макросостояния число возможных микросостояний, так как, во-первых, каждая частица может пребывать только в одном из двух состояний (в левой или правой половине ящика), во-вторых, местонахождение каждой частицы не зависит от положения остальных частиц. Данные условия означают, что распределение частиц подчиняется биномиальному закону. Следовательно, число возможных микросостояний Ω_n для макросостояния, в котором слева находится n, справа — n' $(n + n' = N)$, равно $N!/n!n'!$. Число микросостояний Ω_n для $N = 16$ приводятся в табл. 13.1.

Анализ данных, представленных в табл. 13.1, показывает, что максимальное число микросостояний соответствует значению $n = 8$, то есть равновесному состоянию. Так как число микросостояний является показателем «степени беспорядка» системы, можно утверждать, что равновесное состояние отвечает «состоянию максимального беспорядка». В отличие от этого макросостояние с $n = 0$ отвечает состоянию нулевого беспорядка, так как систему можно обнаружить в единственном микросостоянии. В качестве количественной характеристики степени беспорядка используют энтропию S, введенную в физику Л. Больцманом,

$$S_n = k_B \ln \Omega_n, \tag{13.6}$$

Таблица 13.1. Число микросостояний Ω_n и энтропия $S_n = \ln \Omega_n$ для $N = 16$ частиц в ящике

n	Ω_n	$\ln \Omega_n$
0	1	0
1	16	2,77
2	120	4,79
3	560	6,33
4	1820	7,51
5	4368	8,38
6	8008	8,99
7	11440	9,34
8	12870	9,46
9	11440	9,34
10	8008	8,99
11	4368	8,38
12	1820	7,51
13	560	6,33
14	120	4,79
15	16	2,77
16	1	0

где Ω_n — полное число всех возможных микросостояний, соответствующих n, k_B — постоянная Больцмана.

Данная величина определяется так, чтобы при существовании только одного микросостояния ее значение равнялась нулю и возрастало при увеличении числа микросостояний. Присутствующая в (13.6) постоянная обеспечивает соответствие между термодинамическим и статистическим определением энтропии S. Для целей наших исследований величина постоянной Больцмана не имеет принципиального значения, поэтому в дальнейшем мы будем полагать k_B равной единице, что эквивалентно измерению энтропии в единицах k_B.

Для вычисления энтропии требуется перебор всех возможных микросостояний, что в общем случае оказывается неразрешимой задачей, потому для измерения энтропии применяют метод, в котором используются наблюдаемые характеристики статистической системы. В данном методе использован тот факт, что всякая система, состоящая из большого числа частиц, в течение времени $t \to \infty$ обязательно повторит любое микросостояние (или близкое к нему микросостояние). При этом чем меньше времени проходит до совпадения двух некоррелированных микросостояний, тем меньше имеется различных микросостояний и, следовательно, меньше энтропия системы. Наблюдаемая характеристика системы, которая может быть измерена, — отношение

числа совпавших пар к полному числу проведенных сравнений микросостояний R_n (частота совпадений микросостояний). Энтропия связана с частотой совпадений микросостояний выражением

$$S_n = -\ln R_n. \tag{13.7}$$

Формула (13.7) позволяет построить следующий алгоритм вычисления энтропии, реализующийся следующей последовательностью действий:

1. Создать систему, находящуюся в состоянии равновесия.
2. Наблюдать состояние системы в течение длительного времени, осуществляя на каждом шаге по времени обмен двумя частицами, номера которых выбираются случайным образом независимо друг от друга и микросостояния в соответствующие моменты времени.
3. Вычислить число совпадающих микросостояний.
4. Вычислить значение энтропии в соответствии с (13.7).

Отметим, что формула (13.7) применима только к равновесным системам, поэтому для реализации описанного алгоритма требуется обеспечить постоянство частиц в левой и правой половинах ящика. Следовательно, вычислительный алгоритм, использованный нами в разделе 13.2, в котором число частиц в каждой половине ящика флуктуировало за счет случайных перемещений частиц, требует определенной модификации. Удовлетворить требованию нахождения системы в состоянии равновесия удается, если использовать модель, в которой система эволюционирует путем обмена одной частицы справа с одной частицей слева. При этом возникает необходимость в маркировке каждого микросостояния для их последующего сравнения и подсчета числа совпадающих микросостояний. Один из известных методов, который мы будем использовать в дальнейшем, состоит в описании каждого микросостояния числом M

$$M = \sum_{l=1}^{N_{left}} 2^l, \tag{13.8}$$

где l — номер частицы, а суммирование проводится по всем частицам, находящимся в левой половине ящика.

Для реализации описанного выше алгоритма в MATLAB создадим два файла:

1) файл `Exchange.m`, содержащий описание функции, возвращающей вектор, содержащий мгновенные значения числа;
2) файл `Entropy.m`, содержащий описание функции, возвращающей значение энтропии.

```
% листинг файла Exchange.m
function z = Exchange(NParticle,NExchange)
% функция, возвращающая мгновенные
% значения числа M
% NParticle - полное число частиц системы
% NExchange - число обменов
```

```
% задание начальной конфигурации

NL=NParticle/2;
s=0;
for i=1:NL
     L(i,1)=i;
     L(i,2)=i+NL;
     s=s+2^L(i,1);
end;
% вычисление мгновенных значений числа M
M(1)=s;
for m=1:NExchange
           Ri=floor(NL*rand(1)+1);
           Li=floor(NL*rand(1)+1);
           Rp=L(Ri,2);
           Lp=L(Li,1);
           L(Li,1)=Rp;
           L(Ri,2)=Lp;
           M(m+1)=M(m)+2.^Rp-2.^Lp;
end;
z=M;

% листинг файла Entropy.m
function z = Entropy(M)
% функция, возвращающая значение энтропии
% вектор, возвращенный функцией Exchange
K=0;
Nc=0;
N=length(M);
for i=1:N
   for j=i:N
      K=K+1;
      if M(i)==M(j)
        Nc=Nc+1;
      end;
  end;
  z=-log(Nc/K);
end;
```

Далее необходимо выполнить следующую последовательность команд, сохраненную нами в файле `Glava13_5.m`:

```
% листинг файла Glava13_5.m

N=10; % число частиц системы
Ne=4000; % число обменов
```

```
A=Exchange(N,Ne); % маркировка
                  % мгновенных
                  % микросостояний
Entropy(A) % вычисление энтропии
Ed=log(factorial(N)/factorial(N/2).^2)
Ea= log(factorial(N)/factorial(N/2).^2) % вычисление точного
                                        % значения энтропии
```

Ниже приводятся результаты вычислений, отображаемых в командном окне MATLAB, после исполнения команды `Glava13_5`:

```
Ed =
    5.3867
Ea=
    5.5294
```

***Задача* 13.3**

Модифицируйте функцию `Entropy.m` *так, чтобы она возвращала частоту совпадений R_n и энтропию S_n для каждого микросостояния частиц в ящике. Сравните полученные результаты для R_n, S_n с точными. Оцените погрешности значений R_n, S_n, найденных методом Монте-Карло. Исследуйте зависимость точности вычисляемых значений R_n, S_n для заданного числа частиц от числа перестановок. Рассмотрите случаи систем, содержащих различное число частиц N. Какое наибольшее значение N можно использовать при выбранном способе маркировки микросостояний? Объясните полученные результаты.*

***Задача* 13.4**

Модифицируйте функцию `Exchange.m` *так, чтобы она возвращала массив, содержащий номера частиц данного микросостояния, находящихся в левой и правой половинах ящика в заданные моменты времени.*

1. *Задайте функцию, вычисляющую по заданному массиву временной интервал, соответствующий появлению микросостояния, в котором нет ни одной частицы из присутствующих в начальном микросостоянии (в момент времени $t = 0$). Данное событие можно трактовать как появление первого состояния некоррелированного с начальным микросостоянием, а соответствующий временной интервал как время корреляции T_c.*

2. *Оцените значение T_c для заданного числа частиц по результатам нескольких испытаний.*

3. *Вычислите значение энтропии системы, анализируя микросостояния, достигаемые системой в моменты времени $t > T_c$.*

4. *Оцените точность вычисления значений R_n, S_n в данном случае. Рассмотрите случаи систем, содержащих различное число частиц N.*

Определение энтропии (13.8), использованное нами выше, пригодно только для равновесных статистических систем, то есть систем, находящихся в определенном макросостоянии. В то же время результаты исследования поведения системы невзаимодействующих частиц в ящике, проведенного нами в разделе 13.2, показывают, что даже в положении равновесия количество частиц n, n' в левой и правой половинах ящика меняются во времени. Существование флуктуаций (то есть фактическое отсутствие фиксированного макросостояния) определяет необходимость обобщения (13.8). Для получения обобщенной формулы заметим, что вероятность появления равновесного макросостояния, характеризующегося числом частиц n в левой половине ящика, равна P_n, а энтропия, соответственно, равна S_n. Совокупность всех возможных значений S_n для данной статистической системы можно рассматривать как реализацию некоторой случайной последовательности, а соответствующие значения P_n — как частоты их появления. Следовательно, энтропия S рассматриваемой системы может быть определена как взвешенная сумма, вычисляемая по всем возможным реализациям:

$$S = \sum_n P_n S_n. \tag{13.9}$$

Для модели частиц в ящике вероятность появления макросостояния с n частицами слева P_n может быть вычислена путем подсчета микросостояний, отвечающих данному макросостоянию

$$\frac{N!}{n!\,(N-n)\;!},$$

и делением на полное число возможных микросостояний

$$\sum_{n=0}^{N} \frac{N!}{n!\,(N-n)\;!} = 2^N.$$

***Задача* 13.5**

Вычислите аналитически P_n $(n = 0, 1, \ldots, N)$ для $N = 10, 20$.

Значение вероятности P_n можно также вычислить по последовательности состояний, подсчитав количество появлений каждого макросостояния и разделив соответствующие суммы на полное число событий. Для повышения точности следует проводить вычисление данной величины по ансамблю независимых реализаций. Напомним, что в разделе 13.2 был описан документ, позволяющий генерировать ансамбль независимых случайных реализаций при релаксации статистической системы к равновесию, вычислять усредненные значения параметров, характеризующих макроскопическое состояние системы, и оценивать время релаксации. Для нахождения значения вероятности следует создать файл `Prob.m`, содержащий описание функции, возвращающей значение вероятности обнаружить данное микросостояние, однозначно описываемое числом частиц в левой половине ящика.

```
% листинг файла Prob.m
function z = Prob(B,NParticle,TauRel)
% функция, возвращающая вероятность
```

```
% появления равновесного макросостояния
Nc=size(B,3);
Nr=size(B,2);
for k=0:NParticle
  n=0;
  for j=1:Nc
    for i=TauRel:Nr
      if B(1,i,j)==k
        n=n+1;
      end;
    end;
  end;
  N(k+1)=n;
end;
Nf=sum(N);
z=N/Nf;
```

Далее необходимо выполнить следующую последовательность команд, сохраненную нами в файле `Glava13_6.m`:

```
% листинг файла Glava13_6.m
NTrial=100; % число испытаний
NParticle=100; % число частиц системы
NStep=200; % длина единичной реализации

% генерация ансамбля
% реализаций
E=Ensemble(NTrial,NParticle,NStep);

tau=150; % момент времени, начиная с которого
         % проводится подсчет
         % числа макросостояний системы
% вычисление вероятности появления
% данного макросостояния
A=Prob(E,NParticle,tau);

% вычисление теоретического
% значения вероятности
for n=0:NParticle
     PTheory(n+1)=factorial(NParticle)/...
                  (factorial(n)*factorial(NParticle-n))/2^NParticle;
end;
% визуализация экспериментальной и
% теоретической зависимостей
% вероятности появления данного
% макросостояния от числа частиц,
```

```
% находящихся в левой половине ящика
n=0:NParticle;
plot(n,A,'k',n,PTheory,'--k')
```

Результаты выполнения описанной последовательности команд представлены на рис. 13.9.

Рис. 13.9. Зависимость вероятности появления данного макросостояния от числа частиц, находящихся в левой половине ящика (сплошная кривая — эксперимент, пунктирная кривая — теория)

***Задача* 13.6**

Сравните для заданных значений N (например, 10, 20) вычисленные и точные значения вероятности P_n. Как зависит точность численных значений P_n от числа частиц системы и объема ансамбля реализаций?

***Задача* 13.7**

Вычислите для заданных значений N равновесную энтропию, в соответствии с соотношением (13.9). При этом используйте значения вероятностей P_n, найденные в задаче 13.5, и соответствующие значения энтропии, вычисленные с помощью документа, описанного в данном разделе. Сравните точные и расчетные значения равновесной энтропии. Оцените точность расчетного значения.

***Задача* 13.8**

В функции, возвращающей значения вероятности появления равновесного макросостояния, характеризующегося числом частиц n в левой половине ящика (рис. 13.6), использован не самый эффективный с вычислительной точки зрения алгоритм, так как проводится проверка на наличие всех возможных макросостояний (цикл по переменной k). Можно существенно сократить время вычислений, если предварительно определять минимальное и максимальные отклонения числа частиц от равновесного значения и затем рассматривать только данный диапазон возможных макросостояний. Дополните текст функции `Prob.m` *соответствующим фрагментом. Убедитесь в том, что внесенные вами изменения в программу действительно привели к уменьшению времени вычислений.*

***Задача* 13.9**

Используя усредненную по ансамблю реализаций зависимость $n = \langle n(t) \rangle$, вычислите по точной формуле значения энтропии в соответствующие моменты времени. Изобразите графически зависимость энтропии от времени. Исследуйте, как зависит величина флуктуаций энтропии от числа частиц системы. Объясните полученный результат.

ЛИТЕРАТУРА К ГЛАВЕ 13

1. *Рейф Ф.* Статистическая физика. М.: Наука, 1986.
2. *Матвеев А. Н.* Молекулярная физика. М.: Высшая школа, 1981.
3. *Климонтович Ю. Л.* Статистическая физика. М.: Наука, 1982.
4. Методы Монте-Карло в статистической физике / Под ред. К. Биндера. М.: Мир, 1982.
5. *Гмурман В. Е.* Теория вероятностей и математическая статистика. М.: Высшая школа, 1998.
6. *Гулд Х., Тобочник Я.* Компьютерное моделирование в физике. М.: Мир, 1990.

Глава 14

КОМПЬЮТЕРНОЕ МОДЕЛИРОВАНИЕ МИКРОКАНОНИЧЕСКОГО АНСАМБЛЯ МЕТОДОМ МОНТЕ-КАРЛО

14.1. МИКРОКАНОНИЧЕСКИЙ АНСАМБЛЬ

Введем, следуя общепринятому в статистической физике подходу, понятие «микроканонический ансамбль». Рассмотрим для этого замкнутую систему, состоящую из N частиц. Выберем в качестве макроскопических характеристик системы объем V и полную энергию системы E, которые будем считать постоянными. Кроме того, предположим, что система является изолированной, то есть можно пренебречь влиянием на нее внешних факторов. Замкнутые макроскопические системы, как было показано в главах 10, 13, стремятся перейти в стационарное равновесное состояние, обладающее максимальной энтропией. Отметим, что если на макроскопическом уровне система характеризуется тремя величинами N, V, E, то на микроскопическом уровне существует огромное число конфигураций, реализующих заданное макросостояние N, E, V.

Таким образом, на возможные микросостояния накладывается единственное ограничение — они должны быть таковы, чтобы обеспечить заданные физические характеристики системы: число частиц N, объем V, полную энергию E. Так как причины, по которым следует предпочесть одно состояние другому, отсутствуют, можно постулировать, что система в любой момент времени может оказаться с равной вероятностью в одном из возможных микросостояний (постулат о равенстве априорных вероятностей). Это означает, что для системы с Ω достижимыми состояниями вероятность обнаружить систему в микросостоянии s равна

$$P_s = \begin{cases} \frac{1}{\Omega}, & \text{если } s \text{ — достижимо,} \\ 0, & \text{в противном случае.} \end{cases} \tag{14.1}$$

Из (14.1) видно, что сумма P_s по всем Ω равна единице.

Для определения средних значений физических величин, характеризующих макроскопическую систему, можно использовать два способа. В первом способе измеряют физические величины в течение достаточно большого промежутка времени, за который система успевает побывать в большом числе достижимых микросостояний, и проводят последующее усреднение. Данный подход был уже использован нами в главе 10, когда методом молекулярной динамики рассчитывались средние значения температуры и давления неидеального газа. С точки зрения проведенного усреднения во времени, вероятность P_s есть доля времени, в течение которого система за время наблюдений находится в данном микроскопическом состоянии s.

Несмотря на очевидный физический смысл средних по времени, в статистической физике вводят понятие статистических средних в данный момент времени, которые вычисляются по ансамблю микросостояний, соответствующих заданному макросостоянию. (Гипотеза о совпадении средних значений величин, вычисляемых усреднением по времени и усреднением по ансамблю реализаций, называется эргодической гипотезой.) Полное число систем в ансамбле равно числу возможных микросостояний. Вероятность обнаружить в ансамбле тождественных систем s-ю реализацию, определяется выражением (14.1). Ансамбль систем, характеризуемый величинами E, T, V и описываемый распределением вероятностей вида (14.1), называется микроканоническим ансамблем.

Будем считать, что система характеризуется некоторой величиной A, значение которой в s-м состоянии равно A_s. Тогда среднее значение A по ансамблю определяется как взвешенное среднее

$$\langle A \rangle = \sum_s A_s P_s. \tag{14.2}$$

Из (14.2) видно, что для нахождения среднего по ансамблю значения величины необходимо знать значения вероятностей P_s. При небольшом числе частиц значения вероятностей можно определить методом перебора, аналогично тому, как это сделано в главе 12 при моделировании случайных блужданий. Однако при увеличении числа частиц данный метод становится неприемлемым из-за большого объема вычислений. В этих условиях приходится использовать приближенные методы, среди которых одним из самых эффективных является метод Монте-Карло.

14.2. МОДЕЛИРОВАНИЕ МИКРОКАНОНИЧЕСКОГО АНСАМБЛЯ

Общая идея метода Монте-Карло для микроканонического ансамбля состоит в получении репрезентативной выборки из полного числа микросостояний и оценке с ее использованием значения вероятности P_s. Для этого можно использовать достаточно очевидную процедуру: зафиксировать полное число частиц N и объем системы V, изменять случайным образом координаты и скорости отдельных частиц, принимать конфигурации микросостояний, имеющих заданную полную энергию. К сожалению, данная процедура оказывается весьма неэффективной, так как большинство конфигураций вообще не будут иметь требуемую полную энергию и должны быть отвергнуты.

Эффективная вычислительная процедура метода Монте-Карло для микроканонического ансамбля была предложена в [1]. Она состоит в разделении исходной макроскопической системы на две подсистемы: исходную, называемую далее системой, и подсистему, состоящую из одного элемента, которую авторы назвали демоном (по аналогии с демоном Максвелла). Роль демона аналогична роли члена кинетической энергии в методе молекулярной динамики, обсуждавшегося в главе 10. Обходя элементы системы и передавая энергию, он обеспечивает изменение конфигурации системы. Если энергии, запасенной в мантии демона оказывается достаточно, он отдает энергию тому элементу системы, которому требуется энергия, для осуществления изменения конфигурации. И наоборот, если для изменения конфигурации требу-

ется уменьшить энергию системы, энергия передается частицей демону. Единственное ограничение состоит в том, что энергия демона должна быть положительной. Описанная процедура алгоритмически выглядит следующим образом.

1. Сгенерировать начальную конфигурацию системы с заданным значением полной энергии.

2. Выбрать случайным образом частицу и произвести пробное изменение ее координат.

3. Вычислить полную энергию системы в новом состоянии.

4. Если в новом состоянии энергия системы оказывается меньше, то система отдает энергию демону и новая конфигурация принимается.

5. Если в новом состоянии энергия системы оказывается больше, то новая конфигурация принимается, если энергии демона достаточно, чтобы передать ее системе. В противном случае новая конфигурация не принимается и частица сохраняет свои старые координаты.

6. Если пробное изменение не меняет энергию системы, то новая конфигурация принимается.

Перечисленные выше действия повторяются до получения репрезентативной выборки микросостояний. Можно ожидать, что через некоторый промежуток времени система достигнет равновесного состояния, в котором у системы и демона будут некоторые средние (для каждого) значения энергии. При этом значение полной энергии системы остается постоянной. Так как демон представляет собой только одну степень свободы, в отличие от рассматриваемой статистической системы можно ожидать, что флуктуации системы будут невелики.

Применим описанный алгоритм к классическому одномерному идеальному газу, скорости частиц которого непрерывны и неограниченны, а энергия частиц не зависит от положения частиц, поэтому полная энергия частиц есть сумма кинетических энергий отдельных частиц. Для изменения конфигурации будем случайным образом изменять скорость случайно выбранной частицы.

Для реализации описанного алгоритма в MATLAB создадим два m-файла:

1) файл `Init.m`, содержащий описание функции, возвращающей абсолютные значения скоростей в момент времени $t = 0$;

2) файл `Demon.m`, содержащий описание функции, возвращающей усредненные по ансамблю реализаций значения: энергии демона (`Edave`), средней энергии системы на одну частицу (`Esysave`), средней скорости на одну частицу системы (`Vave`), среднего числа принятия (`Accept`).

```
% листинг файла Init.m
function z = Init(N,Esystem)
% функция, возвращающая
% абсолютные значения
% скоростей в момент
```

```
% времени t = 0
% N - число частиц системы
% Esystem - энергия системы
V=(Esystem./N).^0.5;
for i=1:N
   z(i)=V;
end;

% листинг файла Demon.m
function [Edave,Esysave,Vave,Accept] = Demon(N,Esystem,NTrial,Vel,dV)
% функция, возвращающая усредненные
% по ансамблю реализаций
% значения: энергии демона (Edave),
% средней энергии системы на одну
% частицу (Esysave), средней скорости на одну
% частицу системы (Vave),
% среднего числа принятия (Accept)

Edemon=0;
Vtot=sum(Vel);
Vcum=0;
Escum=0;
Edcum=0;
Accept=0;
for j=1:NTrial
   for i=1:N
      % случайное изменение скорости
      dv=(2*rand(1)-1)*dV;

      % случайный выбор частицы
      Ip=floor(N*rand(1)+1);

      % пробное изменение скорости
      VTrial=Vel(Ip)+dv;

      % пробное изменение энергии
      de=0.5*(VTrial.^2-Vel(Ip)^2);

      % если энергия уменьшается
      % изменение принимается
      if de<=Edemon
         Vel(Ip)=VTrial;
         Vtot=Vtot+dv;
         Accept=Accept+1;
         Edemon=Edemon-de;
         Esystem=Esystem+de;
```

```
      end;
      Edcum=Edcum+Edemon;
      Escum=Escum+Esystem;
      Vcum=Vcum+Vtot;
   end;
end;
Edave=Edcum/(NTrial*N);
Accept=Accept/(NTrial*N);
Esysave=1/N*Escum/(NTrial*N);
Vave=1/N*Vcum/(NTrial*N);
```

Далее необходимо выполнить следующую последовательность команд, сохраненную нами в файле `Glava14_1.m`:

```
% листинг файла Glava14_1.m
Esystem=40; % энергия системы
N=40; % число частиц системы
NTrial=4000; % число испытаний
dV=2*2^0.5; % максимальное значение
            % изменения скорости
Vel=InitD(N,Esystem); % задание
                      % начальных скоростей
% вычисление характеристик
% состояния идеального газа
[Edave Esysave Vave Accept]=...
    Demon(N,Esystem,NTrial,Vel,dV);
Edave
Esysave
Accept
```

Ниже показан результат приведенной выше последовательности команд.

```
Edave =
    0.9588
Esysave =
    0.9760
Accept =
    0.5080
```

***Задача* 14.1**

Проведите численные расчеты для заданных значений числа частиц $N = 10, 20, 40, 80$ *и различных значений отношения* $dV \Big/ \sqrt{\frac{Esystem}{N}}$. *Изобразите графически зависимости коэффициента принятия (*`Accept`*) от* $dV \Big/ \sqrt{\frac{Esystem}{N}}$

при заданном числе частиц. Используя метод наименьших квадратов, подберите аналитическое выражение, описывающее зависимость коэффициента принятия (`Accept`*) от* $dV/\sqrt{\frac{Esystem}{N}}$*. При каком отношении значение коэффициента принятия оказывается* $\cong 0,5$*?*

***Задача* 14.2**

Модифицируйте функцию `Demon.m`*, листинг которой приведен выше, так, чтобы номера частиц, скорость которых подлежит изменению (переменная* `Ip`*), выбирались последовательно. Приводит ли изменение способа выбора номера частицы к изменению средней энергии системы на частицу и средней энергии демона?*

***Задача* 14.3**

Можно ввести кинетическое определение температуры, в соответствии с которым температура определяется из соотношения

$$\frac{1}{2}m\left\langle v^2\right\rangle = \frac{1}{2}T_{kin}, \tag{14.3}$$

где $\frac{1}{2}m\left\langle v^2\right\rangle$ *— средняя кинетическая энергия, приходящаяся на частицу,* T_{kin} *— температура газа, измеряемая в единицах* k_B*. Модифицируйте функцию* `Demon.m` *так, чтобы в программе дополнительно вычислялось и дополнительно выводилось в векторе, возвращаемом данной функцией, значение средней кинетической энергии, приходящейся на одну частицу. Используя соотношение (14.3), получите* T_{kin}*. Как связаны значения температуры и энергии демона?*

Один из наиболее важных результатов, полученных в задаче 14.3, состоит в демонстрации возможности введения кинетического определения температуры как средней на одну частицу кинетической энергии частицы микроканонического ансамбля. При этом значение температуры с некоторой погрешностью оказывается равным энергии демона. В то же время микроканонический ансамбль нельзя считать абсолютно адекватной моделью реальных статистических систем, так как данные системы не являются термодинамически изолированными, но находятся в тепловом контакте с окружающей средой. Это приводит к возникновению потока тепла между изучаемой системой и окружающей средой. Так как размер исследуемой системы обычно много меньше размера окружающей среды, первую будем называть микросистемой, а последнюю — тепловым резервуаром (термостатом). При этом закон сохранения полной энергии относится к составной системе, состоящей из микросистемы и теплового резервуара.

Рассмотрим большое число воображаемых копий микросистемы и теплового резервуара. Микросистема и тепловой резервуар, рассматриваемые как единое целое, являются изолированной термодинамической системой и могут быть описаны с помощью микроканонического ансамбля. Так как в данном случае наибольший интерес представляют равновесные значения физических величин, описывающих микросистему, необходимо знать вероятность P_s, с которой микросистема обнаруживается в состоянии s с энергией E_s. Ансамбль, который описывает распределение вероятностей состояний микросистемы, находящейся в термодинамическом равновесии с тепловым резервуаром, называется каноническим.

В общем случае в качестве микросистемы может выступать любая макроскопическая система, размер которой меньше теплового резервуара, в том числе и отдельная частица, выделенная среди других частиц теплового резервуара. Примером такой микросистемы служит демон, которого можно считать микросистемой, микроканоническое состояние которой определяется только ее энергией. Таким образом, метод отыскания распределения вероятностей в каноническом ансамбле заключается в том, чтобы выполнить численное моделирование демона, который обменивается энергией с идеальным газом, состоящим из N частиц. Идеальный газ в данном случае играет роль термостата и задача состоит в определении плотности вероятности $P(E_d)$ того, что демон имеет энергию E_d. Вычислительный алгоритм, позволяющий решить данную задачу, достаточно очевиден:

1. Провести моделирование канонического ансамбля методом Монте-Карло, сохраняя на каждом шаге Монте-Карло значение энергии демона.
2. Для найденной реализации значений энергии демона вычислить распределение вероятностей $P(E_d)\Delta E$.
3. Определить функциональную зависимость $P(E_d)\Delta E$ от температуры.

Для реализации данного алгоритма в MATLAB создадим m-файл `DemonE.m`, содержащий описание функции, возвращающей мгновенные (на каждом шаге Монте-Карло) значения энергии демона.

```
% листинг файла DemonE.m
function z = DemonE(N,Esystem,NTrial,Vel,dV)
% функция, возвращающая мгновенные
% значения энергии демона
% N - число частиц системы
% Esystem - энергия системы
% NTrial - число испытаний
% Vel - вектор, возвращенный функцией InitD
% dV - максимальное изменение скорости
Edemon=0;
z(1)=Edemon;
k=1;
for j=1:NTrial
   for i=1:N
      dv=(2*rand(1)-1)*dV; % случайное изменение скорости
      Ip=floor(N*rand(1)+1); % случайный выбор номера частицы
      Vtrial=Vel(Ip)+dv;
      de=0.5*(Vtrial.^2-Vel(Ip).^2); % пробное изменение энергии
      if de<=Edemon % если энергия уменьшается,
                    % то пробное изменение
                    % принимается
         Vel(Ip)=Vtrial;
         Edemon=Edemon-de;
      end;
      k=k+1;
```

```
      z(k)=Edemon;
   end;
end;
```

Далее необходимо выполнить следующую последовательность команд, сохраненную нами в файле Glava14_2.m:

```
% листинг файла Glava14_2.m
clear all
Esystem=40; % энергия системы
N=40; % число частиц системы
NTrial=500; % число испытаний
dV=2*2^0.5; % максимальное изменение скорости

% вычисление начальных скоростей частиц
Vel=InitD(N,Esystem);

% вычисление мгновенных
% значений энергии демона
Ed=DemonE(N,Esystem,NTrial,Vel,dV);

% визуализация зависимости
% мгновенных значений энергии демона
% от времени
i=1:length(Ed);
figure(1);
plot(i,Ed,'k');
axis([0 20000 0 8])

% вычисление и визуализация
% распределения вероятностей
x1=min(Ed);
x2=max(Ed);
Nint=50;
i=1:Nint;
x=x1+(x2-x1)/(Nint-1)*(i-1);
h=hist(Ed,x);
figure(2);
bar(x,h);
colormap white

% вычисление и визуализация функции,
% аппроксимирующей распределение

% задание аппроксимирующей
% функции
```

```
F10=inline('u(1)*exp(-u(2)*z)','u','z')

% вычисление коэффициентов
% аппроксимирующей функции
beta=nlinfit(x,h,F10,[3000 1]);

% визуализация гистограммы
% последовательности мгновенных значений
% энергии демона и аппроксимирующей функции
Ni=500;
i=1:Ni;
X(i)=x1+(x2-x1)/(Ni-1)*(i-1);
figure(3);
bar(x,h);
colormap white

hold on
plot(X(i),F10(beta,X(i)),'k')
```

Результаты выполнения описанной последовательности команд представлены ниже на рис. 14.1—14.3.

```
F10 =

 Inline function:
 F10(u,z) = u(1)*exp(-u(2)*z)
```

Рис. 14.1. Зависимость мгновенных значений энергии демона от времени (номера шага Монте-Карло)

Рис. 14.2. Распределение по энергии демона $N \cdot P(E_d)\Delta E$

Таким образом, результаты моделирования идеального газа методами микроканонического и канонического ансамблей позволяют сделать следующие выводы:

Рис. 14.3. Функция, аппроксимирующая распределение по энергии демона $N \cdot P(E_d)\Delta E$

1. Распределение вероятностей энергии демона описывается функцией вида

$$P\left(E_d\right) = Ae^{-\frac{E_d}{T_{term}}}, \tag{14.4}$$

где A — постоянный коэффициент, $T_{term} \approx 0,99013$.

2. Температура идеального газа T_{kin}, определяемая как средняя кинетическая энергия, приходящаяся на одну частицу, совпадает в пределах точности модели с температурой термостата T_{term}, что позволяет сделать вывод о том, что демон находится в термодинамическом равновесии с термостатом.

Распределение вероятности (14.4) называется распределением Больцмана или каноническим распределением. Данное распределение принято записывать в следующем виде:

$$P\left(E\right) = \frac{1}{Z}e^{-\frac{E}{k_b T}}, \tag{14.5}$$

где Z — нормировочный множитель, выбираемый из условия равенства единице суммы по всем состояниям демона, $k_B = 1,38 \cdot 10^{-16}$ *эрг·град*$^{-1}$ — постоянная Больцмана.

Параметр T называется абсолютной температурой. Температура может измеряться в единицах k_B, тогда распределение Больцмана принимает вид

$$P\left(E\right) = \frac{1}{Z}e^{-\frac{E}{T}}. \tag{14.6}$$

Формулы (14.5), (14.6) обеспечивают простой способ вычисления температуры по средней энергии демона E_d, которая по определению равняется

$$\langle E_d \rangle = \frac{\int Ee^{-\frac{E}{T}}dE}{\int e^{-\frac{E}{T}}dE} = T. \tag{14.7}$$

Таким образом, средняя энергия демона равняется температуре термостата. Данный результат справедлив, только при условии, что значения энергии принимают непрерывные значения.

Отметим, что формулы (14.6), (14.7) применимы для любой микросистемы, находящейся в тепловом равновесии с термостатом, — канонического ансамбля, при этом всякое макросостояние задается температурой T, числом частиц системы N и объемом V. В отличие от канонического ансамбля микроканонический ансамбль характеризуется энергией E, числом частиц системы N и объемом V.

***Задача* 14.4**

Исследуйте зависимость точности определения температуры, определяемой с помощью микроканонического и канонического ансамблей от числа частиц системы N и числа шагов метода Монте-Карло.

14.3. МОДЕЛЬ ИЗИНГА

Одной из простейших моделей, используемых в статистической физике для моделирования фазовых переходов в магнитных веществах или бинарных составах, является модель Изинга. Данная модель относится к широкому классу решеточных моделей, в которых рассматриваются локальные взаимодействия, то есть взаимодействия между ближайшими узлами решетки. В магнитных системах локальные взаимодействия обусловлены спинами, расположенными в узлах решетки. Спины могут представлять собой, например, магнитные моменты атомов в твердом теле, взаимодействующие друг с другом и внешним магнитным полем.

Рассмотрим решетку, состоящую из N узлов. Свяжем с каждым i-м узлом решетки число $s_i = \pm 1$, характеризующее направление магнитного момента системы, где $s_i = +1$, если спин ориентирован в положительном направлении оси OZ, и $s_i = -1$, если спин ориентирован в отрицательном направлении оси OZ. (Данная картина характерна для частиц с полуцелым спином, хотя далее мы рассматриваем спины как классические степени свободы и не вводим для них используемых в квантовой механике правил коммутации углового момента.)

Любое микросостояние решетки задается набором переменных

$$\{s_1, s_2, \ldots, s_N\}.$$

Так как макроскопическое свойства системы определяются свойствами ее достижимых микросостояний, необходимо вычислить зависимость энергии E от конфигурации спинов. Полная энергия при наличии магнитного поля h в модели Изинга равняется

$$E = -J \sum_{\langle i,j \rangle}^{N} s_i s_j - h \sum_{i=1}^{N} s_i, \tag{14.8}$$

где обозначение $\langle i, j \rangle$ означает, что сумма берется по всем ближайшим соседним парам спинов, константа обменной связи J характеризует силу взаимодействия соседних спинов (рис. 14.4).

Если $J > 0$, то в состоянии $\uparrow\uparrow$ и $\downarrow\downarrow$, то есть при одинаковой ориентации спинов ближайших соседей энергетически выгоднее состояния $\uparrow\downarrow$ и $\downarrow\uparrow$, у которых соседние спины ориентированы в противоположные стороны. Следовательно, можно ожидать, что

Рис. 14.4. Энергия взаимодействия между ближайшими соседними спинами в отсутствие магнитного поля

для $J > 0$ состояние с наименьшей полной энергией является ферромагнитным, то есть среднее число спинов, сориентированных в одном направлении, не равно нулю. Если $J < 0$, то с энергетической точки зрения более предпочтительными оказываются состояния $\uparrow\downarrow$, $\downarrow\uparrow$, для которых соседние спины антипараллельны. Следовательно, среднее число спинов, сориентированных в одном направлении, равно нулю, то есть спины упорядочены через один (антиферромагнитное состояние). При наложении внешнего магнитного поля, направленного параллельно оси OZ, спины $\uparrow$ и $\downarrow$ приобретают дополнительную внутреннюю энергию, равную $-h$ и $+h$ соответственно.

Выделим основные упрощающие предположения, положенные в основу модели Изинга:

1. Кинетическая энергия узлов решетки принимается равной нулю.
2. В выражении, описывающем энергию взаимодействия, учитывается только вклад от ближайших соседей и предусматриваются только два дискретных состояния для спинов.

Так как в дальнейшем нас будут интересовать термодинамические характеристики данной системы, оказывается удобным измерять энергии связей J и h в единицах температуры. Тогда нагревание системы будет приводить к ослаблению этих связей. Возможные конфигурации системы определяются заданием значений всех спиновых переменных, число которых составляет 2^N, а вклад любой из 2^N спиновых конфигураций s определяется функцией распределения для канонического ансамбля

$$w(s) = \frac{e^{-E(s)}}{Z}, \tag{14.9}$$

где

$$Z(J,h) = \sum_{i}^{N} e^{-E(s_i)}. \tag{14.10}$$

Предваряя применения алгоритма демона к исследованию модели Изинга методом микроканонического ансамбля, необходимо получить выражение, связывающее энергию демона и температуру термостата. Напомним, что в непрерывном случае распределение энергии демона подчинялось формуле Больцмана (14.6). Предположим, что данное распределение вероятностей справедливо для любой макроскопической системы, находящейся в состоянии термодинамического равновесия. Тогда $\langle E_d \rangle$ равно

$$\langle E_d \rangle = \frac{\sum E_d e^{-\frac{E_d}{T}}}{\sum e^{-\frac{E_d}{T}}}, \tag{14.11}$$

где суммы вычисляются по всем возможным значениям E_d.

Минимальная ненулевая потеря энергии системы в нулевом магнитном поле составляет, как видно из рис. 14.5, $2sJ$, где s — суммарный спин ближайших соседей, опрокидывающегося спина.

Рис. 14.5. Изменение энергии, обусловленное изменениями ориентации центрального спина

В одномерном случаем суммарный спин ближайших соседей равен 0, или 2, то есть минимальная ненулевая потеря энергии равняется $2J$. Следовательно, энергия демона, может равняться $0, 2J, 4J, \ldots$ Если ввести обозначение $x = 2J/T$, то выражение (14.11) для бесконечной решетки принимает вид

$$\left\langle \frac{E_d}{T} \right\rangle = \frac{\sum\limits_{n=0}^{\infty} (x \cdot n)\, e^{-n \cdot x}}{\sum\limits_{n=0}^{\infty} e^{-n \cdot x}}. \tag{14.12}$$

Бесконечные суммы, стоящие в числителе и знаменателе, могут быть вычислены аналитически. Так как обсуждение методов их вычисления выходит за рамки нашей книги, мы используем для ее вычисления встроенный в MATLAB символьный процессор[1]:

$$\sum_{n=0}^{\infty} (x \cdot n)\, e^{-n \cdot x} \to x \cdot \frac{\exp(-x)}{(1 - \exp(-x))^2}, \tag{14.13}$$

$$\sum_{n=0}^{\infty} e^{-n \cdot x} \to \frac{1}{1 - \exp(-x)}. \tag{14.14}$$

Разделив (14.13) на (14.14) и выполнив очевидное преобразование, получаем

$$\left\langle \frac{E_d}{T} \right\rangle = \frac{x}{e^x - 1}. \tag{14.15}$$

Подставив в (14.15) вместо x выражение $2J/T$ и решив получившееся уравнение относительно T, получаем

$$T = \frac{2J}{\ln\left(1 + 2J/\langle E_d \rangle\right)}. \tag{14.16}$$

[1] Сумма бесконечного ряда (14.11) находится выполнением следующей последовательности команд:

```
» expr=sym('x*n*exp(-x*n)') % задание выражения, стоящего под знаком
                            % суммы
  expr =
  x*n*exp(-x*n)
» symsum(expr,'n',0,inf) % вычисление суммы ряда
  ans =
  x*exp(-x)/(1-exp(-x))^2
```

Аналогично находится сумма бесконечного ряда (14.4).

***Задача* 14.5**

Получите аналитические выражения для средней энергии демона для системы, состоящей из конечного числа спинов.

1. *Может ли быть найдено аналитическое выражение для зависимости температуры от энергии демона в данном случае?*
2. *Сравните значения средней энергии демона для системы, состоящей из N спинов $\langle E_d \rangle_N$, получаемые по точной формуле с соответствующими значениями $\langle E_d \rangle_\infty$, получаемыми по формуле (14.15).*
3. *Постройте зависимости $|\langle E_d \rangle_N - \langle E_d \rangle_\infty|$ при различных температурах.*
4. *Оцените погрешность определения значения температуры, получаемого по формуле (14.16), для системы, состоящей из N спинов. Как зависит данная погрешность от N и T?*

Отметим, что в одномерной модели Изинга демон должен выбирать спины случайно, что позволит избежать подсчета периодически повторяющихся конфигураций. Так как нашей целью является оценка термодинамических характеристик бесконечной термодинамической системы, следует учесть краевые условия. В качестве таковых в описываемом ниже документе нами выбраны периодические (тороидальные) краевые условия: решетка считается кольцом, в котором спины s_i, находящиеся в узлах $i = 1$ и $i = N$, взаимодействуют друг с другом. Это обеспечивает равное число взаимодействий для всех спинов в изучаемой системе.

Для моделирования одномерной модели Изинга методом микроканонического ансамбля в MATLAB необходимо создать файл `Ising.m`, содержащий описание функции, возвращающей значения полной энергии системы, энергии демона, намагниченности и среднее число принятия решений.

```
% листинг файла Ising.m
function [Es,Ed,SpM,Accept]=Ising(Nspin,J,h,Esi,NTrial)
% функция, возвращающая мгновенные значения
% полной энергии системы(Es),
% энергии демона (Ed),
% намагниченность (SpM)
% и среднее число принятия (Accept)
% Nspin - число спинов системы
% J - константа обменного взаимодействия
% h - внешнее магнитное поле
% Esi - конечная энергия системы
% NTrial - число испытаний
% задание конфигурации спинов
% в момент времени t = 0

for i=1:Nspin
  s(i)=1;
end;
```

```
M=Nspin;
Esystem=-(J+h)*Nspin;
Edemon=2*J*ceil((Esi-Esystem)/(2*J));
Es(1)=Esystem; % энергия системы
               % в момент времени t = 0
Ed(1)=Edemon; % энергия демона
              % в момент времени t = 0
SpM(1)=M; % магнитный момент системы
          % в момент времени t = 0
Accept=0;
k=1;
% реализация метода микроканонического ансамбля
for i=1:NTrial
  for j=1:Nspin
    Ispin=floor(Nspin*rand(1)+1); % случайный выбор
                                  % номера спина
    % периодические граничные условия
    if Ispin==1
       Left=s(Nspin);
    else
       Left=s(Ispin-1);
    end;
    if Ispin==Nspin
      Right=s(1);
    else
      Right=s(Ispin+1);
    end;
    de=2*s(Ispin)*(-h+J*(Left+Right)); % пробное изменение
                                       % энергии спина
    if de<=Edemon % принятие
                  % пробного изменения
       s(Ispin)=-s(Ispin);
       Accept=Accept+1;
       Edemon=Edemon-de;
       Esystem=Esystem+de;
    end;
    k=k+1;
    Es(k)=Esystem;
    Ed(k)=Edemon;
    SpM(k)=sum(s);
  end;
end;
Accept=Accept/(NTrial*Nspin);
```

Далее необходимо выполнить следующую последовательность команд, сохраненную нами в файле `Glava14_3.m`:

```
% листинг файла Glava14_3.m
Nspin=200; % число спинов системы
J=1; % константа обменного взаимодействия
h=0; % внешнее поле
Esi=-10; % задание конечного значения энергии системы
NTrial=100; % число испытаний

% вычисление мгновенных значений
% полной энергии системы (Es),
% энергии демона (Ed),
% мгновенной намагниченности (SpM),
% числа принятия решений (Accept)
[Es Ed SpM Accept]=Ising(Nspin,J,h,Esi,NTrial);

% визуализация зависимостей мгновенных
% значений полной энергии
% системы и энергии демона от времени
i=1:1200;
figure;
plot(i,Es(i), 'k');

figure;
plot(i,Ed(i), 'k');
% вычисление распределения вероятностей
% мгновенных значений энергии демона
Nint=50;
i=1:Nint;
x1=min(Ed);
x2=max(Ed);
x=x1+(x2-x1)/Nint*i;
h=hist(Ed,x);

% нахождение и визуализация функции,
% описывающей распределение вероятности
F10=inline('u(1)*exp(-u(2)*z)','u','z')
beta=nlinfit(x,h,F10,[10000 0.05]);

figure
bar(x,h);
colormap white
hold on

Ni=500;
j=1:Ni;
X(j)=x1+(x2-x1)/Ni*j;
```

```
plot(X,F10(beta,X),'k')
hold off

% вычисление средней намагниченности на
% один спин
Emean=mean(Es)/Nspin

% вычисление средней энергии на
% один спин
Mean=mean(SpM)/Nspin

% вычисление температуры системы
Tmean=2/log(1+2./mean(Ed))
```

Ниже представлены результаты выполнения описанной последовательности команд.

1. Сообщения, выводимые в командное окно:

```
      F10 =
       Inline function:
       F10(u,z) = u(1)*exp(-u(2)*z)
  Emean =
     -0.0928
  Mean =
   -2.5549e-004
  Tmean =
      9.5313
```

2. Графики, вычисленных зависимостей (рис. 14.6—14.8).

Рис. 14.6. Зависимость мгновенных значений полной энергии системы от времени

Рис. 14.7. Зависимость мгновенных значений энергии демона от времени

Рис. 14.8. Функция распределения энергии демона

***Задача* 14.6**

Так как изначально неизвестны конфигурации спинов, обеспечивающие искомое значение энергии, в качестве начальной конфигурации мы выбираем конфигурацию с минимальной энергией, поэтому для достижения равновесного состояния с заданной полной энергией системе требуется некоторое конечное время (время релаксации). Это хорошо видно из рис. 14.6, 14.7.

1. *Аппроксимируйте зависимости E_{system}, E_{demon} методом наименьших квадратов.*

2. *Оцените время релаксации полной энергии системы τ_{system} и время релаксации энергии демона τ_{demon}. Как зависят τ_{system}, τ_{demon} от числа спинов* Nspin, *образующих систему?*

3. *Сравните полученные результаты, с результатами, полученными при моделировании случайных блужданий.*

***Задача* 14.7**

Создайте собственные MATLAB-функции, позволяющие исключать переходный период и вычислять равновесные значения средней полной энергии системы $\langle E_{system} \rangle$, среднюю энергию демона $\langle E_{demon} \rangle$, среднюю намагниченность системы $\langle M \rangle$, средний квадрат намагниченности $\langle M^2 \rangle$. Сравните значения равновесной температуры системы с соответствующим значением температуры, полученной без учета процесса релаксации системы к равновесному состоянию.

***Задача* 14.8**

После достижения равновесного состояния значения полной энергии системы и энергии демона, как видно из рис. 14.6, 14.7, испытывают случайные флуктуации около средних значений. Исследуйте зависимость величины

флуктуаций средней полной энергии системы $\langle E_{system}\rangle$, средней энергии демона $\langle E_{demon}\rangle$, средней намагниченности системы $\langle M\rangle$, среднего квадрата намагниченности $\langle M^2\rangle$ от Nspin. Сравните полученные результаты с результатами, полученными при моделировании случайных блужданий.

Задача 14.9

Вычислите $\langle E_{system}\rangle$ для систем, состоящих из различного числа спинов N. Сравните полученные результаты с точным ответом для бесконечной одномерной решетки

$$\frac{\langle E_{system}\rangle}{N} = -th\left(\frac{J}{T}\right). \tag{14.17}$$

Как полученные результаты зависят от числа спинов N и количества шагов метода Монте-Карло?

Задача 14.10

Проведите вычисления и постройте зависимость $\langle M^2\rangle$ от температуры T для системы, состоящей из N спинов ($N = 200$). Аппроксимируйте данную зависимость методом наименьших квадратов. Зависит ли вид аппроксимирующей функции от числа спинов T?

Задача 14.11

Введите ненулевое магнитное поле h и вычислите $\langle E_{demon}\rangle$, $\langle M\rangle$, $\langle M^2\rangle$, как функции от h, для заданной полной энергии. Определите связь $\langle E_{demon}\rangle$с температурой для $h \neq 0$. Сравните значения равновесных температур для случаев $h = 0$, $h \neq 0$ при одинаковых значениях средней полной энергии.

Обобщим метод микроканонического ансамбля, использованный нами для одномерной системы спинов, на случай двумерной системы (рис. 14.9).

Как и в одномерном случае, выбираем периодические (тороидальные) краевые условия, то есть будем представлять решетку кольцом, в котором спины s_{ij}, находящиеся в узлах ($i = 1$, $j = 1, 2, \ldots, N$), взаимодействуют со спинами, находящимися в узлах ($i = N, j = 1, 2, \ldots, N$), а спины, расположенные в узлах ($i = 1, 2, \ldots, N, j = 1$), взаимодействуют со спинами, расположенными в узлах ($i = 1, 2, \ldots, N, j = N$) (см. рис. 14.10).

Рис. 14.9. Квадратная двумерная решетка, состоящая из $N = 16$ спинов

Это обеспечивает равное число взаимодействий для всех спинов в изучаемой системе. Взаимодействие выбранного спина с соседями можно рассматривать как взаимодействие с одним спином, величина которого равна алгебраической сумме величин четырех соседних спинов, которая, как очевидно, может принимать три значения: 0, 2, 4. Следовательно, минимально возможное значение изменения энергии при опрокидывании одного спина составляет $2J - (-2J) = 4J$, в отличие от одномерного случая, в котором квант изменения энергии равнялся $2J$. Отмеченное

Рис. 14.10. Задание граничных условий для квадратной двумерной решетки, состоящей из $N = 16$ спинов

обстоятельство определяет необходимость замены в (14.16) $2J \to 4J$:

$$T = \frac{4J}{\ln\left(1 + 4J/\langle E_d \rangle\right)}. \tag{14.18}$$

Алгоритм моделирования двумерной системы, состоящей из N спинов, методом Монте-Карло реализуется следующей последовательностью действий:

1. Задание числа спинов решетки $Nspin$.
2. Задание числа шагов метода Монте-Карло на спин.
3. Задание ориентации спинов в узлах квадратной решетки в момент времени $t = 0$ (начальной конфигурации системы).
4. Выбор случайным образом одного из спинов системы.
5. Вычисление пробного изменения энергии.
6. Если пробное изменение приводит к уменьшению энергии системы, то система отдает энергию демону и новая конфигурация принимается.
7. Если пробное изменение увеличивает энергию системы, то новая конфигурация принимается в том случае, если демон имеет достаточную энергию для передачи ее системе.
8. Если пробное изменение не меняет энергию системы, то принимается новая конфигурация.
9. Повторение п. 4—8 (число повторений равно числу спинов в системе).
10. Повторение п. 4—9 (число повторений равно числу шагов метода Монте-Карло на спин).

Для реализации описанного алгоритма моделирования двумерной системы спинов методом Монте-Карло в MATLAB создадим файл `Ising2.m`, содержащий описание функции, возвращающей мгновенные значения: энергии системы, энергии демона, намагниченности, числа принятия решений, а также мгновенные конфигурации спинов.

```
% листинг файла Ising2.m
function [Es,Ed,SpM,A,S] = Ising2(Nspin,J,h,Esi,NTrial)
% функция, возвращающая мгновенные
% значения:
% энергии системы (Es),
% энергии демона (Ed),
% намагниченности (SpM),
% числа принятия решений (A),
% а также мгновенные конфигурации спинов (S)
% Nspin - число спинов решетки
% J - константа обменного взаимодействия
% h - напряженность внешнего магнитного поля
% Esi - конечная энергия системы
% NTrial - число испытаний

Ns=Nspin.^0.5; % число спинов вдоль
               % одной стороны
s=ones(Ns,Ns); % начальная конфигурация
               % спинов
Esystem=-(J+h)*Nspin; % начальная энергия
                      % системы
% начальная энергия демона
Edemon=4*J*floor((Esi-Esystem)/(4*J));

Es(1)=Esystem;
Ed(1)=Edemon;

S=s;
k=1;
for i=1:NTrial
  Accept=0;
  for j=1:Nspin
     % случайный выбор узла сетки
     Ix=floor(Ns*rand(1)+1);
     Iy=floor(Ns*rand(1)+1);
     % граничные условия
     if Ix==1
        Left=Ns;
     else
        Left=Ix-1;
     end;
```

```
      if Ix==Ns
        Right=1;
      else;
         Right=Ix+1;
      end;
      if Iy==1
         Down=Ns;
      else;
         Down=Iy-1;
      end;
      if Iy==Ns
         Up=1;
      else
         Up=Iy+1;
      end;
      % пробное изменение энергии
      de=2*s(Iy,Ix)*...
            (-h+...
             J*(s(Iy,Left)+...
             s(Iy,Right)+...
             s(Down,Ix)+...
             s(Up,Ix)));
      if de<=Edemon % принятие пробного
                    % изменения энергии
         s(Iy,Ix)=-s(Iy,Ix);
         Accept=Accept+1;
         Edemon=Edemon-de;
         Esystem=Esystem+de;
      end;
      k=k+1;
      Es(k)=Esystem;
      Ed(k)=Edemon;
      A(k-1)=Accept;
      s1=sum(s);
      SpM(k)=sum(s1);
      S=cat(3,S,s);
   end;
end;
A=A/NTrial;
```

Далее необходимо выполнить следующую последовательность команд, сохраненную нами в файле `Glava14_4.m`:

```
Nspin=81; % число спинов системы
J=1; % константа обменного взаимодействия
h=0; % напряженность внешнего магнитного поля
```

```
Esi=-10; % конечная энергия системы
NTrial=50; % число испытаний
[Es,Ed,SpM,A,S]=Ising2(Nspin,J,h,Esi,NTrial);

% визуализация зависимости
% мгновенных значений полной энергии
% системы и энергии  демона от времени
i=1:2000;
figure(1);
plot(i,Es(i))
figure(2);
plot(i,Ed(i))

% визуализация мгновенных конфигураций
% системы в виде векторного поля
i=1:9;
j=1:9;
V(i,j)=0;
U(i,j)=0;
Z(i,j)=0;

figure(3);
quiver3(Z,U,V,S(:,:,1));
colormap white

figure(4)
quiver3(Z,U,V,S(:,:,100))
colormap white

figure(5)
quiver3(Z,U,V,S(:,:,1000))
colormap white

figure(6);
quiver3(Z,U,V,S(:,:,2000));
colormap white

% визуализация мгновенных
% конфигураций системы в виде
% прямоугольных параллелепипедов
figure(7);
bar3(S(:,:,1));
axis([0 10 0 10 -1.5 1.5]);
colormap white
```

```
figure(8);
bar3(S(:,:,100));
axis([0 10 0 10 -1.5 1.5]);
colormap white

figure(9);
bar3(S(:,:,1000));
axis([0 10 0 10 -1.5 1.5]);
colormap white

figure(10);
bar3(S(:,:,2000));
axis([0 10 0 10 -1.5 1.5]);
colormap white

mean(Es); % средняя энергия системы
Emean=mean(Es)/Nspin % средняя энергия
                     % системы на один спин

SpMmean=mean(SpM)/Nspin % средняя
                        % намагниченность
                        % на один спин

Amean=mean(A) % среднее значение
              % коэффициента принятия
              % на один шаг
Tmean=4/log(1+4/mean(Ed)) % равновесная
                          % температура
                          % системы
```

Ниже представлены результаты выполнения описанной последовательности команд.

1. Сообщения, выводимые в командное окно:

```
Emean =
   -0.1737
SpMmean =
    0.6109
Amean =
    0.2440
Tmean =
    2.5704
```

2. Графики, вычисленных зависимостей (рис. 14.11—14.20).

Рис. 14.11. Зависимость мгновенных значений энергии системы от времени

Рис. 14.12. Зависимость мгновенных значений энергии демона от времени

Рис. 14.13. Конфигурация системы в момент времени $t = 0$

Рис. 14.14. Конфигурация системы в момент времени $t = 100$

Задача 14.12

Проверьте, что распределение энергии демона описывается функцией распределения Больцмана. Сравните значение средней температуры системы, вычисленной по формуле (14.18), и значение температуры, полученное из распределения Больцмана.

Задача 14.13

Решите задачи 14.6—14.11 для рассмотренной выше двумерной системы спинов. Сравните полученные результаты с аналогичными результатами, полученными для одномерной системы спинов.

Рис. 14.15. Конфигурация системы в момент времени $t = 1000$

Рис. 14.16. Конфигурация системы в момент времени $t = 2000$

Рис. 14.17. Конфигурация системы в момент времени $t = 0$

Рис. 14.18. Конфигурация системы в момент времени $t = 100$

Рис. 14.19. Конфигурация системы в момент времени $t = 1000$

Рис. 14.20. Конфигурация системы в момент времени $t = 2000$

ЛИТЕРАТУРА К ГЛАВЕ 14

1. *Гулд Х., Тобочник Я.* Компьютерное моделирование в физике. Т. 2. М.: Мир, 1990.

2. *Кунин С.* Вычислительная физика. М.: Мир, 1992.

3. *Ландау Л. Д., Лифшиц Е. М.* Статистическая физика. М.: Наука, 2000.

4. Методы Монте-Карло в статистической физике / Под ред. К. Биндера. М.: Мир, 1982.

5. *Дайсон Ф., Монторолл Э., Кац М., Фишер М.* Устойчивость и фазовые переходы. М.: Мир, 1973.

6. *Хеерман Д. В.* Методы компьютерного эксперимента в теоретической физике. М.: Наука, 1990.

Глава 15

МОДЕЛИРОВАНИЕ КАНОНИЧЕСКОГО АНСАМБЛЯ МЕТОДОМ МОНТЕ-КАРЛО

15.1. ВВЕДЕНИЕ

В предыдущей главе мы использовали метод микроканонического ансамбля для моделирования поведения статистических систем. Одним из несомненных достоинств данного метода является очевидная простота исходного постулата о равенстве априорных вероятностей появления возможных микросостояний. Необходимо также отметить, что при проведении численных экспериментов с использованием микроканонического ансамбля демон не предъявляет никаких требований к качеству генерируемой последовательности случайных чисел. Данное обстоятельство определило известную популярность метода микроканонического распределения и его модификаций. Например, в [1, 2] описаны быстрые вычислительные алгоритмы, использующие один бит памяти на спин и большое число демонов.

В то же время алгоритму демона присущ целый ряд недостатков, пожалуй, главный из которых состоит в трудности получения системы с заранее заданной внутренней энергией. Необходимо также отметить, что с физической точки зрения более естественно изучать зависимости термодинамических величин, характеризующих макроскопическую систему, от температуры, но не от полной внутренней энергии. Так как в натурных экспериментах измеряется именно температура системы, но не ее внутренняя энергия.

В главе рассматривается метод канонического ансамбля, позволяющий моделировать поведение системы, находящейся при постоянной температуре. Данный метод используется для моделирования идеального классического газа и модели Изинга.

15.2. КАНОНИЧЕСКИЙ АНСАМБЛЬ

Подавляющее большинство физических систем не являются изолированными. Это означает, что они обмениваются энергией с окружающей средой. Так как размеры изучаемых физических систем, как правило, малы по сравнению с размерами окружающей среды, можно считать что любое изменение энергии малой системы не влияет заметным образом на температуру большой системы. (Например, включение в сеть утюга (малая система) не оказывает заметного влияния на температуру воздуха в комнате (большая система)). Следовательно, большую систему можно рассматривать как тепловой резервуар с заданной абсолютной температурой T (термостат). При приведении макроскопической системы в контакт с термостатом система стремится перейти

в равновесное состояние путем обмена энергией с термостатом. Процесс обмена энергией продолжается до тех пор, пока температура системы и температура термостата не станут равными.

В главе 14 мы показали, что в качестве малой системы может выступать система, состоящая из одной частицы (демон). Вероятность P_s того, что демон находится в микросостоянии s с энергией E_s, описывается формулой

$$P_s = \frac{1}{Z} e^{-\frac{E_s}{T}}, \tag{15.1}$$

где T — температура, измеряемая в энергетических единицах, Z — нормировочный множитель, выбирающийся из условия

$$\sum_{s=1}^{M} P_s = 1. \tag{15.2}$$

Подставив (15.1) в (15.2), найдем

$$Z = \sum_{s=1}^{M} e^{-\frac{E_s}{T}}. \tag{15.3}$$

Суммы в (15.2), (15.3) вычисляются по всем M микросостояниям системы.

Знание вероятности P_s позволяет вычислить средние по ансамблю макроскопические характеристики $\langle A \rangle$ в соответствии с общим правилом

$$\langle A \rangle = \sum_{s=1}^{M} A_s P_s. \tag{15.4}$$

Например, средняя по ансамблю микросостояний энергия $\langle E \rangle$ равна

$$\langle E \rangle = \sum_s E_s P_s = \frac{1}{Z} \sum_s E_s e^{-\frac{E_s}{T}} \tag{15.5}$$

Отметим, что из-за большого числа слагаемых прямое вычисление сумм в (15.5) невозможно. Действительно, даже для достаточно малой решетки размером 16×16 имеется $2^{256} \approx 10^{77}$ конфигураций. В этих условиях для оценки средних значений макроскопических характеристик системы приходится использовать вероятностные методы, одним из вариантов которых является алгоритм Метрополиса, рассматриваемый нами в следующем разделе.

15.3. АЛГОРИТМ МЕТРОПОЛИСА ДЛЯ КАНОНИЧЕСКОГО АНСАМБЛЯ

Рассмотрим систему, состоящую из N частиц, находящихся в объеме V при постоянной температуре T. Сгенерируем некоторое ограниченное число конфигураций m

из полного числа возможных конфигураций M. Тогда оценка среднего значения $\langle A\rangle$ может быть получена из следующего выражения

$$\langle A\rangle \approx \sum_{s=1}^{m} A_s P_s = \frac{\sum\limits_{s=1}^{m} A_s e^{-\frac{E_s}{T}}}{\sum\limits_{s=1}^{m} e^{-\frac{E_s}{T}}}, \tag{15.6}$$

где E_s, A_s — полная энергия системы E и значение физической величины A в конфигурации s.

Для вычисления $\langle A\rangle$ в соответствии с (15.6) можно предложить достаточно очевидную процедуру, которая состоит в генерации случайной конфигурации, вычислении E_s, A_s и произведения

$$A_s e^{-\frac{E_s}{T}}$$

и подсчете соответствующего вклада каждой конфигурации в сумму (15.6). Однако появление каждой конкретной конфигурации достаточно маловероятно, следовательно, большое число конфигураций будут давать малый вклад в суммы (15.6), поэтому, вообще говоря, такой вычислительный алгоритм также будет недостаточно эффективным.

Для устранения отмеченного недостатка можно воспользоваться методом существенной выборки, который применялся нами ранее в главе 11 для вычисления определенных интегралов методом Монте-Карло. Идея данного метода состоит в генерации конфигурации в соответствии с функцией распределения вероятностей π_s. Так как дальнейшее усреднение проводится по m конфигурациям смещенной выборки, то для исключения данного смещения при вычислении суммы в (15.6) каждая конфигурация должна браться с весовым множителем $1/\pi_s$:

$$\langle A\rangle \approx \frac{\sum\limits_{s=1}^{m} A_s \frac{1}{\pi_s} e^{-\frac{E_s}{T}}}{\sum\limits_{s=1}^{m} \frac{1}{\pi_s} e^{-\frac{E_s}{T}}}. \tag{15.7}$$

Наиболее целесообразно оказывается выбрать $1/\pi_s$ в виде [3]:

$$\pi_s = \frac{e^{-\frac{E_s}{T}}}{\sum\limits_{s=1}^{m} e^{-\frac{E_s}{T}}}. \tag{15.8}$$

В этом случае, как легко убедиться прямой подстановкой (15.8) в (15.7), $\langle A\rangle$ равно

$$\langle A\rangle \approx \frac{\sum\limits_{s=1}^{m} A_s \sum\limits_{s=1}^{m} e^{-\frac{E_s}{T}}}{\sum\limits_{s=1}^{m}\sum\limits_{s=1}^{m} e^{-\frac{E_s}{T}}} = \frac{\sum\limits_{m=1}^{m} A_s}{m}, \tag{15.9}$$

где при преобразовании знаменателя учтено условие нормировки

$$\sum_{s=1}^{m} e^{-\frac{E_s}{T}} = 1.$$

Формула (15.9) позволяет дать общую формулировку алгоритма Метрополиса:

1. Сформировать начальную конфигурацию системы.
2. Произвести случайное пробное изменение начальной конфигурации.
3. Вычислить изменение энергии системы ΔE, обусловленное произведенным пробным изменением конфигурации.
4. Если $\Delta E \leq 0$, то принять новую конфигурацию системы и осуществить переход к п. 8.
5. Если $\Delta E \geq 0$, то вычислить «вероятность перехода»
$$W = e^{-\frac{\Delta E}{T}}.$$
6. Сгенерировать случайное число r с равномерным законом распределения в интервале $[0, 1]$.
7. Если $r \leq W$, принять новую конфигурацию, в противном случае сохранить предыдущую конфигурацию.
8. Определить значения требуемых физических величин.
9. Повторить п. 2—8 для получения достаточного числа конфигураций.
10. Вычислить средние по статистически независимым друг от друга конфигурациям.

Описанный алгоритм можно трактовать как случайные блуждания. Действительно, пронумеруем различные конфигурации порядковыми номерами $i = 1, 2, 3, \ldots$ Тогда каждую конфигурацию можно считать некоторой «точкой», а вычислительный процесс в соответствии с алгоритмом Метрополиса случайным блужданием по данным точкам. Действия, выполняемые в соответствии с п. 3—7, позволяют определить условную вероятность того, что в «момент времени» прохожий будет находиться в точке i при условии, что в момент времени t он находился в точке j, то есть найти отношение $P(i)/P(j)$. Отметим, что поскольку вычисляется отношение вероятностей, нет необходимости проводить их нормировку. Так как конфигурации генерируются с вероятностью, пропорциональной требуемой вероятности, для вычисления средних следует использовать формулу (15.9). Отметим, что данным способом оказывается возможным оценить средние значения макроскопических величин, но не нормировочный коэффициент Z.

Существует строгое математическое доказательство того факта, что после достаточно большого числа шагов алгоритм Метрополиса генерирует состояния, энергия которых распределена по Больцману. Однако, следуя выбранному подходу (физическому по своей сути), мы сознательно отказываемся от математически строгого доказательства в пользу эмпирического подхода, то есть подтверждения данного вывода на примере модели идеального газа и одномерной системы спинов.

Отметим, что выбор в качестве распределения Больцмана не является единственно возможным. Оказывается, что к распределению Больцмана в асимптотическом

пределе приводят и другие распределения вероятности. Можно показать, что существует единственное необходимое условие, заключающееся в том, чтобы

$$W(1 \to 2)\, e^{-\frac{E_1}{T}} = W(2 \to 1)\, e^{-\frac{E_2}{T}}. \tag{15.10}$$

Равенство (15.10) называется принципом детального равновесия.

На примере классического идеального газа покажем, что алгоритм Метрополиса приводит для отдельных микросостояний к распределению Больцмана. Любое микросостояние газа полностью описывается заданием скорости частиц, так как энергия идеального газа зависит только от скорости частиц. При этом скорость классической частицы является величиной непрерывной, в то время как для количественного описания системы необходимо иметь счетное число микросостояний. Следовательно, необходимо провести разбиение интервала возможных значений скоростей на конечное число достаточно малых дискретных интервалов. Например, для 20 частиц и разделении интервала возможных скоростей на 10 интервалов полное число возможных микросостояний составляет 10^{20}. Совершенно очевидно, что весьма проблематично не только пронумеровать данные состояния, но и вычислить за приемлемое время точные оценки вероятностей и средних величин. Полученная оценка подтверждает правомерность использования модели идеального газа для демонстрации асимптотических характеристик алгоритма Метрополиса.

Для реализации алгоритма Метрополиса в модели идеального газа, состоящего из одной частицы, в MATLAB создадим файл `IdealGas.m`, содержащий описание функции, возвращающей мгновенные значения скорости, энергии, среднее число принятия.

```
% листинг файла IdealGas.m
function [V,Es,Accept] = IdealGas(Vel,T,dVmax,NTrial)
% функция, возвращающая
% мгновенные значения скорости (V),
% энергии (E),
% среднее число принятия (Accept)
% Vel - начальная скорость
% T - температура газа
% dVmax - максимальное изменение скорости
% NTrial - число испытаний

beta=1/T;
E=Vel.^2/2; % начальная
            % энергия
Accept=0;

for i=1:NTrial
   dV=(2*rand(1)-1)*dVmax;
   Vtrial=Vel+dV; % случайное изменение
                  % скорости
   de=0.5*(Vtrial.^2-Vel.^2); % пробное
                              % изменение энергии
```

```
   if de>0
     if exp(-beta*de)>=rand(1)
        % шаг принимается
       Vel=Vtrial;
       Accept=Accept+1;
       E=E+de;
     end;
   else
      % шаг не принимается
      Vel=Vtrial;
      Accept=Accept+1;
      E=E+de;
   end;
   Es(i)=E;
   V(i)=Vel;
end;
Accept=Accept/NTrial;
```

Далее необходимо выполнить следующую последовательность команд, сохраненную нами в файле `Glava15_1.m`:

```
% листинг файла Glava15_1.m
Vel=0; % начальная скорость частицы
T=0.5; % температура газа
dVmax=4; % максимальное изменение скорости
NTrial=10^4; % число испытаний

% вычисление мгновенных
% значений скорости (V),
% энергии (E),
% среднего числа принятия (Accept)
[V Es Accept]=IdealGas(Vel,T,dVmax,NTrial);

Vmean=mean(V) % средняя скорость частицы
Esmean=mean(Es) % средняя энергия частицы
Accept % число принятия

i=1:NTrial;
% визуализация зависимости
% мгновенных значений энергии
% от времени
figure;
plot(i,Es,'k');

% вычисление и визуализация
% функции распределения  по
```

```
% энергии идеального газа,
% состоящего из одной частицы

x1=min(Es);
x2=max(Es);

Nint=100; % число интервалов гистограммы
i=1:Nint;
x(i)=x1+(x2-x1)/Nint*i;

h=hist(Es,x);
figure;
bar(x,h);
colormap white
```

Ниже представлены результаты выполнения описанной последовательности команд.

1. Текстовые сообщения, выводимые в командное окно:

```
    Vmean =
  -7.6218e-004
Esmean =
    0.2425
Accept =
    0.2883
```

2. Графики вычисленных зависимостей (рис. 15.1—15.2).

Рис. 15.1. Зависимость мгновенных значений энергии идеального газа, состоящего из одной частицы, от времени

Рис. 15.2. Распределение случайной последовательности, представленной на рис. 15.1

***Задача* 15.1**

1. *Постройте зависимость* $\ln(P(E))$ *от* E. *(Указание: создайте функцию, формирующую из массивов* `x`, `h` *матрицу, не содержащую значений* $h_i = 0$, *и используйте ее для дальнейшего решения задачи.)*

2. *Аппроксимируйте данную зависимость функцией вида* $ax + b$. *Сравните значение коэффициента* a *с точным значением* $-1/T$. *Постройте график функции* e^{ax+b} *и сравните его с графиком распределения плотности вероятности. С какой точностью распределение газа, полученное в результате моделирования, аппроксимируется больцмановским распределением?*

***Задача* 15.2**

Проведите вычисления для различных значений температуры и постройте зависимость энергии газа от температуры. Можно ли аппроксимировать данную зависимость функцией вида $ax + b$? *Чему равен коэффициент* a *в данном случае? Какой физический смысл имеет коэффициент* a? *Объясните полученный результат с точки зрения теоремы о равнораспределении (см. раздел 10.5).*

***Задача* 15.3**

Проведите вычисления для различных значений температуры и постройте зависимости дисперсии и среднеквадратичного отклонения скорости от температуры. Выберите подходящие функции и аппроксимируйте данные зависимости методом наименьших квадратов. Объясните полученные результаты.

Для моделирования одномерного классического идеального газа, состоящего из N частиц, следует использовать модифицированную функцию `IdealGas2`, сохраненную в файле `IdealGas2.m`. Функция `IdealGas2` отличается от функции `IdealGas`, описанной выше, тем, что:

1) полная энергия газа вычисляется как сумма кинетических энергий всех частиц;

2) помимо случайного выбора величины изменения скорости производится случайный выбор номера той частицы, для которой производится пробное изменение скорости.

Все остальные действия проводятся в соответствии с алгоритмом Метрополиса.

```
% листинг файла IdealGas2.m
function [V,Es,Accept] = IdealGas2(Vel,T,dVmax,NTrial,Np)
% функция, возвращающая
% мгновенные значения средней скорости (V),
% энергии (E),
% числа принятия (Accept)
% Vel - начальная скорость
```

```
% T - температура газа
% dVmax - максимальное изменение скорости
% NTrial - число испытаний
% Np - число частиц

beta=1/T;
i=1:Np;
v(i)=Vel;
Eold=dot(v,v)/2;
V(1)=mean(v);
Es=Eold;
Accept=0;
k=2;
for i=1:NTrial
  for j=1:Np
     N=floor(Np*rand(1)+1); % случайный
                            % выбор частицы
    dV=(2*rand(1)-1)*dVmax; % случайное изменение
                            % скорости
     Vtemp=v(j);
     v(j)=v(j)+dV;
     Enew=0.5*dot(v,v);
     de=Enew-Eold; % пробное
                   % изменение энергии
    if de>0
       % шаг принимается
       if exp(-beta*de)>rand(1)
         Accept=Accept+1;
         Eold=Enew;
       else
         % шаг не принимается
         v(j)=Vtemp;
       end;
    else
       Accept=Accept+1;
       Eold=Enew;
    end;
    V(k)=mean(v);
    Es(k)=Eold/Np;
    k=k+1;
  end;
end;
Accept=Accept/(NTrial*Np);
```

Далее необходимо выполнить следующую последовательность команд, сохраненную нами в файле `Glava15_2.m`:

```
% листинг файла Glava15_2.m
clear all
Vel=0; % начальная скорость частицы
T=5; % температура газа
dVmax=0.5; % максимальное изменение скорости
NTrial=2000; % число испытаний
Np=80; % число частиц

% вычисление мгновенных
% значений скорости (V),
% энергии (E),
% среднего числа принятия (Accept)
[V Es Accept]=IdealGas2(Vel,T,dVmax,NTrial, Np);

Vmean=mean(V) % средняя скорость частицы
Esmean=mean(Es) % средняя энергия частицы
Accept % число принятия

i=1:size(Es,2);
% визуализация зависимости
% мгновенных значений энергии от времени
figure;
plot(i,Es,'k');

% вычисление и визуализация
% функции распределения  по
% энергии идеального газа,
% состоящего из одной частицы

x1=min(Es);
x2=max(Es);

Nint=100; % число интервалов гистограммы

i=1:Nint;
x(i)=x1+(x2-x1)/Nint*i;

h=hist(Es,x);
figure;
bar(x,h);
colormap white
```

Ниже представлены результаты выполнения описанной последовательности команд.

1. Текстовые сообщения, выводимые в командное окно:

```
Vmean =
   -0.0520
Esmean =
    2.5185
Accept =
    0.9552
```

2. Графики вычисленных зависимостей (рис. 15.3–15.4).

Рис. 15.3. Зависимость мгновенной энергии классического идеального газа от времени

Рис. 15.4. Гистограмма распределения мгновенных значений энергии классического идеального газа

Рассмотрим полученные результаты для случая $T = 5$, $Vel = 0$, $\Delta v_{\max} = 0$, $Ntrial = 2000$, $Np = 80$, опустив промежуточные вычисления, которые аналогичны вычислениям, описанным выше, для классического идеального газа, состоящего из одной частицы.

Анализ зависимости, представленной на рис. 15.3, показывает, что существует некоторый переходный период конечной длительности, в течение которого система приходит в «тепловое равновесие» с термостатом. После достижения равновесия температура системы испытывает флуктуации относительно среднего значения, соответствующего заданной температуре.

***Задача* 15.4**

Проведите численные эксперименты и найдите зависимость числа принятия на один шаг $mean\left(A^{\langle 2\rangle}\right)/Np$ *от* $\Delta v_{\max}$. *Для значения* $\Delta v_{\max}$, *обеспечивающего значение коэффициента принятия, близкого к 0,5, вычислите среднюю кинетическую энергию одной частицы и среднюю скорость частиц. Как связаны значение температуры и средняя кинетическая энергия одной частицы? Сравните полученный результат с результатами задачи 15.2.*

***Задача* 15.5**

Выберите критерий достижения системой теплового равновесия. Оцените число шагов Монте-Карло на частицу, необходимых для достижения системой теплового равновесия.

Исследуйте зависимость времени достижения равновесия от начальных скоростей системы, $\Delta v_{\max}$, числа частиц газа. Какими надо выбирать начальные скорости при заданном числе частиц и для того, чтобы система достигала равновесия за наименьшее время?

***Задача* 15.6**

Исключите временной интервал, соответствующий неравновесному состояню системы. Для данного случая вычислите средние значения скорости, энергии системы и числа принятия. Сравните вычисленные значения с аналогичными значениями, полученными по последовательности с неисключенным процессом релаксации. Как зависит разность между этими значениями от числа частиц N и числа шагов Монте-Карло (NTrial)?

***Задача* 15.7**

Вычислите среднюю энергию на частицу для $T = 0{,}5,\ 1,\ 5,\ 10,\ 20,\ 30,\ 40,\ 50,\ 70,\ 90,\ 100$, исключив временной интервал, соответствующий неравновесному состоянию газа. Оцените теплоемкость газа c_v

$$c_v = \left(\frac{\partial E}{\partial T}\right)_{V\,=\,\mathrm{const}}.$$

Сравните полученный результат с результатом, полученным в задаче 15.2.

***Задача* 15.8**

Вычислите средний квадрат флуктуаций энергии

$$\left\langle \Delta E^2 \right\rangle = \left\langle E^2 \right\rangle - \left\langle E \right\rangle^2$$

для $T = 10, 50, 100$.

Сравните величину отношения $\left\langle \Delta E^2 \right\rangle / T^2$ с теплоемкостью, вычисленной в задаче 15.8.

***Задача* 15.9**

Используя функцию `IsingM`*, сохраненную в файле* `IsingM.m`*, листинг которого приведен ниже, промоделируйте одномерную систему спинов в соответствии с алгоритмом Метрополиса.*

```
% листинг файла IsingM.m
function [Es,SpM,Accept,S]=Ising(Nspin,J,h,NTrial,T)
% функция, возвращающая значения:
% энергии системы (Es),
% магнитного момента системы (SpM),
% принятия решений (Accept),
% мгновенных конфигураций спинов (S)
% Nspin - число спинов
```

```
% J - константа обменного взаимодействия
% h - напряженность внешнего магнитного поля
% NTrial - число испытаний
% T - температура системы

beta=1/T;
s=ones(Nspin);
S(1)=s;
M=Nspin;
Esystem=-(J+h)*Nspin;
Edemon=2*J*ceil((Esi-Esystem)/(2*J));
Es(1)=Esystem;
SpM(1)=M;
Accept=0;
k=1;

for i=1:NTrial
  for j=1:Nspin
    Ispin=floor(Nspin*rand(1)+1); % выбор
                                  % случайного
                                  % спина
      Left=s(Nspin);
    else
      Left=s(Ispin-1);
    end;
    if Ispin==Nspin
      Right=s(1);
    else
      Right=s(Ispin+1);
    end;
    de=2*s(Ispin)*(-h+J*(Left+Right)); % пробное
                                       % изменение
                                       % энергии
    if (de<=0) | (exp(-beta*de)>rand(1))
      s(Ispin)=-s(Ispin);
      Accept=Accept+1;
      Esystem=Esystem+de;
    end;
    k=k+1;
    S=cat(3,S,s);
    Es(k)=Esystem/Nspin;
    SpM(k)=sum(s)/Nspin;
  end;
end;
Accept=Accept/(NTrial*Nspin);
```

1. *Проведите вычисления для следующих значений переменных* $Nspin = 100$, $J = 1$, $h = 0$, $NTrial = 1000$, $T = 1, 2, 5, 10, 20, 30, 40, 50$. *Постройте график зависимости полной энергии системы, приходящейся на один спин от температуры.*

   Аппроксимируйте методом наименьших данную зависимость функцией вида $ax + b$. *Чему равен угловой коэффициент прямой?*

2. *Используя найденный коэффициент пропорциональности между энергией системы и температурой, постройте зависимость температуры системы от времени. Чему равна начальная «температура системы»? Оцените время, необходимое системе для достижения равновесия.*

3. *Измените функцию* `IsingM.m` *так, чтобы все спины системы были ориентированы в момент времени* $t = 0$ *хаотично. Чему равна начальная температура системы в данном случае? Оцените время, необходимое системе для достижения равновесия.*

4. *Определите зависимость средней энергии, приходящейся на один спин,* $\langle E \rangle$ *и средней намагниченности на один спин* $\langle M \rangle$ *от температуры в интервалах температур* $[0, 1{,}5]$ *и* $[5, 50]$. *Представьте данные зависимости графически и проанализируйте их качественные особенности. Зависят ли полученные результаты от начальной конфигурации системы?*

5. *Сравните результаты вычислений с точным ответом для нулевого магнитного поля*
$$\langle E \rangle = -Nspin \cdot th\left(\frac{J}{T}\right). \tag{15.11}$$

6. *Постройте зависимость коэффициента принятия решений в интервалах температур* $[0{,}1, 5]$ *и* $[5, 50]$. *Как зависит эффективность алгоритма Метрополиса от температуры?*

7. *Вычислите плотности вероятностей для различного числа спинов Nspin и различных температур T.*

***Задача* 15.10**

Решите задачи 15.1—15.10, используя вариант вероятности перехода, следующий из принципа детального равновесия (15.10):

$$W'(1 \to 2) = \frac{e^{-\frac{E_2}{T}}}{e^{-\frac{E_1}{T}} + e^{-\frac{E_2}{T}}}. \tag{15.12}$$

Для реализации симметричной формы вероятности перехода следует вычислять величину

$$W'(1 \to 2) = \frac{e^{-\frac{\Delta E}{T}}}{1 + e^{-\frac{\Delta E}{T}}}. \tag{15.13}$$

где $\Delta E_2 - E_1$.

15.4. МОДЕЛИРОВАНИЕ ДВУМЕРНОЙ МОДЕЛИ ИЗИНГА МЕТОДОМ КАНОНИЧЕСКОГО АНСАМБЛЯ

Целый ряд чистых материалов (например, железо, никель) и различных сплавов в отсутствие внешнего магнитного поля имеют спонтанную намагниченность. Данные свойства проявляются только при температуре ниже определенной температуры T_C, называемой критической температурой или температурой Кюри. При температурах $T > T_C$ спонтанная намагниченность исчезает, то есть происходит переход из упорядоченного режима, в котором спины расположены упорядоченно, в хаотический режим, в котором ориентация спинов является случайной.

В соответствии с современными представлениями механизм ферромагнетизма имеет квантово-механическую природу, поэтому для построения адекватной модели изучаемого явления необходимо использовать математический аппарат квантовой механики. Однако оказывается, что даже изучение классических двумерных и трехмерных моделей Изинга дает возможность выявить общие закономерности поведения магнитных систем вблизи фазового перехода. В то же время важно понимать, что классические модели Изинга, которым присущи известные ограничения, не могут дать полного адекватного описания ферромагнетизма, что особенно заметно проявляется при температурах, близких к абсолютному нулю.

Приступая к исследованию модели Изинга, необходимо определить макроскопические характеристики системы, представляющие интерес с физической точки зрения. К таковым характеристикам обычно относят среднюю энергию системы $\langle E \rangle$, среднюю намагниченность $\langle M \rangle$, удельную теплоемкость системы при постоянном объеме c_v и магнитную восприимчивость. Отметим, что наиболее часто исследователей интересуют равновесные значения указанных величин. Способ определения одной из макроскопических характеристик системы — удельной теплоемкости c_v, равной по определению

$$c_v = \frac{\partial \langle E \rangle}{\partial T}, \tag{15.14}$$

подробно обсуждался в предыдущем разделе.

Альтернативный метод измерения теплоемкости c_v основан на использовании связи теплоемкости флуктуациями полной энергии в каноническом ансамбле. Для нахождения соотношения, связывающего удельную теплоемкость c_v с флуктуациями энергии системы в каноническом ансамбле, подставим в (15.14) выражение для средней энергии ансамбля (15.5):

$$c_v = \frac{\partial}{\partial T} \frac{1}{Z} \sum_{s=1}^{M} E_s e^{-\frac{E_s}{T}}, \tag{15.15}$$

где

$$Z = \sum_{s=1}^{M} e^{-\frac{E_s}{T}}.$$

Выполнив дифференцирование в (15.15), получим

$$c_v = -\frac{1}{Z^2}\frac{\partial Z}{\partial T}\sum_{s=1}^{M} E_s e^{-\frac{E_s}{T}} - \frac{1}{Z}\cdot\frac{1}{T^2}\sum_{s=1}^{M} E_s^2 e^{-\frac{E_s}{T}}. \tag{15.16}$$

Учитывая, что

$$\frac{\partial Z}{\partial T} = \frac{1}{T^2}\sum_{s=1}^{M} E_s e^{-\frac{E_s}{T}},$$

из (15.16) окончательно найдем

$$c_v = \frac{1}{T^2}\left(\langle E\rangle^2 - \langle E^2\rangle\right). \tag{15.17}$$

Изотермическая магнитная восприимчивость χ, характеризующая чувствительность системы к воздействию внешнего поля при постоянной температуре, по определению равна

$$\chi = \lim_{h\to 0}\frac{\partial\langle \mathrm{M}\rangle}{\partial h}. \tag{15.18}$$

Данную величину, точно так же, как удельную теплоемкость c_v, можно связать с флуктуациями намагниченности системы. Запишем энергию системы в виде

$$E_s = E_{0,s} - h\mathrm{M}_s, \tag{15.19}$$

где $E_{0,s}$ — энергия s-го состояния в отсутствие магнитного поля, H — наложенное внешнее магнитное поле, M_s — намагниченность в s-м состоянии.

Средняя намагниченность $\langle \mathrm{M}\rangle$ по определению равна

$$\langle \mathrm{M}\rangle = \frac{1}{Z}\sum_{s=1}^{M} \mathrm{M}_s e^{-\frac{E_s}{T}}. \tag{15.20}$$

Продифференцировав (15.19), по h получаем

$$\frac{\partial E_s}{\partial h} = -\mathrm{M}_s. \tag{15.21}$$

Используя (15.21), имеем

$$\frac{\partial Z}{\partial h} = \sum_{s=1}^{M}\frac{1}{T}\mathrm{M}_s e^{-\frac{E_s}{T}}. \tag{15.22}$$

Следовательно,

$$\frac{\partial\langle \mathrm{M}\rangle}{\partial h} = -\frac{1}{Z^2}\frac{\partial Z}{\partial h}\sum_{s=1}^{M}\mathrm{M}_s e^{-\frac{E_s}{T}} - \frac{1}{Z}\frac{1}{T}\sum_{s=1}^{M}\frac{\partial E_s}{\partial h}\mathrm{M}_s e^{-\frac{E_s}{T}}. \tag{15.23}$$

Подставив в (15.23) выражения (15.21), (15.22), получим

$$\frac{\partial\langle \mathrm{M}\rangle}{\partial h} = \frac{1}{T}\left(\left\langle \mathrm{M}^2\right\rangle - \langle \mathrm{M}\rangle^2\right). \tag{15.24}$$

Подставив (15.24) в (15.18), окончательно найдем

$$\chi = \frac{1}{T}\left(\left\langle \mathrm{M}^2 \right\rangle - \langle \mathrm{M} \rangle^2\right). \tag{15.25}$$

где M, $\left\langle \mathrm{M}^2 \right\rangle$ отвечают нулевому внешнему полю.

Отметим, что соотношения (15.17), (15.25) являются общими соотношениями между функциями отклика и равновесными флуктуациями.

Следующий шаг после выбора интересующих равновесных характеристик системы и задания способа их определения состоит в реализации алгоритма Метрополиса, который в общем виде приведен в разделе 15.2. Применяя алгоритм Метрополиса, можно сгенерировать спиновые конфигурации $\boldsymbol{S}$ с вероятностью $w(\boldsymbol{S})$, а затем усреднить искомые величины по всем конфигурациям. Для генерации спиновых конфигураций можно, например, выбрать такой пробный шаг от $\boldsymbol{S}$ к $\boldsymbol{S}_t$, при котором все спины одновременно меняются случайным образом. Однако такой пробный шаг приводит к получению конфигурации, значительно отличающейся от исходной конфигурации $\boldsymbol{S}$, и следовательно, вероятность отбросить данную конфигурацию была бы велика, поэтому оказывается более предпочтительным использовать более мелкие шаги. Для этого будем рассматривать конфигурации, отличающиеся переворотом одного спина. При этом переворачиваемый спин s_{ij} выбирается случайным образом. Принятие данного пробного шага зависит от отношения весовых функций

$$r = \frac{w(\mathbf{S}_t)}{w(\mathbf{S})} = e^{-\frac{E_{S_t}}{T}} e^{\frac{E_S}{T}}. \tag{15.26}$$

Если $r > 1$ или если $r < 1$, но больше, чем равномерно распределенное на отрезке $[0, 1]$ случайное число, то спин s_0 переворачивается и принимается новая конфигурация, в противном случае спин остается неизменным. На одном шаге Монте-Карло предпринимается N попыток переворота спина ($N = Nspin$).

Для реализации описанного алгоритма в MATLAB создадим три файла:

1) файл `Espin.m`, содержащий описание функции, возвращающей энергию заданной конфигурации спинов;

2) файл `InitSp.m`, содержащий описание функции, возвращающей энергию системы, значение полного магнитного момента, матрицу, содержащую информацию о конфигурации системы, в момент времени $t = 0$;

3) файл `IsingKA.m`, содержащий описание функции, возвращающей мгновенные значения энергии системы, число принятия, мгновенный магнитный момент системы, мгновенные конфигурации спинов системы.

```
% листинг файла Espin.m
function z = Espin(Nspin,s,h,J)
% функция, возвращающая энергию
% заданной конфигурации спинов
% Nspin - число спинов системы
% s - матрица, содержащая информацию
```

```
% о положении спинов
% h - напряженность внешнего магнитного поля
% J - константа обменного взаимодействия

Ns=Nspin.^0.5;
E=0;
for i=1:Ns
   for j=1:Ns
     % проверка периодических граничных условий
     if j==1
        Left=Ns;
     else
        Left==j-1;
     end;
     if j==Ns
        Right=1;
     else
        Right=j+1;
     end;
     if i==Ns
       Up=1;
     else
       Up=i+1;
     end;
     if i==1
        Down=Ns;
     else
        Down=i-1;
     end;
     % энергия системы
     E=E-s(i,j)*(h+J*(s(i,Left)+s(i,Right)+s(Down,j)+s(Up,j)));
   end;
end;
z=E;

% листинг файла InitSp.m
function [E,s,M] = InitSp(Nspin,J,h)
% функция, возвращающая:
% энергию системы (E),
% значение полного магнитного момента (S),
% матрицу, содержащую информацию о
% конфигурации системы, в момент времени t = 0
% Nspin - число спинов системы
% J - константа обменного взаимодействия
% h - напряженность внешнего магнитного поля
```

```
Ns=Nspin.^0.5;
M=0;
for i=1:Ns
   for j=1:Ns
      s(i,j)=1;
      M=M+s(i,j);
   end;
end;
E=Espin(Nspin,s,h,J);

% листинг файла IsingKA.m
function [E,Accept,M,S] = IsingKA(Nspin,J,h,...
                                  NTrial,T,...
                                  Sstart,Mstart,Estart)
% функция, возвращающая
% мгновенные значения энергии системы (E),
% число принятия (Accept),
% мгновенный магнитный момент системы (M),
% мгновенные конфигурации спинов системы (S)
% Nspin - число спинов системы
% J - константа обменного взаимодействия
% h - напряженность внешнего магнитного поля
% NTrial - число испытаний
% T - температура системы
% Sstart - матрица, содержащая информацию
% об ориентации спинов в
% момент времени t = 0
% Mstart - значение магнитного момента
% системы в момент времени t = 0
% Estart - энергия системы в момент времени t = 0

Ns=Nspin.^0.5;
M(1)=Mstart;
S=Sstart;
E(1)=Estart;
s=Sstart;
Accept=0;
k=2;

for i=1:NTrial
   for j=1:Nspin
      % случайный выбор номера узла
      % для опрокидывания спина
      Ix=floor(Ns*rand(1)+1);
      Iy=floor(Ns*rand(1)+1);
      % проверка периодических
```

```
      % граничных условий
      if Ix==1
         Left=Ns;
      else
         Left=Ix-1;
      end;
      if Ix==Ns
         Right=1;
      else
         Right=Ix+1;
      end;
      if Iy==Ns
         Up=1;
      else
         Up=Iy+1;
      end;
      if Iy==1
         Down=Ns;
      else
         Down=Iy-1;
      end;
      % пробный переворот спина
      Temp=s(Iy,Ix);
      s(Iy,Ix)=-s(Iy,Ix);
      de=2*s(Iy,Ix)*(h+...
                     J*(s(Iy,Left)+...
                     s(Iy,Right)+...
                     s(Down,Ix)+...
                     s(Up,Ix)));
      if or(de<=0,rand(1)<=exp(-de./T))
         % пробный переворот спина принимается
         Accept=Accept+1;
         E(k)=E(k-1)-de;
         M(k)=M(k-1)+2*s(Iy,Ix);
      else
         % пробный переворот спина
         % не принимается
         s(Iy,Ix)=Temp;
         E(k)=E(k-1);
         M(k)=M(k-1);
      end;
      S=cat(3,S,s);
      k=k+1;
   end;
end;
```

```
Accept=Accept/(NTrial*Nspin); % среднее число
                              % принятия
```

Далее необходимо выполнить следующую последовательность команд, сохраненную нами в файле Glava15_3.m:

```
% листинг файла Glava15_3.m
Nspin=64; % число спинов системы
J=1; % константа обменного взаимодействия
h=0; % напряженность внешнего магнитного поля
T=1.5; % температура системы
NTrial=100; % число испытаний

% вычисление параметров
% начальной конфигурации
[Estart,Sstart,Mstart]=InitSp(Nspin,J,h);

% вычисление параметров текущих
% конфигураций в соответствии
% с алгоритмом Метрополиса
[E,Accept,M,S] = IsingKA(Nspin,J,h,...
                         NTrial,T,...
                         Sstart,Mstart,Estart);

% визуализация зависимости
% мгновенных значений энергии и магнитного
% момента системы от времени
i=1:length(E);
figure(1);
plot(i,E,'k');
axis([0 length(E) -300 10])

figure(2);
plot(i,M,'k')

% визуализация мгновенных конфигураций системы
figure(3);
bar3(S(:,:,1));
colormap white;
axis([0 9 0 9 -1 1])

figure(4);
bar3(S(:,:,10));
colormap white;
axis([0 9 0 9 -1 1])
```

```
figure(5);
bar3(S(:,:,100));
colormap white;
axis([0 9 0 9 -1 1])

figure(6);
bar3(S(:,:,500));
colormap white;
axis([0 9 0 9 -1 1])

figure(7);
bar3(S(:,:,2000));
colormap white;
axis([0 9 0 9 -1 1])
% выделение мгновенных значений
% энергии и магнитного момента,
% соответствующих равновесному
% состоянию системы
j=1:length(E)-2000;
E1(j)=E(j+2000);
M1(j)=M(j+2000);

% вычисление теплоемкости
% в соответствии с (15.17)
E2=E1.^2;
c=(1/T.^2)*(mean(E2)-mean(E1).^2)/Nspin
M2=M1.^2;
% вычисление магнитной восприимчивости
% в соответствии с (15.25)
xi=(1/T)*(mean(M2)-mean(M1).^2)
% вычисление намагниченности системы
% в соответствии с (5.25)
M1mean=mean(M1)
```

Ниже представлены результаты выполнения описанной последовательности команд.

1. Сообщения, выводимые в командное окно:

```
      c =
      0.1574
   xi =
      0.8792
   M1mean =
     -0.1881
```

2. Графики вычисленных зависимостей (рис. 15.5—15.11).

Рис. 15.5. Зависимость мгновенных значений энергии системы спинов от времени

Рис. 15.6. Зависимость мгновенных значений магнитного момента системы спинов от времени

Рис. 15.7. Конфигурация системы в момент времени $t = 0$

Рис. 15.8. Конфигурация системы в момент времени $t = 10$

Рис. 15.9. Конфигурация системы в момент времени $t = 100$

Рис. 15.10. Конфигурация системы в момент времени $t = 500$

Рис. 15.11. Конфигурация системы в момент времени $t = 2000$

***Задача* 15.11**

Из выражения для полной энергии системы

$$E = -J \sum_{\langle i,j \rangle}^{N} s_i s_j - h \sum_{i=1}^{N} s_i$$

видно, что вклад в значение переменной r, вычисляемой в соответствии с (15.26), дают только слагаемые, содержащие s_{ij}:

$$r = e^{-2\Delta s_{ij}(JS+h)}, \tag{15.27}$$

где Δs_{ij} — изменение s_{ij} спина, обусловленное его переворотом, $\mathbf{S}$ — сумма четырех спинов, соседних с переворачиваемым спином. Так как $\mathbf{S}$ может принимать только 5 различных значений 0, при $h = 0$ имеется всего 5 различных значений энергии (см. рис. 15.12): $0, \pm 2J, \pm 4J, \pm 8J$. Следовательно, чтобы не вычислять каждый раз значение экспоненты, для уменьшения времени счета можно один раз вычислить и запомнить таблицу значений r. Внесите в файл `Ising2.m` *необходимые изменения и попробуйте оценить, на сколько сократилось время вычислений.*

***Задача* 15.12**

Исследуйте зависимость времени релаксации системы к равновесному состоянию τ_{rel} от температуры системы T_s и числа спинов системы $Nspin$. Отобразите зависимости $\tau_{rel} = \tau_{rel}(T_s)$, $\tau_{rel} = \tau_{rel}(Nspin)$. Аппроксимируйте данные зависимости методом наименьших квадратов.

***Задача* 15.13**

В описанном выше мы оценивали равновесные значения энергии, удельной теплоемкости, магнитного момента и намагниченности, как средние значения, вычисляемые по соответствующим мгновенным значениям после достижения системой состояния равновесия. В данном подходе априори полагает-

Рис. 15.12. Возможные изменения энергии в двумерной модели Изинга при перевороте одного спина

ся, что каждое из мгновенных значений является статистически независимым от остальных значений, то есть «время корреляции» между выборками равно нулю. При более правильном подходе следует вычислить автокорреляционные функции

$$C_E(\tau) = \frac{\langle E(t)\,E(t+\tau)\rangle - \langle E\rangle^2}{\langle E^2\rangle - \langle E\rangle^2},$$

$$C_M(\tau) = \frac{\langle M(t)\,M(t+\tau)\rangle - \langle M\rangle^2}{\langle M^2\rangle - \langle M\rangle^2},$$

где $\langle\,\rangle$ — усреднение по всем возможным значениям t; оценить число шагов метода Монте-Карло, за которое значение автокорреляционной функции уменьшается в e раз; и вычислить среднее значение энергии, удельную теплоемкость, средний магнитный момент, удельную намагниченность, проводя выборку значений, отстоящих друг от друга не менее чем на t_{corr}. Дополните описанный выше документ, собственной функцией, возвращающей значения

автокорреляционной функции. Вычислите значения автокорреляционных функций системы, состоящей из $N = 64$ спинов, при температурах T, меняющихся в интервале $[1{,}5, 3{,}5]$ с шагом ΔT. Как зависит время корреляции от температуры? Дополните описанный выше документ функцией, возвращающей среднее значение энергии системы, удельную теплоемкость, средний магнитный момент, удельную намагниченность, вычисляемые по значениям, отстоящими друг от друга по времени не менее чем на $t_{corr}(T)$. Сравните полученные термодинамические характеристики системы с аналогичными характеристиками, найденными усреднением по всей реализации.

***Задача* 15.14**

Для более реалистичного описания процесса релаксации системы к равновесию следует проводить выбор спинов, подлежащих опрокидыванию, случайным образом. Однако, если мы интересуемся равновесными свойствами системы, то можно сэкономить машинное время, выбирая спины последовательно. Как влияет на время корреляции способ выбора спинов, подлежащих опрокидыванию. Почему нежелателен последовательный выбор спинов в одномерной модели Изинга?

В пределе бесконечно большой решетки ($Nspin \to \infty$) модель Изинга имеет точное решение [13]. В предельном случае при $h = 0$ энергия системы E дается выражением

$$\frac{E}{Nspin} = -J \cdot \coth(2x)\left[1 + \frac{2}{\pi}\kappa' \cdot K_1(\kappa)\right], \tag{15.28}$$

где

$$x = \tanh\left(\frac{J}{T}\right)$$

$$\kappa = 2\frac{\sinh(2x)}{\cosh^2(2x)},$$

$$\kappa' = 2\tanh^2(2x) - 1,$$

$$\cosh(x) = \frac{e^x + e^{-x}}{2},$$

$$\sinh(x) = \frac{e^x - e^{-x}}{2},$$

$$\tanh(x) = \frac{e^x - e^{-x}}{e^x + e^{-x}},$$

$$\coth(x) = \frac{e^x + e^{-x}}{e^x - e^{-x}};$$

$K_1(\kappa)$ — полный эллиптический интеграл первого рода:

$$K_1(\kappa) = \int\limits_0^{\frac{\pi}{2}} \frac{d\phi}{\sqrt{1 - \kappa^2 \sin^2\phi}}.$$

Удельная намагниченность при $T < T_c$

$$\frac{M}{Nspin} = \pm \frac{\left(1+z^2\right)^{\frac{1}{4}}\left(1-6z^2+z^4\right)^{\frac{1}{8}}}{\left(1-z^2\right)^{\frac{1}{2}}}, \tag{15.29}$$

где

$$z = e^{-2\tanh\left(\frac{2J}{T}\right)}.$$

***Задача* 15.15**

Для бесконечной решетки вычислите и отобразите графически зависимости энергии, удельной теплоемкости и намагниченности на один спин при $h = 0$ *от температуры в диапазоне температур* $[0{,}1; 3{,}5]$.

***Задача* 15.16**

Решив задачу 15.15, вы обнаружили, что в двумерной модели Изинга возникает фазовый переход: намагниченность системы в точке фазового перехода становится равной нулю, а теплоемкость системы и ее магнитная восприимчивость расходятся. Проведите численные исследования поведения модели Изинга вблизи точки фазового перехода и сравните ваши результаты с точным решением. Каков размер магнитных доменов при температуре, близкой к критической? Как влияет конечный размер решетки на особенности зависимостей термодинамических величин от температуры?

***Задача* 15.17**

Используйте описанный выше документ для исследования термодинамических свойств модели Изинга при $h \neq 0$.

ЛИТЕРАТУРА К ГЛАВЕ 15

1. *Creutz M.* Microcanonical Monte Simulation, Phys. Rev. Letts. 50, 1411. 1983.
2. *Bhanot G., Creutz M., Neuberger H.*, Microcanonical simulation of Ising system, Nucl. Phys. 235, 417. 1984.
3. *Metropolis N., Rosenbluth A. W., Rosenbluth M. N., Teller A. H., Keller E.* Equation of state calculations for fast computing machines, J. Chem. Phys. 6, 1087. 1953.
4. *Гулд Х., Тобочник Я.* Компьютерное моделирование в физике. Т. 2. М.: Мир, 1990.
5. *Киттель Ч.* Введение в физику твердого тела. М.: Наука, 1978.
6. *Кунин С.* Вычислительная физика. М.: Мир, 1992.
7. *Ландау Л. Д., Лифшиц Е. М.* Статистическая физика. М.: Наука, 2000.
8. Методы Монте-Карло в статистической физике / Под ред. К. Биндера. М.: Мир, 1982.

9. *Дайсон Ф., Монторолл Э., Кац М., Фишер М.* Устойчивость и фазовые переходы. М.: Мир, 1973.

10. *Матвеев А. Н.* Молекулярная физика. М.: Высшая школа, 1981.

11. *Вильсон К., Когут Дж.* Ренормализационная группа и е-разложения. М.: Мир, 1975.

12. *Хеерман Д. В.* Методы компьютерного эксперимента в теоретической физике. М.: Наука, 1990.

13. *Хуанг К.* Статистическая механика. М.: Мир, 1966.

Глава 16

МОДЕЛИРОВАНИЕ КВАНТОВЫХ СИСТЕМ

16.1. ВВЕДЕНИЕ

В квантовой механике любая система полностью описывается заданием волновой функции $\psi(\vec{r},t)$ (в общем случае комплексной), квадрат модуля $|\psi(\vec{r},t)|^2$ которой определяет плотность вероятности обнаружить частицу в момент времени t в некоторой точке пространства с радиусом-вектором $\vec{r}$. Так как частица может находиться в произвольной точке пространства, вероятность того, что частица находится в элементарном объеме $dV = d^3\vec{r}$ в точке пространства с радиусом-вектором $\vec{r}$ в момент времени t, равна

$$P(\vec{r},t)\,d^2\vec{r} = C^2|\psi(\vec{r},t)|^2 d^3\vec{r}, \tag{16.1}$$

где C — нормировочная постоянная, выбираемая из условия

$$\int\limits_{-\infty}^{\infty} \psi^*(\vec{r},t)\,\psi(\vec{r},t)\;d^3\vec{r} = 1,$$

$$\int\limits_{-\infty}^{\infty} \psi^*(\vec{r},t)\,\psi(\vec{r},t)\;d^3\vec{r} = 1, \tag{16.2}$$

здесь $\psi^*(\vec{r},t)$ — функция комплексно-сопряженная $\psi(\vec{r},t)$.

Если частица движется в потенциале $U(\vec{r},t)$, то временная эволюция функции $\psi(\vec{r},t)$ описывается нестационарным уравнением Шредингера

$$i\hbar\frac{\partial\psi(\vec{r},t)}{\partial t} = -\frac{\hbar^2}{2m}\Delta\psi(\vec{r},t) + U(\vec{r},t)\,\psi(\vec{r},t), \tag{16.3}$$

где m — масса частицы, $\hbar$ — постоянная Планка, деленная на 2π, $\Delta = \frac{\partial^2}{\partial x^2} + \frac{\partial^2}{\partial y^2} + \frac{\partial^2}{\partial z^2}$ — оператор Лапласа.

Каждой физической классической характеристике частицы (например, координате, скорости, импульсу, моменту импульса, энергии и др.) ставится в соответствии некоторый оператор. Наблюдаемое значение величины величины A есть среднее значение соответствующего оператора $\widehat{A}$:

$$\langle A\rangle = \int \psi^*(\vec{r},t)\widehat{A}\psi(\vec{r},t)\,d^3\vec{r}. \tag{16.4}$$

Например, оператор, соответствующий импульсу p, имеет вид

$$\widehat{p} = -i\hbar\frac{\partial}{\partial\vec{r}},$$

а его среднее

$$\langle P\rangle = -i\hbar\int\psi^*\left(\vec{r},t\right)\frac{\partial\psi\left(\vec{r},t\right)}{\partial\,\vec{r}}d^3\vec{r}. \tag{16.5}$$

В одномерном случае уравнение Шредингера принимает вид

$$i\hbar\frac{\partial\psi\left(x,t\right)}{\partial\,t} = -\frac{\hbar^2}{2m}\frac{\partial^2}{\partial\,x^2}\psi\left(x,t\right) + U\left(x,t\right)\psi\left(x,t\right). \tag{16.6}$$

Для независящего от времени потенциала можно искать решения уравнения (16.3), (16.6) в виде

$$\psi\left(\vec{r},t\right) = \varphi\left(\vec{r},t\right)e^{-\frac{iEt}{\hbar}}, \tag{16.7}$$

$$\psi\left(x,t\right) = \varphi\left(x,t\right)e^{-\frac{iEt}{\hbar}}, \tag{16.8}$$

соответственно. Здесь E — энергия частицы.

Частица, находящаяся в стационарном состоянии, описываемом волновой функцией (16.7), (16.8), имеет вполне конкретное значение энергии E. Например, подставив (16.8) в (16.8), получаем стационарное уравнение Шредингера:

$$-\frac{\hbar^2}{2m}\frac{\partial^2\varphi\left(x\right)}{\partial\,x^2} + U\left(x\right)\varphi\left(x\right) = E\varphi\left(x\right). \tag{16.9}$$

Уравнение (16.9) можно записать в виде

$$\left[-\frac{\hbar^2}{2m}\frac{\partial^2}{\partial\,x^2} + U\left(x\right)\right]\varphi\left(x\right) = E\varphi\left(x\right). \tag{16.10}$$

Из (16.10) видно, что

$$-\frac{\hbar^2}{2m}\frac{\partial^2}{\partial\,x^2} + U\left(x\right)$$

является оператором энергии (оператор Гамильтона), $\varphi(x)$ — собственная функция гамильтониана, E — собственное значение оператора $\hat{H}$, то есть

$$\widehat{H}\varphi\left(x\right) = E\varphi\left(x\right). \tag{16.11}$$

Вообще говоря, оператор $\hat{H}$ может иметь n собственных функций $\phi_n(x)$ и n соответствующих им собственных значений энергии E_n, которые в ряде случаев могут совпадать (вырожденные состояния) или принимать континуум непрерывных значений (непрерывный спектр). Основным состоянием называется состояние с наименьшей энергией E_1.

Общее решение (16.10) можно представить в виде суперпозиции собственных функций оператора, отвечающего выбранной физической величине. Например, если $\hat{H}$ не зависит от времени, то можно записать

$$\psi\left(x,t\right) = \sum_n c_n\varphi_n\left(x\right)e^{\frac{iE_nt}{\hbar}}, \tag{16.12}$$

где $\varphi_n(x)$ — собственные функции оператора, а суммирование проводится по всем дискретным состояниям.

В случае непрерывного спектра суммирование заменяется интегрированием по возможным значениям энергии (спектру).

Собственные функции $\varphi_n(x)$, соответствующие различным собственным значениям, являются ортогональными, то есть

$$\int\limits_{-\infty}^{+\infty} \varphi^*{}_n\,(x)\,\varphi_{n'}\,(x)\,dx = \delta_{nn'}. \qquad (16.13)$$

Коэффициенты c_n в (16.12) определяются по известным значениям функции $\varphi(x,t)$. Например, если известны значения $\varphi(x,0)$, то, воспользовавшись свойством ортогональности (16.13), легко получить

$$c_n = \int \varphi^*\,(x)\,\psi\,(x,0)\,dx. \qquad (16.14)$$

Коэффициент c_n можно интерпретировать как амплитуду вероятности обнаружить в измерении значение полной энергии, равное E_n.

16.2. МЕТОДЫ ЧИСЛЕННОГО РЕШЕНИЯ СТАЦИОНАРНОГО УРАВНЕНИЯ ШРЕДИНГЕРА

Рассмотрим решения одномерного стационарного уравнения Шредингера (16.10) частицы, движущейся в одномерном потенциале $U(x)$. Будем при этом полагать, что его форма имеет вид, представленный на рис. 16.1: в точках $x_{\min}$, $x_{\max}$ потенциал становится бесконечно большим.

Рис. 16.1. К постановке задачи решения стационарного уравнения Шредингера

Это означает, что в точках $x_{\min}$, $x_{\max}$ расположены вертикальные стенки, а между ними находится яма конечной глубины. Для удобства дальнейшего решения запишем уравнение Шредингера (16.10) в виде

$$\frac{d^2\psi}{d\,x^2} + k^2\,(x)\,\psi\,(x) = 0, \qquad (16.15)$$

где

$$k^2\,(x) = \frac{2m}{\hbar^2}\,[E - U\,(x)]\,. \qquad (16.16)$$

С математической точки зрения задача состоит в отыскании собственных функций оператора

$$\frac{d^2}{dx^2} + k^2\,(x)\,,$$

отвечающих граничным условиям

$$\psi_n\left(x_{\min}\right) = \psi_n\left(x_{\max}\right) = 0, \tag{16.17}$$

и соответствующих собственных значений энергии E_n.

Так как

$$k^2(x) \geq 0$$

при $x \in [x_{\text{left}}, x_{\text{right}}]$,

$$k^2(x) < 0$$

при $x \in [x_{\min}, x_{\text{left}}]$, $x \in [x_{\text{right}}, x_{\max}]$, то можно ожидать, что собственному решению данной задачи соответствует собственная функция:

1) осциллирующая в классически разрешенной области движения $E \geq U(x)$, когда $x \in [x_{\text{left}}, x_{\text{right}}]$;

2) экспоненциально затухающая в запрещенных областях, где $E < U(x)$, когда $x \in [x_{\min}, x_{\text{left}}]$, $x \in [x_{\text{right}}, x_{\max}]$;

3) тождественно равная нулю ($\psi \equiv 0$), когда $x \leq x_{\min}$, $x \geq x_{\max}$.

Так как в рассматриваемом случае все состояния частицы в потенциальной яме оказываются связанными (то есть локализованными в конечной области пространства), спектр энергий является дискретным. Так как все состояния частицы в потенциальной яме оказываются связанными (то есть локализованными в конечной области пространства), спектр энергий является дискретным.

В общем случае частица, находящаяся в потенциальной яме конечных размеров

$$\begin{array}{lll} U(x) < 0, & if & x \in [x_{\min}, x_{\max}], \\ U(x) = 0, & if & x \notin [x_{\min}, x_{\max}], \end{array}$$

имеет дискретный спектр при $E < 0$ и непрерывный спектр при $E \geq 0$.

Традиционно для численного решения задачи о нахождении собственных значений уравнения Шредингера используется метод пристрелки. Идея метода пристрелки состоит в следующем. Допустим, в качестве искомого значения ищется одно из связанных состояний, поэтому в качестве пробного начального значения энергии выбираем отрицательное собственное значение.

Проинтегрируем уравнение Шредингера каким-либо известным численным методом на интервале $[x_{\min}, x_{\text{left}}]$. По ходу интегрирования от $x_{\min}$ в сторону больших значений x сначала вычисляется решение $\psi_+(x)$, экспоненциально нарастающее в пределах классически запрещенной области. После перехода через точку поворота x_{right}, ограничивающую слева область движения, разрешенную классической механикой, решение уравнения становится осциллирующим. Если продолжить интегрирование далее за правую точку поворота x_{right}, то решение становится численно неустойчивым. Это обусловлено тем, что даже при точном выборе собственного значения, для которого выполняется условие $\psi_+(x_{\max})$, решение в области $[x_{\text{right}}, x_{\max}]$ всегда может содержать некоторую примесь экспоненциально растущего решения, не имеющего физического содержания.

Отмеченное обстоятельство является общим правилом: интегрирование по направлению внутрь области, запрещенной классической механикой, будет неточным. Следовательно, для каждого значения энергии более разумно вычислить еще одно решение $\psi_-(x)$, интегрируя уравнение (16.10) от $x_{\max}$ в сторону уменьшения x. Критерием правильного выбора данного значения энергии является совпадение значений функций $\psi_+(x)$ и $\psi_-(x)$ в некоторой промежуточной точке x_m. (Так как функции $\psi_+(x)$ и $\psi_-(x)$ являются решениями однородного уравнения (16.10), их всегда можно отнормировать так, чтобы в точке x_m выполнялось условие $\psi_+(x_m) = \psi_-(x_m)$.) Обычно в качестве данной точки выбирают левую точку поворота x_{left}.

Помимо совпадения значений функций в точке x_m для обеспечения гладкости сшивки решений потребуем совпадения значений их производных $\psi'_+(x_m) = \psi'_-(x_m)$:

$$\left.\frac{d\psi_+}{dx}\right|_{x=x_m} = \left.\frac{d\psi_-}{dx}\right|_{x=x_m}. \tag{16.18}$$

Используя в (16.18) простейшие левую и правую конечно-разностные аппроксимации производных функций $\psi_+(x)$, $\psi_-(x)$ в точке x_m, находим эквивалентное условие гладкости сшивки решений:

$$f \equiv \frac{\psi_+(x_m) + \psi_-(x_m) - [\psi_+(x_m - h) + \psi_-(x_m + h)]}{\max|\psi|} = 0. \tag{16.19}$$

Число $\max(\psi)$ является масштабирующим множителем, который выбирается из условия $f \cong 1$. Если точки поворота отсутствуют, то есть $E > 0$, то в качестве x_m можно выбрать любую точку отрезка $[x_{\min}, x_{\max}]$. Для потенциалов, имеющих более двух точек поворота и, соответственно, три или более однородных решений, общее решение получается сшивкой отдельных кусков.

В описанном ниже документе для интегрирования дифференциального уравнения второго порядка мы используем метод Нумерова. Для получения вычислительной схемы аппроксимируем вторую производную трехточечной разностной формулой:

$$\frac{y_{n+1} - 2y_n + y_{n-1}}{h^2} = y''_n + \frac{h^2}{12} y_n^{(\mathrm{IV})} + O\left(h^4\right). \tag{16.20}$$

Из уравнения (16.15) имеем

$$y_n^{(\mathrm{IV})} = \frac{d^2}{d\,x^2}\left(-k^2(x)\,y\right)\Big|_{x=x_n} = -\frac{\left(k^2 y\right)_{n+1} - 2\left(k^2 y\right)_n + \left(k^2 y\right)_{n-1}}{h^2}. \tag{16.21}$$

Подставив (16.21) в (16.20) и перегруппировав члены, получаем

$$\left(1 + \frac{h^2}{12}k_{n+1}^2\right) y_{n+1} - 2\left(1 - \frac{5h^2}{12}k_n^2\right) y_n + \left[1 + \frac{h^2}{12}k_{n-1}^2\right] y_{n-1} = 0. \tag{16.22}$$

Разрешив (16.22) относительно y_{n+1} или y_{n-1}, найдем рекуррентные формулы для интегрирования уравнения (16.10) вперед или назад по x с локальной погрешностью $O\left(h^6\right)$. Отметим, что погрешность данного метода оказывается на порядок выше, чем погрешность метода Рунге–Кутта четвертого порядка. Кроме того, данный алгоритм

более эффективен, потому что значения функции $k^2(x)$ вычисляются только в узлах сетки.

Для нахождения численного решения оказывается удобным провести обезразмеривание уравнения (16.10), используя в качестве единиц измерения расстояния a — ширину потенциальной ямы, в качестве единиц измерения энергии $|U_0|$ — модуль минимального значения потенциала. В выбранных единицах измерения уравнение (16.10) принимает следующий вид:

$$\frac{d^2\psi}{d\,\tilde{x}^2} + \gamma^2\left(\tilde{E} - \tilde{U}\left(\tilde{x}\right)\right)\psi = 0, \tag{16.23}$$

где

$$\begin{aligned} \tilde{x} &= \frac{x}{a}, \\ \tilde{E} &= \frac{E}{U_0}, \\ \tilde{U}\left(\tilde{x}\right) &= \frac{U\left(\tilde{x}\right)}{U_0}, \\ \gamma^2 &= \frac{2ma^2U_0}{\hbar^2}. \end{aligned} \tag{16.24}$$

Таким образом, вычислительный алгоритм для нахождения собственных функций и собственных значений уравнения Шредингера реализуется следующей последовательностью действий:

1. Задать выражение, описывающее безразмерный потенциал $U(\tilde{x})$.
2. Задать значение γ^2.
3. Задать пространственную сетку, на которой проводится интегрирование уравнения (16.9).
4. Задать $x_{\min}, x_{\max}$.
5. Задать начальное значение энергии E_{start}.
6. Задать конечное значение энергии E_{fin}.
7. Задать шаг изменения энергии ΔE.
8. Проинтегрировать уравнение (16.10) для значения энергии E_i слева направо на отрезке $[x_{\min}, x_m]$.
9. Проинтегрировать уравнение (16.10) для значения энергии E_{i+1} справа налево на отрезке $[x_{\max}, x_m]$.
10. Вычислить значения переменной f_i для значения энергии E_i.
11. Увеличить текущее значение энергии на ΔE: $E_{i+1} = E_i + \Delta E$.
12. Проинтегрировать уравнение (16.10) для значения энергии E_{i+1} слева направо на отрезке $[x_{\min}, x_m]$.

13. Проинтегрировать уравнение (16.10) для значения энергии E_{i+1} справа налево на отрезке $[x_{\max}, x_m]$.

14. Вычислить значения переменной f_{i+1} для значения энергии E_{i+1}.

15. Сравнить знаки f_i, f_{i+1}.

16. Если $f_i \cdot f_{i+1} > 0$ и $E_{i+1} \leq E_{\max}$, увеличить текущее значение энергии на ΔE и повторить действия, описанные в п. 8—15.

17. Если $f_i \cdot f_{i+1} < 0$ уточнить собственное значение методом линейной интерполяции.

18. Если $E_{i+1} \leq E_{\max}$, повторить действия, описанные в п. 8—18.

19. Если $E_{i+1} > E_{\max}$, закончить вычисления.

Для реализации описанного вычислительного алгоритма в MATLAB необходимо создать три m-файла:

1) файл `Num.m`, содержащий описание функции, возвращающей для заданной энергии величину f, вычисляемую в соответствии с (16.19), и значения волновой функции в узлах координатной сетки;

2) файл `U.m`, содержащий описание функции, возвращающей значение потенциала;

3) файл `Elevel.m`, содержащий описание функции, возвращающей собственные значения и собственные функции уравнения Шредингера.

Ниже приведены листинги данных файлов.

```
% листинг файла Num.m
function [f,psi]=Num(gamma,E,V,Xmin,Xmax,Ngreed)
% функция, возвращающая для заданной энергии
% величину f, вычисляемую
% в соответствии с (16.19), и
% значения волновой функции (psi)
% в узлах координатной сетки
% gamma - коэффициент, входящий
% в обезразмеренное уравнение
% Шредингера (16.23)
% E - заданное значение энергии
% V - вектор, содержащий значения потенциала
% в узлах координатной сетки
% Xmin - левая граница координатной сетки
% Xmax - правая граница координатной сетки
% Ngreed - число узлов координатной сетки

dx=(Xmax-Xmin)/Ngreed;
c=dx.^2*gamma^2/12;
```

```
Imatch=1;
psi(1)=0;
psi(2)=9.99999*10^-10;

% интегрирование уравнения Шредингера
% справа налево
Kim1=c*(E-V(1));
Ki=c*(E-V(2));
for i=2:Ngreed-1
   Kip1=c*(E-V(i+1));
   if and(Ki*Kip1<=0,Ki>0)
      Imatch=i;
      i=Ngreed;
   end;
   if i<=Ngreed-1
      psi(i+1)=(psi(i)*(2-10*Ki)-...
                psi(i-1)*(1+Kim1))/(1+Kip1);
      if abs(psi(i+1))>=10^10
        for k=1:i+1
           psi(k)=psi(k)*9.99999*10^-6;
        end;
     end;
     Kim1=Ki;
     Ki=Kip1;
  end;
end;
if Imatch==1
  Imatch=Ngreed-10;
end;

% интегрирование уравнения Шредингера
% слева направо
psi_Left=psi(Imatch);
psi(Ngreed)=0;
psi(Ngreed-1)=9.99999*10^-10;
Kip1=c*(E-V(Ngreed));
Ki=c*(E-V(Ngreed-1));
for i=Ngreed-1:-1:Imatch+1
   Kim1=c*(E-V(i-1));
   psi(i-1)=(psi(i)*(2-10*Ki)-...
             psi(i+1)*(1+Kip1))/(1+Kim1);
   if abs(psi(i))>10^10
     for k=Ngreed:-1:i
        psi(k)=psi(k)*9.99999*10^-6;
     end;
```

```
  end;
  Kip1=Ki;
  Ki=Kim1;
end;
if psi_Left<0
   for i=Imatch:Ngreed
      psi(i)=-psi(i);
   end;
end;
psi1=abs(psi);
Psimax=max(psi1);

% вычисление разности между волновыми
% функциями, полученными
% интегрированием уравнения Шредингера
% слева направо и справа налево,
% в узле с номером Imatch
f=(psi_Left+psi(Imatch)-...
   (psi(Imatch-1)+...
   psi(Imatch+1)))/Psimax;

% листинг файла U.m
function z = U(x,Xmin,Xmax)
% функция, возвращающая значения потенциала
% в узлах координатной сетки
% x - вектор, содержащий координаты узлов
% Xmin - левая граница потенциала
% Xmax- правая граница потенциала

if and(Xmin<=x,x<=Xmax)
  z=-1;
else
  z=10^309;
end;

% листинг файла Elevel.m
function [EL,Psi]=Elevel(gamma,dE,V,Emax,Xmin,Xmax,Ngreed)
% функция, возвращающая
% собственные значения и собственные функции
% уравнения Шредингера
% gamma - коэффициент, входящий
% в обезразмеренное уравнение
% Шредингера (16.23)
% dE - приращение энергии
% V - вектор, содержащий значения потенциала
% в узлах координатной сетки
```

```
% Emax - максимальное значение энергии
% Xmin - левая граница координатной сетки
% Xmax - правая граница координатной сетки
% Ngreed - число узлов координатной сетки

Tolf=10^-6;
Emin=-1;
E=Emin+dE;
m=1;
Start=0;
while E<Emax
  if Start==0
    E1=E;
    [f,psi]=Num(gamma,E,V,Xmin,Xmax,Ngreed);
    E=E+dE;
    F1=f;
    Start=1;
  end;
  E2=E;
  [f,psi]=Num(gamma,E,V,Xmin,Xmax,Ngreed);
  F2=f;
  if F1*F2>0
    E1=E;
    F1=F2;
    E=E+dE;
  end;
  if F1*F2<0
  % уточнение собственного
  % значения энергии
  % и вычисление значений
  % собственной функции
  % в узлах координатной сетки
    a=(E2-E1)/(F2-F1);
    E=E1-a*F1;
    [f,psi]=Num(gamma,E,V,Xmin,Xmax,Ngreed);
    F1=F2;
    E1=E2;
    EL(m,1)=m;
    EL(m,2)=E;
    if m==1
      Psi0=psi';
    else
      Psi0=cat(2,Psi0,psi');
    end;
    m=m+1;
```

```
    E=E+dE;
  end;
end;
dx=(Xmax-Xmin)/(Ngreed-1);
N1=size(Psi0,2);
% нормировка собственных волновых функций
for i=1:N1
  S=Psi0(:,i);
  S1=S.^2;
  Norm=0;
  for j=1:Ngreed-1
    Norm=Norm+0.5*(S1(j)+S1(j+1))*dx;
  end;
  S=S./(Norm^0.5);
  if i==1
    Psi=S;
  else
    Psi=cat(2,Psi,S);
  end;
end;
```

Далее необходимо выполнить следующую последовательность команд, сохраненную нами в файле `Glava16_1.m`:

```
% листинг файла Glava16_1.m
Xmin=-0.5; % левая граница координатной сетки
Xmax=0.5; % правая граница координатной сетки
gamma=20;
Ngreed=500; % число узлов координатной сетки
% вычисление координат узлов сетки
i=1:Ngreed;
x(i)=Xmin+(Xmax-Xmin)/(Ngreed-1)*(i-1);
dE=10^-3; % приращение энергии
Emax=-0; % максимальное значение энергии
V(i)=U(x(i),Xmin,Xmax); % вычисление значения
                        % потенциала в узлах
                        % координатной сетки
% вычисление собственных значений и
% собственных функций
% уравнения Шредингера
[EL,Psi]=Elevel(gamma,dE,V,Emax,Xmin,Xmax,Ngreed);

% вывод в командном окне
% собственных значений
% уравнения Шредингера
EL
```

```
% визуализация собственных функций
% уравнения Шредингера
figure
plot(x(i),Psi(:,1),'k',...
     x(i),Psi(:,2),'--k',...
     x(i),Psi(:,3),'-.k');

figure
plot(x(i),Psi(:,4),'k',...
     x(i),Psi(:,5),'--k',...
     x(i),Psi(:,6),'-.k');
```

Ниже представлены результаты выполнения описанной последовательности команд.

1. Сообщения, выводимые в командное окно:

```
EL =
    1.0000   -0.9751
    2.0000   -0.9005
    3.0000   -0.7761
    4.0000   -0.6020
    5.0000   -0.3782
    6.0000   -0.1046
```

2. графики волновых функций (рис. 16.2, 16.3)

Рис. 16.2. Волновые функции, соответствующие первому, второму и третьему собственным значениям энергии

Рис. 16.3. Волновые функции, соответствующие четвертому, пятому и шестому собственным значениям энергии

***Задача* 16.1**

1. *Проверьте условие ортогональности собственных функций уравнения Шредингера (16.12).*

2. *Оцените точность выполнения данного условия. Исследуйте зависимость точности ортогональности собственных функций уравнения Шредингера от шага изменения энергии ΔE и числа узлов пространственной сетки Ngreed.*

***Задача* 16.2**

Для бесконечной прямоугольной ямы шириной a удается найти аналитические выражения для собственных функций уравнения и собственных значений Шредингера [1, задача 18]:

$$n = 1, \qquad E_1 = \frac{\hbar^2\pi^2}{2ma^2}, \qquad \psi_1 = \left(\frac{a}{2}\right)^{-\frac{1}{2}} \cos\left(\frac{\pi\, x}{\mathrm{a}}\right),$$

$$n = 2, \quad E_2 = 4\frac{\hbar^2\pi^2}{2ma^2} = 4E_1, \quad \psi_2 = \left(\frac{a}{2}\right)^{-\frac{1}{2}} \sin\left(\frac{2\pi\, x}{a}\right),$$

$$n = 3, \quad E_3 = 9\frac{\hbar^2\pi^2}{2ma^2} = 9E_1, \quad \psi_3 = \left(\frac{a}{2}\right)^{-\frac{1}{2}} \cos\left(\frac{3\pi\, x}{a}\right),$$

$$n = 4, \quad E_4 = 16\frac{\hbar^2\pi^2}{2ma^2} = 16E_1, \quad \psi_4 = \left(\frac{a}{2}\right)^{-\frac{1}{2}} \sin\left(\frac{4\pi\, x}{a}\right),$$

$$n = 5, \quad E_5 = 25\frac{\hbar^2\pi^2}{2ma^2} = 25E_1, \quad \psi_5 = \left(\frac{a}{2}\right)^{-\frac{1}{2}} \cos\left(\frac{5\pi\, x}{\mathrm{a}}\right),$$

$$n = 6, \quad E_5 = 36\frac{\hbar^2\pi^2}{2ma^2} = 36E_1, \quad \psi_6 = \left(\frac{a}{2}\right)^{-\frac{1}{2}} \sin\left(\frac{6\pi\, x}{a}\right).$$

Используя безразмерные переменные, определяемые в соответствии с уравнением (16.23), получите выражения для безразмерных значений энергии и собственных функций. Сравните численные значения энергий и волновых функций с соответствующими значениями, вычисляемыми по аналитическим выражениям. Оцените точность численных значений энергии и волновых функций. Почему точность зависит от номера собственного значения энергии?

***Задача* 16.3**

Для различных собственных значений энергии:

1. *Постройте зависимость плотности вероятности $P(x)$ от* x.

2. *Вычислите $\langle x\rangle$, $\langle x^2\rangle$ в соответствии с (16.4).*

3. *Вычислите $\langle p\rangle$, $\langle p^2\rangle$ в соответствии с (16.4).*

4. *Постройте зависимость разности между соседними уровнями энергии от номера уровня.*

***Задача* 16.4**

Определите влияние малого возмущения на собственные значения и собственные волновые функции частицы в бесконечной прямоугольной яме, в которой находится небольшой прямоугольный горб высотой U_b и шириной b, симметрично относительно точки $x = 0$.

1. *Выберите $U_b \ll U_0$, $b \ll a$ и определите как зависит энергия основного состояния и соответствующие собственные функции от U_b, b.*

2. *Пусть ψ_0 — собственная функция, отвечающая наименьшему собственному значению энергии, ψ_b — собственная функция основного состояния при $U_b \neq 0$. Вычислите значение выражения*

$$\frac{\int\limits_0^a \psi_b(x)\,\psi(x)\;dx}{\int\limits_0^a \psi^2(x)\,dx}$$

при различных значениях b. Как связана данная величина с изменением относительных значений энергии?

Указание: *предваряя решение задачи, запишите ее условие в безразмерных единицах.*

***Задача* 16.5**

Так как при $x = |a/2|$ глубина ямы конечна, функция $\psi(x)$ не равна нулю в запрещенной классической механикой области, в которой $E < U_0$ и $x > |a/2|$. Измените функцию, описывающую зависимость потенциала от координаты, и рассмотрите движение частицы в прямоугольной потенциальной яме конечной глубины.

1. *Вычислите собственные функции и собственные значения энергии, начав с уровней близких к дну ямы. Как меняются волновые функции по мере удаления от дна ямы?*

2. *Определите «глубину проникания под барьер» как расстояние от $x = |a/2|$ до точки, в которой значение функции составляет $1/e \approx 0,37$ от ее максимального значения. Как зависит глубина проникания от величины U_0?*

***Задача* 16.6**

1. *Получите несколько первых собственных значений энергии и собственные функции гармонического осциллятора, описываемого потенциалом $U(x) = kx^2/2$. Сравните полученные значения с известными аналитическими результатами.*

2. *Постройте зависимость плотности вероятности $P(x)$ от x.*
3. *Вычислите $\langle x \rangle$, $\langle x^2 \rangle$ в соответствии с (16.4).*
4. *Вычислите $\langle p \rangle$, $\langle p^2 \rangle$ в соответствии с (16.4).*
5. *Постройте зависимость разности между соседними уровнями энергии от номера уровня.*

***Задача* 16.7**

Получите несколько первых собственных значений энергии и собственные функции ангармонического осциллятора, описываемого потенциалом $U(x) = kx^2/2 + bx^3$. Вычислите энергию основного состояния для $k = 1$, $b = 0{,}1$, $0{,}2$, $0{,}3$. Как зависят энергия основного состояния и форма волновых функций от b?

***Задача* 16.8**

Получите численное решение для основного состояния линейного потенциала

$$U(x) = |x|.$$

(Данный потенциал может быть использован при моделировании связанной системы кварк–антикварк.)

16.3. МОДЕЛИРОВАНИЕ КОЛЕБАНИЙ ДВУХАТОМНОЙ МОЛЕКУЛЫ В КВАЗИКЛАССИЧЕСКОМ ПРИБЛИЖЕНИИ

В данном разделе мы рассматриваем задачу об описании колебаний двухатомной молекулы, методы аналитического решения которой подробно рассматриваются в работах [2–4]. Следуя выбранному подходу, обсудим постановку задачи, при этом основное внимание сосредоточим на вычислительных проблемах, возникающих при использовании для ее решения в MATLAB.

Двухатомная молекула состоит из двух ядер, связанных электронами. Поскольку масса ядер на три порядка больше массы электронов, можно считать, что распределение электронов мгновенно подстраивается к изменению положения тяжелых ядер. (В физике данная модель получила название приближение Бора–Оппенгеймера.)

Это позволяет свести рассматриваемую задачу к задаче о движении двух ядер в потенциальном поле, зависящем от расстояния между ними. Причем на больших расстояниях потенциал должен быть притягивающим (силы Ван-дер-Ваальса), а на близких расстояниях — отталкивающим (кулоновские силы и обменное взаимодействие). Таким свойством обладает потенциал Ленарда-Джонса

$$V(r) = 4V_0 \left[\left(\frac{\sigma}{r}\right)^{12} - \left(\frac{\sigma}{r}\right)^{6} \right], \tag{16.25}$$

где V_0, σ — параметры потенциала (рис. 16.4), который достигает своего минимума в точке $r_{\min} = 2^{1/6}\sigma$. Напомним, что данный потенциал уже рассматривался нами при

моделировании неидеального газа методом молекулярной динамики в главе 10.

Принимая во внимание большую массу ядер, можно провести дальнейшее упрощение задачи, считая движение молекулы суммой двух независимых друг от друга движений: медленного вращения молекулы как единого целого и более быстрого колебательного движения ядер атомов относительно центра масс. Первое движение хорошо описывается как квантово-механическое вращение жесткой системы типа «гантель». Колебательное движение ядер описывается уравнением Шредингера (16.9), в котором $U(x)$ — потенциал Ленарда-Джонса.

Рис. 16.4. Потенциал Ленарда-Джонса

Напомним, что классическое финитное движение в потенциале $V(r)$ возможно в диапазоне энергий $-V0 < E < 0$. При этом изменение межъядерного расстояния имеет характер периодического движения между точками поворота, которые мы будем в дальнейшем обозначать $r_{\text{left}}, r_{\text{right}}$. Для частицы, двигающейся в потенциальном поле, выполняется закон сохранения энергии

$$E = \frac{p^2}{2m} + V(r), \tag{16.26}$$

где p — импульс относительного движения ядер.

Следовательно, в фазовом пространстве движение представляет собой некоторую замкнутую кривую, в каждой точке которой выполняется условие (16.26). В явной форме уравнение траектории в фазовом пространстве получается решением уравнения (16.26) относительно p:

$$p = \pm[2m\,(E - V\,(r))]^{1/2}. \tag{16.27}$$

В классическом случае движение частицы возможно при любом значении энергии в диапазоне $[-V_0; 0]$. Для нахождения дискретного энергетического спектра E_n уравнения Шредингера вводится переменная величина — действие, зависящее от энергии системы [5]:

$$S\,(E) = \oint k(r)dr, \tag{16.28}$$

где $k(r) = \hbar^{-1}p(r)$ — волновое число волны де Бройля, а интегрирование ведется по одному полному периоду колебания.

По своему физическому смыслу переменная действия соответствует площади, охваченной траекторией в фазовом пространстве и измеряемой в единицах $\hbar$. В соответствии с правилом квазиклассического квантования разрешены лишь те значения энергии E_n, которым соответствует действие, равное $2\pi(n + 1/2)$, где $n = 0, 1, 2, \ldots$ Таким образом, используя (16.27) и замечая, что траектория симметрична относитель-

но прямой $x = 0$, получаем

$$S(E_n) = 2\left[\frac{2m}{\hbar^2}\right]^{1/2} \int\limits_{r_{\text{left}}}^{r_{\text{right}}} [E_n - V(r)]^{1/2} dr = 2\pi\left(n + \frac{1}{2}\right). \tag{16.29}$$

В граничных точках интервала интегрирования подынтегральное в (16.29) равно нулю. Для удобства решения данной задачи введем следующие безразмерные переменные:

$$\begin{aligned} \varepsilon &= \frac{E}{V_0}, \\ \tilde{x} &= \frac{r}{\sigma}, \\ \gamma &= \left[\frac{2ma^2V_0}{\hbar^2}\right], \end{aligned} \tag{16.30}$$

используя которые выражение (16.29) можно записать в следующем виде:

$$S(\varepsilon_n) = \gamma \int\limits_{x_{\text{left}}}^{x_{\text{right}}} [\varepsilon_n - v(\tilde{x})]^{1/2} d\tilde{x} = 2\pi\left(n + \frac{1}{2}\right), \tag{16.31}$$

где

$$v(\tilde{x}) = 4\left[\frac{1}{\tilde{x}^{12}} - \frac{1}{\tilde{x}^6}\right].$$

Из (16.31) видно, что величина γ является мерой квантового характера движения в данной задаче, так как в классическом пределе $\hbar \to 0$, $m \to \infty$ становится большим. Для молекул H_2, HD, O_2 величина γ составляет 21,7, 24,8, 150 соответственно.

Сравнив (16.31) и (16.28), видим, что выражение $\tilde{k}(\tilde{x}) = \gamma \cdot \sqrt{\varepsilon_n - v(\tilde{x})}$ является обезразмеренным волновым числом волны де Бройля.

В соответствии с вышеизложенным решение рассматриваемой задачи разбивается на следующие этапы:

1. Нахождение зависимости $S(\varepsilon)$ и спектра ε_n колебаний двухатомной молекулы в соответствии с (16.31).
2. Построение спектра совместно с зависимостью $V(\tilde{x})$.
3. Решение уравнений Гамильтона и построение траектории на фазовой плоскости.

Будем находить разрешенные значения энергетического спектра ε_n в соответствии со следующим алгоритмом:

1. Найти по таблице значений функции $S(\varepsilon)$ для заданного квантового числа n интервал, на котором выполняется условие

$$S_{j-1} \leqslant 2\pi\left(n + \frac{1}{2}\right) \leqslant S_j.$$

2. Если $S_{j-1} = 2\pi\left(n + \frac{1}{2}\right)$ принять в качестве значения n-го энергетического спектра E_{j-1}.

3. Если $S_j = 2\pi\left(n + \frac{1}{2}\right)$ принять в качестве значения n-го энергетического спектра E_j.
4. Если $S_{j-1} < 2\pi\left(n + \frac{1}{2}\right) < S_j$ принять в качестве n-го значения энергетического спектра, значение, получаемое с помощью линейной интерполяции зависимости $S(\varepsilon)$ на отрезке $[E_{j-1}, E_j]$:

$$E_n := \frac{\left(n + \frac{1}{2}\right)\pi - b}{a},$$

где

$$a = \frac{S_j - S_{j-1}}{E_j - E_{j-1}},$$

$$b = \frac{S_{j-1}E_j - S_jE_{j-1}}{E_j - E_{j-1}}.$$

5. Увеличить квантовое число на единицу $n = n + 1$.
6. Если

$$2\pi\left(n + \frac{1}{2}\right) \leqslant \max\left(S\left(E\right)\right),$$

повторение п. 1–6.

7. Если

$$2\pi\left(n + \frac{1}{2}\right) > \max\left(S\left(E\right)\right),$$

завершение работы.

Геометрическая интерпретация описанного алгоритма представлена на рис. 16.5.

Рис. 16.5. Геометрическая интерпретация алгоритма вычисления энергетического спектра

Для реализации описанного выше алгоритма вычисления спектра колебаний двухатомной молекулы в MATLAB необходимо создать два m-файла:

1) файл `Action.m`, содержащий описание функции, возвращающей значения функции для заданных значений энергии;

2) файл `Spectr.m`, содержащий описание функции, возвращающей собственные значения энергетического спектра колебаний двухатомной молекулы.

```
% листинг файла Action.m
function z = Action(Emin,Emax,Xmin,gamma)
% функция, возвращающая значения функции
% для заданных значений энергии;
% Emin - минимальное значение энергии
% Emax - максимальное значение энергии
% Xmin - координата минимума
% потенциальной энергии
% gamma - коэффициент, входящий
% в обезразмеренное уравнение
% Шредингера (16.23)
% нахождение корней уравнения U(x)=Emax

s1='4*(1./r.^12-1./r.^6)';
s2=num2str(Emax);
s=strcat(s1,'-',s2);
x0=0.1;
% левая точка поворота
Xleft=fzero(inline(s,'r'),[0.1 Xmin]);
x0=10*Xmin;
% правая точка остановки
Xrihgt=fzero(inline(s,'r'),[Xmin x0]);
Emin=Emin+10^-3;
Ne=100; % число значений зависимости  (S(E))
De=(Emax-Emin)/(Ne-1);
Np=1000; % число узлов сетки, используемых
         % для вычисления интеграла
j=1:Np;
for i=1:Ne
     E=Emin+De*(i-1);
     x0=0.1;
     s2=num2str(E);
     s='';
     s=strcat(s1,'-',s2);
     % левая точка остановки,
     % соответствующая энергии E
     Xleft=fzero(inline(s,'r'),[x0 Xmin]);
     x0=10*Xmin;
% правая точка остановки,
% соответствующая энергии E
     Xright=fzero(inline(s,'r'),[Xmin x0]);
     z(i,1)=E;
     x(j)=Xleft+(Xright-Xleft)/(Np-1)*(j-1);
```

```
    s=strcat('(',s2,'-',s1,')','.^0.5');
    F=inline(s,'r');
    S(j)=gamma*real(F(x(j)));
    I=trapz(x,S); % вычисление интеграла (16.31)
    z(i,2)=I;
end
```

Обращаем внимание читателя на строчку программы

```
S(j)=gamma*real(F(x(j)));
```

в файле `Action.m`, в которой в вектор S заносится значение действительной части подынтегральной функции в (16.31). Необходимость данной операции обусловлена тем, что для нахождения координат точек остановки мы используем встроенную в MATLAB функцию `fzero`, возвращающую приближенное значения корня нелинейного уравнения. Так как любое численное решение имеет погрешность, может оказаться, что значение соответствующего корня окажется левее истинного значения левой точки поворота и/или правее правой точки поворота, а потому значение подынтегральной функции и, соответственно, интеграла (16.31) становятся мнимыми.

```
% листинг программы Spectr.m
function [Xleft,Xright,EL] = Spectr(z,Xmin)
% функция, возвращающая собственные
% значения энергетического спектра
% колебаний двухатомной молекулы (EL),
% и соответствующие данным значениям
% координаты точек поворота (Xleft,Xright)
% z - матрица, возвращенная функцией Action
% Xmin - координата минимума
% потенциальной энергии

n=0;
M1=max(z(:,2));
j=1;
s1='4*(1./r.^12-1./r.^6)';
while and((n+0.5)*pi<M1,j+1<length(z))
    while z(j,2)<(n+0.5)*pi
        j=j+1;
    end;
    if j+1<length(z);
      x=[z(j-1,1) z(j,1)];
      y=[z(j-1,2) z(j,2)];
      a=(y(2)-y(1))/(x(2)-x(1));
      b=y(1)-a*x(1);
      EL(n+1)=((n+0.5)*pi-b)/a;
      x0=0.1;
      s2=num2str(EL(n+1));
```

```
      s='';
      s=strcat(s1,'-',s2);
      Xleft(n+1)=fzero(inline(s,'r'),[x0 Xmin]);
      x0=10*Xmin;
      Xright(n+1)=fzero(inline(s,'r'),[Xmin x0]);
      n=n+1;
    end;
end;
```

Далее необходимо выполнить следующую последовательность команд, сохраненную нами в файле `Glava16_2.m`:

```
% листинг файла Glava16_2.m
Xmin=2^(1/6); % координата минимума потенциала
gamma=80; % коэффициент, входящий
          % в обезразмеренное уравнение
          % Шредингера (16.23)
Emin=-1; % минимальное значение энергии
Emax=-10^-4; % максимальное значение энергии

% вычисление зависимости S(E)
z = Action(Emin,Emax,Xmin,gamma);

% вычисление энергетического
% спектра колебаний двухатомной
% молекулы
[Xleft,Xright,EL] = Spectr(z,Xmin);

EL % вывод значений энергетического спектра

% визуализация зависимости S(E)
figure(1);
plot(z(:,1),z(:,2))
```

Ниже представлены результаты выполнения описанной последовательности команд.

1. Сообщения, выводимые в командное окно:

```
EL =
  Columns 1 through 7
   -0.9346   -0.8120   -0.7002   -0.5987
             -0.5071   -0.4249   -0.3519
  Columns 8 through 14
   -0.2875   -0.2312   -0.1825   -0.1411
             -0.1064   -0.0778   -0.0548
  Columns 15 through 16
   -0.0368   -0.0233
```

2. Зависимость действия от энергии (рис. 16.6).

Рис. 16.6. Зависимость действия от энергии

Рассмотрим задачу отображения потенциала Ленарда-Джонса и энергетического спектра колебаний двухатомной молекулы на одном чертеже. Данный чертеж представляет собой график функции $V = V(x)$ и набор отрезков прямых $y = E_n$, параллельных оси абсцисс, координаты левых и правых концов которых есть решения уравнения $V(r) = E$. К сожалению, графические средства MATLAB не позволяют автоматически отобразить на одном чертеже зависимость $V = V(x)$ и энергетический спектр колебаний двухатомной молекулы, поскольку функции, зависящие от одной переменной, строятся линиями, последовательно соединяющими каждую пару точек. Для выхода из данной ситуации можно использовать следующий подход, основную идею которого иллюстрирует рис. 16.7.

Рис. 16.7. К объяснению метода формирования матрицы для построения на одном чертеже энергетического спектра колебаний двухатомной молекулы и зависимости $v = v(x)$

Основная идея метода основана на формировании матрицы, в одном столбце которой в определенном порядке расположены значения потенциала $V(x)$ и энергетического спектра ε_n, в другом — соответствующие значения переменной x и координаты точек остановки. Причем каждое значение спектра в первый столбец матрицы заносится три раза подряд, при этом в соответствующие строки нулевого столбца матрицы

последовательно заносятся значения x_{left}, x_{right}, x_{left}. Очевидно, что при последовательном соединении точек линия через данные точки будет проводиться дважды: от точки с координатами $(x_{\text{left}}, \varepsilon_n)$ к точке с координатами $(x_{\text{right}}, \varepsilon_n)$, и наоборот. Так как происходит возвращение в начальную точку, то дальнейшее построение графика начнется с той же самой точки, и искажений не возникнет (см. рис. 16.7).

Другая проблема, связанная с формированием указанной матрицы, состоит в необходимости упорядоченного размещения в массиве значений зависимости $V = V(x)$ и энергетического спектра ε_n. Для автоматизации процедуры упорядочивания можно использовать тот факт, что при $x < x_{\min}$ функция $V = V(x)$ является монотонно убывающей. Следовательно, нужно создать матрицу, которая содержит в нулевом столбце значения аргумента на интервале $a, x_{\min}$, в первом — соответствующие значения функции $V = V(x)$. Дополнить ее координатами точек остановки и значениями энергетического спектра ε_n в соответствии с описанной выше процедурой, затем провести упорядочивание матрицы в порядке убывания значений энергии. После этого дополнить упорядоченную матрицу значениями координат x из отрезка $[x_{\min}, b]$ и соответствующими значениями функции $V = v(x)$. Далее отобразить зависимость, сохраненную в описанной матрице, на графике. (Здесь $[a, b]$ — выбранный интервал для построения графика функции $V = V(x)$.)

Для реализации алгоритма создадим m-файл `Potential_and_Spectr.m`, содержащий описание функции, возвращающей искомую матрицу.

```
% листинг файла Potential_and_Spectr.m
function z = Potential_and_Spectr(EL,Xleft,Xright,x1,x2)
% функция, возвращающая матрицу,
% используемую для построения на
% одном чертеже графика потенциала
% и значений энергетического спектра
% EL - вектор, содержащий значения
% энергетического спектра
% Xleft - вектор,
% содержащий координаты
% левых точек поворота
% Xright - вектор,
% содержащий координаты
% правых точек поворота
% x1 - левая граница интервала
% построения графика
% x2 - правая граница интервала
% построения графика

% задаем координатную сетку
N0=100;
Dx=(x2-x1)/(N0-1);
i=1:N0;
x0=x1+Dx*(i-1);
```

```
% вычисляем занчения потенциала
% в узлах координантной сетки
y0=V(x0);

% получаем минимальное значение потенциала
% и соответствующий ему номер элемента
% вектора y0
[Ymin N]=min(y0);

% заносим в векторы X0,Y0
% значения координат из интервала [x1,x0(N)]
% и соответствующие значения потенциала
for i=1:N
  X0(i)=x0(i);
  Y0(i)=y0(i);
end;

% формироваение матрицы в соответствии с
% алгоритмом
K=length(EL);
m=1;
for i=N:N+K-1
  X0(i+m)=Xleft(i-N+1);
  Y0(i+m)=EL(i-N+1);
  m=m+1;
  X0(i+m)=Xright(i-N+1);
  Y0(i+m)=EL(i-N+1);
  m=m+1;
  X0(i+m)=Xleft(i-N+1);
  Y0(i+m)=EL(i-N+1);
end;
N1=length(Y0);
for i=1:N1+1
  Sy=-10^309;
  for j=1:N1
    if Sy<Y0(j)
      Sy=Y0(j);
      N2=j;
    end;
  end;
  z(i,1)=X0(N2);
  z(i,2)=Y0(N2);
  Y0(N2)=-10^309;
end;
Xmax=max(X0);
```

```
i=1;
Xt=x0(N);
while Xt<Xmax
  z(i+N1,1)=Xt;
  z(i+N1,2)=V(Xt);
  i=i+1;
  Xt=Xt+Dx;
end;


function z = V(x)
% функция, возвращающая значение
% потенциала Ленарда-Джонса
z=4*(1./x.^12-1./x.^6);
```

Далее необходимо выполнить следующую последовательность команд, сохраненную нами в файле `Glava16_3.m`:

```
% листинг файла Glava16_3.m
Xmin=2^(1/6); % координата минимума потенциала
gamma=80; % коэффициент, входящий
          % в обезразмеренное уравнение
          % Шредингера (16.23)
Emin=-1; % минимальное значение энергии
Emax=-10^-4; % максимальное значение энергии

% вычисление зависимости S(E)
z = Action(Emin,Emax,Xmin,gamma);

% вычисление энергетического
% спектра колебаний двухатомной
% молекулы
[Xleft,Xright,EL] = Spectr(z,Xmin);

x1=0.95;
x2=2.4;

% формирование матрицы, используемой для
% построения на одном чертеже
% графика потенциала и значений
% энергетического спектра
Z=Potential_and_Spectr(EL,Xleft,Xright,x1,x2);
% визуализация потенциала взаимодействия
% ядер двухатомной молекулы
% и энергетического спектра колебаний
plot(Z(:,1),Z(:,2),'k')
```

Результат выполнения описанной последовательности команд представлен на рис. 16.8.

Рис. 16.8. Зависимость потенциала взаимодействия ядер от расстояния между ними и энергетический спектр колебаний двухатомной молекулы

При построении фазовой траектории колебаний двухатомной молекулы, задаваемой (16.27), необходимо учитывать, что данная траектория является двузначной функцией, поэтому ее построение требует выполнения предварительных действий, которые достаточно очевидны. Для построения траектории движения на фазовой плоскости сначала необходимо сформировать матрицу, в верхней половине которой находятся строки, содержащие значения переменной $[x_{\text{left}}, x_{\text{right}}]$ и соответствующие им значения импульса $p(x)$, в нижней половине матрицы содержатся значения координаты x, изменяющиеся от x_{right} до x_{left}, и соответствующие им значения импульса $-p(x)$, а затем отобразить искомую зависимость.

Отметим, что поведение рассматриваемой нами модели определяется единственным параметром γ, являющимся комбинацией физических параметров системы: характерного размера системы a, массы системы m и глубины потенциала V_0, величину которого мы задавали при нахождении энергетического спектра колебаний двухатомной молекулы. В этой ситуации для описания особенностей движения молекулы в фазовом пространстве удобнее использовать не импульс, пропорциональный массе (значение которой мы не задавали), но безразмерный волновой вектор волны де Бройля, который, с одной стороны, прямо пропорционален импульсу, и зависит только от γ — с другой.

Для реализации данного алгоритма мы создали m-файл `PTr.m`, содержащий описание функции, возвращающей координаты точек фазовой траектории для каждого собственного значения энергии.

```
% листинг файла PTr.m
function z = PTr(EL,Xleft,Xright,gamma,Np)
% функция, возвращающая
% координаты точек фазовой траектории для
% каждого собственного значения энергии
N=length(EL);
```

```
k=1:Np;
for i=1:N
  dx=(Xright(i)-Xleft(i))/(Np-1);
  x(k)=Xleft(i)+dx*(k-1);
  for j=1:Np
    Z(j,1)=x(j);
    Z(j,2)=real(p(x(j),EL(i),gamma));
  end;
  for j=Np+1:2*Np
    Z(j,1)=x(2*Np-j+1);
    Z(j,2)=-Z(2*Np-j+1,2);
  end;
  Z(2*Np+1,1)=x(1);
  Z(2*Np+1,2)=Z(1,2);
  if i==1
    z=Z;
  else
    z=cat(3,z,Z);
  end;
end;

function z=p(x,E,gamma)
% функция, возвращающая значения импульса
z=gamma*(E-V(x)).^0.5;

function z=V(x)
% функция, возвращающая значения
% потенциала Ленарда-Джонса
z=4*(1./x.^12-1./x.^6);
```

Далее необходимо выполнить следующую последовательность команд, сохраненную нами в файле `GLava16_4.m`:

```
Xmin=2^(1/6); % координата минимума потенциала
gamma=80; % коэффициент, входящий в
          % обезразмеренное уравнение
          % Шредингера (16.22)
Emin=-1; % минимальное значение энергии
Emax=-10^-4; % максимальное значение энергии

% вычисление зависимости S(E)
z = Action(Emin,Emax,Xmin,gamma);

% вычисление энергетического
% спектра колебаний двухатомной
% молекулы
```

```
[Xleft,Xright,EL] = Spectr(z,Xmin);

Np=2000; % число точек одной фазовой траектории

% вычисление координат точек
% фазовых траекторий для
% каждого значения
% энергетического спектра
Ph=PTr(EL,Xleft,Xright,gamma,Np);

% визуализация фазовых траекторий
plot(Ph(:,1,1),Ph(:,2,1),'k')
hold on
for k=2:length(EL)
        plot(Ph(:,1,k),Ph(:,2,k),'k');
end;
hold off
```

Результат выполнения описанной последовательности команд представлен на рис. 16.9.

Рис. 16.9. Фазовые траектории колебаний двухатомной молекулы, соответствующие первым шестнадцати уровням энергетического спектра

Другой метод построения фазовой траектории основан на использовании зависимостей $r = r(t)$, $p = p(t)$, которые являются решением системы уравнений Гамильтона:

$$\begin{aligned} r' &= \frac{p\,(r)}{m}, \\ p' &= -\frac{\partial V\,(r)}{\partial\, r}. \end{aligned} \tag{16.32}$$

Продифференцировав по времени первое уравнение в (16.32) и подставив в него второе уравнение, для переменой r получаем уравнение второго порядка:

$$r'' = -\frac{1}{m}\frac{\partial V\,(r)}{\partial\, r}. \tag{16.33}$$

Для выбора начальных условий уравнения (16.33) будем считать, что в момент времени $t = 0$ частица находится в одной из точек остановки, для определенности — в левой, то есть $r(0) = x_{\text{left}}$, $r'(0) = 0$. Для проведения численного решения уравнения (16.33) целесообразно предварительно записать систему в безразмерных переменных:

$$\tilde{x} = \frac{r}{\sigma},$$

$$\tilde{t} = \frac{t}{T},$$

$$\tilde{E} = \frac{E}{V_0},$$

где σ — параметр потенциала, характеризующий его ширину (см. (16.25)), T — характерное время движения.

Потенциал Ленарда-Джонса и уравнение движения в выбранных безразмерных переменных имеют следующий вид:

$$\tilde{V}(\tilde{x}) = 4\left(\frac{1}{\tilde{x}^{12}} - \frac{1}{\tilde{x}^{6}}\right), \tag{16.34}$$

$$\tilde{x}'' = \frac{24V_0T^2}{m\sigma^2}\left(\frac{2}{\tilde{x}^{13}} - \frac{1}{\tilde{x}^{7}}\right). \tag{16.35}$$

Для выбора переменной T найдем разложение потенциала в ряд Тейлора вблизи минимума потенциала ($r_{\min} = 2^{1/6}\sigma$):

$$V(\Delta r) \approx -V_0 + \frac{18 \cdot 2^{\frac{2}{3}}}{\sigma^2} V_0 \Delta r. \tag{16.36}$$

Сравнив (16.36) с выражением для потенциальной энергии линейного гармонического осциллятора

$$U(x) = \frac{1}{2}\chi\, x^2, \tag{16.37}$$

приходим к выводу о том, что вблизи минимума потенциал Ленарда-Джонса ведет себя подобно потенциалу гармонического осциллятора, у которого

$$\chi = \frac{2 \cdot 18 \cdot 2^{\frac{2}{3}}}{\sigma^2} V_0.$$

Следовательно, в качестве характерного времени T можно выбрать период колебаний гармонического осциллятора, в данном случае равный

$$T = 2\pi\sqrt{\frac{m}{\chi}} = 2\pi\frac{2^{-\frac{1}{3}}}{6}\sqrt{\frac{m\sigma^2}{V_0}}. \tag{16.38}$$

Подставляя (16.38) в (16.35), окончательно получаем

$$\tilde{x}'' = \frac{2^{\frac{7}{3}} \cdot \pi^2}{3}\left(\frac{2}{\tilde{x}^{13}} - \frac{1}{\tilde{x}^{7}}\right). \tag{16.39}$$

Для нахождения численных решений (16.39) необходимо создать m-файл `Acc2.m`, содержащий описание функции, возвращающей значения первых производных.

```
% листинг файла Acc2.m
function z = Acc2(t,x);
% функция, возвращающая значения
% первых производных ДУ (16.39)

z=zeros(2,1);
z(1)=x(2);
z(2)=2^(7/3)*pi^2/3*(2./x(1).^13-1./x(1).^7);
```

Далее необходимо выполнить следующую последовательность команд, сохраненную нами в файле `Glava16_5.m`:

```
% листинг файла Glava16_5.m
Xmin=2^(1/6); % координата минимума
              % потенциала
gamma=80; % коэффициент, входящий
          % в обезразмеренное уравнение
          % Шредингера (16.23)
Emin=-1; % минимальное значение энергии
Emax=-10^-4; % максимальное значение энергии

% вычисление зависимости S(E)
z = Action(Emin,Emax,Xmin,gamma);

% вычисление
% координат точек остановки и
% энергетического спектра колебаний
% двухатомной молекулы
[Xleft,Xright,EL] = Spectr(z,Xmin);

N=5; % номер выбранного уровня

% вычисление численного
% решения ДУ (16.39)
R0=[Xleft(5) 0]; % вектор начальных условий
Np=5000; % число узлов временной сетки
Tfin=1.8; %  правая граница
          %  временного интервала
[T,M]=ode45('Acc2',[0:Tfin/Np:Tfin],R0);

% визуализация зависимостей x = x(t), v = v(t)
% и фазовой траектории
figure(1);
plot(T,M(:,1), 'k')

figure(2);
```

```
plot(T,M(:,2), 'k')

figure(3);
plot(M(:,1),M(:,2), 'k')
```

Результаты выполнения описанной последовательности команд представлены на рис. 16.10–16.12.

Рис. 16.10. Зависимость расстояния между ядрами двухатомной молекулы, находящейся на пятом энергетическом уровне, от времени

Рис. 16.11. Зависимость скорости движения ядер двухатомной молекулы, находящейся на пятом энергетическом уровне, от времени

Рис. 16.12. Фазовая траектория колебаний двухатомной молеклы, находящейся на пятом энергетическом уровне

Задача 16.9

Вычислите энергетический спектр колебаний двухатомной молекулы, используя метод, описанный в разделе 16.2. Сравните результаты, полученные в квазиклассическом приближении, с результатами, полученными численным интегрированием уравнения Шредингера. Постройте зависимость рассто-

яниями между значениями энергетического спектра от номера квантового числа.

Задача 16.10

Используя квазиклассическое приближение, определите собственные значения энергии частицы, находящейся в прямоугольной яме с бесконечно высокими прямоугольными стенками. Сравните полученные значения энергии с соответствующими значениями, найденными аналитическими методами. Зависит ли точность квазиклассического приближения для рассматриваемого потенциала от номера квантового числа?

Задача 16.11

Используя квазиклассическое приближение, найдите собственные значения энергии для потенциалов, рассмотренных в задачах 16.5—16.7. Сравните соответствующие значения энергий, полученные прямым численным интегрированием стационарного уравнения Шредингера, с результатами, полученными в квазиклассическом приближении. Зависит ли разность между значениями энергии, полученными численным интегрированием (16.10), и соответствующими значениями энергии, полученными в квазиклассическом приближении от номера квантового числа? Объясните полученный результат.

Задача 16.12

Поле представляет собой две одинаковые потенциальные ямы, разделенные барьером (рис. 16.13). При непроницаемом барьере существуют уровни энергии, отвечающие движению частицы только в одной из ям. В данном случае вследствие симметрии ям будут существовать одинаковые значения энергии, соответствующие двум различным состояниям (когда частица находится в правой или левой ямах), которые называются вырожденными. Возможность прохождения частицы через потенциальный барьер приводит к снятию вырождения и расщеплению каждого из уровней на два близких уровня, соответствующих состоянию, когда частица может одновременно находиться в обеих ямах. Выберите для описания потенциал подходящую функцию и определите величину расщепления уровней, используя квазиклассическое приближение и метод численного интегрирования уравнения Шредингера, описанный в разделе 16.2

Рис. 16.13. Потенциал в задаче 16.12

16.4. НЕСТАЦИОНАРНОЕ УРАВНЕНИЕ ШРЕДИНГЕРА

В общем случае для описания эволюции квантовой системы во времени требуется найти решение $\psi(\vec{r}, t)$ нестационарного уравнения Шредингера (16.3). С математической точки зрения данная задача относится к краевым задачам дифференциальных уравнений в частных производных, в которых требуется найти решение уравнения $\psi(\vec{r}, t)$, если известны значения данной функции в момент времени $t = 0$ — $\psi(\vec{r}, 0)$. Рассмотрим численные методы решения нестационарного уравнения Шредингера на примере одномерного уравнения (16.6).

Введем в рассмотрение пространственную сетку с числом узлов N и шагом по пространству $h = 1/N$, на которой вместо непрерывной производной будем использовать ее конечно-разностную аппроксимацию:

$$\left.\frac{\partial^2 \psi}{\partial x^2}\right|_{t=t_n} = \frac{\psi_{j+1}^n - 2\psi_j^n + \psi_{j-1}^n}{h^2}. \tag{16.40}$$

Производную по времени аппроксимируем простейшей разностной формулой первого порядка точности, считая дискретный шаг изменения времени равным Δt:

$$\left.\frac{\partial \psi}{\partial t}\right|_{x=x_n} = \frac{\psi_i^{n+1} - \psi_i^n}{\Delta t}. \tag{16.41}$$

Подставляя (16.40), (16.41) в (16.6), получаем

$$i\hbar \frac{\psi_j^{n+1} - \psi_j^n}{\Delta t} = -\frac{\hbar^2}{2m} \frac{\psi_{j+1}^n - 2\psi_j^n + \psi_{j-1}^n}{h^2} + U_j, \tag{16.42}$$

здесь i — комплексная единица.

При $j = 1$ и $j = N - 1$ в уравнении (16.42) используют значения функции на границе области, в которой ищется решение уравнения, — ψ_0 и ψ_N . Из (16.42) можно выразить ψ_j^{n+1}:

$$\psi_j^{n+1} = \psi_j^n + i\left(-\frac{\psi_{j+1}^n - 2\psi_j^n + \psi_{j-1}^n}{2h^2} + U_j\right)\Delta t, \tag{16.43}$$

здесь и далее в данном разделе мы полагаем $\hbar \equiv 1$, $m \equiv 1$.

Для более компактной записи (16.43) введем следующие обозначения

$$-\frac{\psi_{j+1}^n - 2\psi_j^n + \psi_{j-1}^n}{2h^2} + U_j = \left(\widehat{D}\psi_j^n\right). \tag{16.44}$$

Используя (16.44), запишем (16.43) в виде

$$\psi_j^{n+1} = \left(1 - i\Delta t \widehat{D}\right)\psi_j^n. \tag{16.45}$$

Разностная схема (16.45) называется явной, так как она позволяет по известным значениям функции ψ в момент времени t_n вычислить значения функции ψ в момент

времени t_{n+1}. Однако данная схема, в которой эволюция решения во времени определяется оператором $\left(1 - i\Delta t\widehat{D}\right)$, оказывается неустойчивой при любом шаге Δt. Так, все собственные значения данного оператора

$$1 - i\varepsilon_\lambda \Delta t,$$

равные по абсолютной величине

$$\left[1 + (\varepsilon_\lambda \Delta t)^2\right]^{1/2},$$

оказываются больше единицы.

Один из способов, позволяющий сделать схему устойчивой, состоит в использовании пространственной производной в момент времени t_{n+1}:

$$i\frac{\psi_j^{n+1} - \psi_j^n}{\Delta t} = \frac{\psi_{j+1}^{n+1} - 2\psi_j^{n+1} + \psi_{j-1}^{n+1}}{h^2}. \tag{16.46}$$

В выбранных обозначениях данная схема имеет следующий вид:

$$\frac{\psi_j^{n+1} - \psi_j^n}{\Delta t} = -i\widehat{D}\psi_j^{n+1}. \tag{16.47}$$

Так как неизвестные значения входят в обе части уравнения (16.47), данная вычислительная схема называется неявной. Уравнение (16.47) можно разрешить относительно ψ_j^{n+1}:

$$\psi_j^{n+1} = \frac{1}{1 + i\Delta t\widehat{D}}\psi_j^n. \tag{16.48}$$

Так как абсолютная величина собственных значений оператора перехода, стоящего в правой части (16.48)

$$\left(1 + \varepsilon_\lambda^2 \Delta t^2\right)^{-\frac{1}{2}},$$

всегда меньше единицы, вычислительная схема оказывается устойчивой при любых значениях Δt.

Однако данная схема, так же как и явная, не может быть использована для решения уравнения Шредингера (16.6), так как получаемое численное решение не является унитарным — модуль волновой функции оказывается монотонно убывающим во времени.

Свободной от указанных недостатков оказывается следующая модификация неявной разностной схемы:

$$\psi_j^{n+1} = \left[\frac{1 - i\frac{1}{2}\Delta t\widehat{D}}{1 + i\frac{1}{2}\Delta t\widehat{D}}\right]\psi_j^n. \tag{16.49}$$

Оператор, стоящий в правой части (16.49), оказывается эрмитовым унитарным оператором, собственные значения которого равны единице, поэтому модуль волновой функции остается постоянным при переходе от одного момента времени к следующему. Кроме того, можно показать, что соотношение (16.49) аппроксимирует точный оператор перехода $e^{-i\Delta t\widehat{D}}$ с точностью до членов $O(h^2)$.

Для проведения практических расчетов оказывается более удобным переписать (16.49) в следующем эквивалентном виде:

$$\psi_j^{n+1} = \left(\frac{2}{1+i\frac{1}{2}\Delta t\widehat{D}} - 1\right)\psi_j^n \equiv \chi_j^n - \psi_j^n, \tag{16.50}$$

который устраняет один проход по узлам сетки, необходимый для вычисления числителя.

Для нахождения на каждом шаге по времени значений промежуточной функции χ, определяемой уравнением

$$\left(1 + i\frac{1}{2}\Delta t\widehat{D}\right)\chi_j^n = 2\psi_j^n, \tag{16.51}$$

перепишем данное уравнение в явном виде

$$-\frac{i\Delta t}{2h^2}\chi_{j+1}^n + \left[1 + \frac{i\Delta t}{h^2} + \frac{i\Delta t}{2}U_j\right]\chi_{j+1}^n - \frac{i\Delta t}{2h^2}\chi_{j-1}^n = 2\psi_j^n \tag{16.52}$$

и, разделив обе части (16.52) на $-i\Delta t/2h^2$, окончательно получим

$$\chi_{j+1}^n + \left[-2 + \frac{2ih^2}{\Delta t} - h^2U_j\right]\chi_j^n + \chi_{j-1}^n = \frac{4ih^2}{\Delta t}\psi_j^n. \tag{16.53}$$

Таким образом, на каждом шаге по времени для нахождения значений χ_{j+1}^n приходится решать систему линейных уравнений N-го порядка, которая оказывается трехдиагональной (в каждое уравнение системы входят только три рядом расположенных неизвестных: переменная, соответствующая диагональному коэффициенту, и переменные, стоящие слева и справа от диагонального элемента). Для систем линейных уравнений с трехдиагональной матрицей известен эффективный вычислительный алгоритм, называемый методом прогонки.

Проведем обсуждение данного алгоритма на самом общем случае системы линейных уравнений

$$A\vec{x} = \vec{b}, \tag{16.54}$$

где A — трехдиагональная матрица, $\vec{x}$ — вектор-столбец неизвестных, $\vec{b}$ — вектор-столбец свободных членов.

Запишем данную систему в виде, аналогичном (16.53):

$$A_j^- x_{j-1} + A_j^0 x_j + A_j^+ x_{j+1} = b_j, \tag{16.55}$$

здесь A_j^-, A_j^0, A_j^+ обозначены отличные от нуля элементы матрицы A.

В частности, для системы линейных уравнений (16.53)

$$A_j^- = A_j^+ = 1,$$

$$A_j^0 = -2 + \frac{2ih^2}{\Delta t} - h^2 U_j.$$

Для решения системы уравнений (16.55) предположим, что искомое решение удовлетворяет рекуррентному соотношению, связывающему значения x_j в соседних узлах j и $j+1$:

$$x_{j+1} = \alpha_j x_j + \beta_j, \tag{16.56}$$

где α_j, β_j — неизвестные коэффициенты, которые необходимо определить.

Подставляя (16.56) в (16.55), получаем

$$A_j^- x_{j-1} + A_j^0 x_j + A_j^+ \left(\alpha_j x_j + \beta_j\right) = b_j. \tag{16.57}$$

Разрешив (16.57) относительно x_j, получаем

$$x_j = \gamma_j A_j^- x_{j-1} + \gamma_j \left(A_j^+ \beta_j - b_j\right), \tag{16.58}$$

где

$$\gamma_j = -\frac{1}{A_j^0 + A_j^+ \alpha_j}. \tag{16.59}$$

Сравнивая (16.56) и (16.58), видим, что они тождественны, если коэффициенты удовлетворяют следующим рекуррентным соотношениям:

$$\alpha_{j-1} = \gamma_j A_j^-, \tag{16.60}$$

$$\beta_{j-1} = \gamma_j \left(A_j^+ \beta_j - b_j\right). \tag{16.61}$$

Таким образом, решение трехдиагональной системы уравнений осуществляется следующей последовательностью действий:

1. Вычисление коэффициентов γ_j, α_j, β_j по формулам (16.59)–(16.61), в которых $j = N-2, N-3, \ldots, 0$, $\beta_{N-1} = x_N$ (обратный ход).

2. Вычисление x_j по формуле (16.56), в которой $j = 0, 1, \ldots, N-1$, где x_0 известно из начальных условий (прямой ход).

Описанный алгоритм решения системы линейных уравнений, состоящий из обратного и прямого ходов, требует $2N$ арифметических операций, в отличие от прямых методов (например, умножение на обратную матрицу), требующий порядка N^2 операций.

Для реализации описанного алгоритма в MATLAB создадим четыре m-файла:

1) файл `Barrier.m`, содержащий описание функции, возвращающей значения потенциального барьера в узлах пространственной сетки;

2) файл `Package.m`, содержащий описание функции, возвращающей значения действительной и мнимой частей волнового пакета;

3) файл `Coeff.m`, содержащий описание функции, возвращающей значения коэффициентов, используемых в методе прогонки;

4) файл `Psi.m`, содержащий описание функции, возвращающей значения волновой функции в момент времени $t + \Delta t$.

```
% листинг файла Barrier.m
function z = Barrier(x,v0,x0,A,Mode);
% функция, возвращающая значения
% потенциального барьера в заданном
% узле координатной сетки
% x - координата узла
% v0 - высота барьера
% x0 - координата, определяющая
% положение максимума огибающей пакета
% A - ширина барьера
% Mode - переменная, определяющая тип барьера

if Mode==1
    % прямоугольный барьер
    z=Rect(x,v0,x0,A);
end;

if Mode==2
   % барьер, описываемый кривой Гаусса
    z=Gauss(x,v0,x0,A);
end;

if Mode==3
   % ступенчатый барьер
   z=StepB(x,v0,x0,A);
end;

if Mode==4
    % параболический барьер
    z=Parabolic(x,v0,x0,A);
end;

function z = Rect(x,v0,x0,A)
% функция, возвращающая значения
% прямоугольного барьера
if and(x0-A/2<=x,x<=x0+A/2)
   z=v0;
else
   z=0;
end;
```

```
function z = Gauss(x,v0,x0,A)
% функция, возвращающая значения
% потенциального барьера,
% описываемого Гауссовой кривой
z=v0*exp(-(x-x0).^2*log10(20)/A.^2);

function z = StepB(x,v0,x0,A)
% функция, описывающая ступенчатый барьер
z=(v0*atan((x-x0)*2*6.313/A*2/pi)+1)/2;

function z = Parabolic(x,v0,x0,A)
% функция, описывающая
% параболический барьер
if and(x0-A/2<=x,x<=x0+A/2)
   z=-v0*(2/A).^2*(x-x0).^2+v0;
else
   z=0;
end;

% листинг файла Package.m
function z = Package(x,k0,x0,sigma,Mode)
% функция, возвращающая значения
% действительной и мнимой частей
% волнового пакета
% x - пространственная координата
% k0 - волновой вектор пакета
% sigma - ширина пакета
% Mode - переменная, определяющая
% тип огибающей пакета:
% Mode=1 - гауссов пакет
% Mode=2 - релеевский пакет

Xmin=min(x);
Xmax=max(x);

if Mode==1
  z=exp(j*k0*x)*0.9.*exp(-(x-x0).^2*log10(2)/sigma.^2);
end;

if Mode==2
   z=exp(j*k0*x)*sigma.^2./(sigma.^2+(x-x0).^2);
end;
A=Norm(x,z);
z=z/A.^0.5;
```

```
function z = Norm(x,Z)
% функция, возвращающая значение
% нормировочного множителя
z1=abs(Z);
z1=z1.^2;
z=trapz(x,z1);

% листинг функции Coeff.m

function z=Coeff(dt,dx,V)
% функция, возвращающая значения
% коэффициентов в методе прогонки
% dt - шаг по времени
% dx - шаг пространственной сетки
% V - вектор, содержащий значения барьера

Npoints=length(V);
for k=1:Npoints
    Az(k)=-2+2*j*dx^2/dt-dx^2*V(k);
end;
alpha(Npoints-1)=0;
gamma(Npoints)=-1/Az(Npoints);
Ap=1;
k=Npoints;
while k>=2
    alpha(k-1)=gamma(k)*Ap;
    gamma(k-1)=-1/(Az(k-1)+Ap*alpha(k-1));
    k=k-1;
end;
z=gamma;

% листинг функции Psi.m

function z=Psi(Coeff,psi,dx,dt,x)
% функция, возвращающая значения
% волновой функции в момент времени
% Coeff - вектор, возвращенный функцией Coeff
% psi - вектор, содержащий значения
% волновой функции в момент
% времени t = 0;
% dx - шаг пространственной сетки
% dt - шаг по времени
% x - вектор, содержащий координаты узлов
% пространственной сетки

Npoints=length(psi);
```

```
% обратный ход
Ap=1;
beta(Npoints)=0+0*j;
k=Npoints-1;
while k>=2
    beta(k-1)=Coeff(k)*(Ap*beta(k)-4*j*dx^2/dt*psi(k));
    k=k-1;
end;
% прямой ход
xi(1)=0+0*j;
for k=1:Npoints-1
   xi(k+1)=Coeff(k)*xi(k)+beta(k);
   psi(k+1)=xi(k+1)-psi(k+1);
end;
fi=abs(psi).^2;
B=trapz(x,fi);
z=psi/B^0.5;
```

Далее необходимо выполнить следующую последовательность команд, сохраненную нами в файле `Glava16_6.m`:

```
% листинг файла Glava16_6.m
Xmin=-200; % левая граница
           % пространственной сетки
Xmax=200; % правая граница
          % пространственной сетки
Npoints=500; % число узлов
             % пространственной сетки
dt=0.5; % шаг по времени
% вычисление координат узлов
% пространственной координатной сетки
i=1:Npoints+1;
dx=(Xmax-Xmin)/Npoints;
x(i)=Xmin+dx*(i-1);

% вычисление значений волновой функции
% в момент времени t = 0
k0=0.5;
x0=-100;
sigma=2;
z = Package(x,k0,x0,sigma,2);

% вычисление значений потенциального барьера
% в узлах координатной сетки
v0=1.5;
x0=0;
```

```
A=60;
Mode=4;
for i=1:Npoints+1
   V(i)=Barrier(x(i),v0,x0,A,Mode);
end;

% вычисление значений волновой функции
% в последовательные моменты времени
C=Coeff(dt,dx,V);
for i=1:300
        if i==1
           Z=Psi(C,z,dx,dt,x);
           z1=Z;
        else
           Z=Psi(C,Z,dx,dt,x);
           z1=cat(3,z1,Z);
        end;
     end;

% визуализация решений уравнения
% Шредингера
figure(1);
plot(x,V,'k',x,abs(z1(1,:,1)),'k');

figure(2);
plot(x,V,'k',x,abs(z1(1,:,10)),'k');

figure(3);
plot(x,V,'k',x,abs(z1(1,:,20)),'k');

figure(4);
plot(x,V,'k',x,abs(z1(1,:,50)),'k');

figure(5);
plot(x,V,'k',x,abs(z1(1,:,75)),'k');

figure(6);
plot(x,V,'k',x,abs(z1(1,:,100)),'k');

figure(7);
plot(x,V,'k',x,abs(z1(1,:,125)),'k');

figure(8);
plot(x,V,'k',x,abs(z1(1,:,150)),'k');
```

```
figure(9);
plot(x,V,'k',x,abs(z1(1,:,175)),'k');

figure(10);
plot(x,V,'k',x,abs(z1(1,:,200)),'k');

figure(11);
plot(x,V,'k',x,abs(z1(1,:,225)),'k');

figure(12);
plot(x,V,'k',x,abs(z1(1,:,250)),'k');

figure(13);
plot(x,V,'k',x,abs(z1(1,:,275)),'k');

figure(14);
plot(x,V,'k',x,abs(z1(1,:,300)),'k');
```

Результат выполнения описанной последовательности команд представлен на рис.16.14–16.27.

Рис. 16.14. Огибающая волнового пакета в момент времени $t = 0$

Рис. 16.15. Огибающая волнового пакета в момент времени $t = 10$

Для рассмотрения эволюции волнового пакета в динамике необходимо создать анимационный клип, выполнив следующую последовательность команд:

```
   >> for k=1:300 % запись в avi-файл
                  % 300 последовательных изображений
       h=plot(x,V,'k',x,abs(z1(1,:,k)),'k');
       set(h,'EraseMode','xor');
       F(k)=getframe;
    end;
>> movie(F,1); % воспроизведение анимационного клипа
```

Рис. 16.16. Огибающая волнового пакета в момент времени $t = 20$

Рис. 16.17. Огибающая волнового пакета в момент времени $t = 50$

Рис. 16.18. Огибающая волнового пакета в момент времени $t = 75$

Рис. 16.19. Огибающая волнового пакета в момент времени $t = 100$

Рис. 16.20. Огибающая волнового пакета в момент времени $t = 125$

Рис. 16.21. Огибающая волнового пакета в момент времени $t = 150$

Рис. 16.22. Огибающая волнового пакета в момент времени $t = 175$

Рис. 16.23. Огибающая волнового пакета в момент времени $t = 200$

Рис. 16.24. Огибающая волнового пакета в момент времени $t = 225$

Рис. 16.25. Огибающая волнового пакета в момент времени $t = 250$

Рис. 16.26. Огибающая волнового пакета в момент времени $t = 275$

Рис. 16.27. Огибающая волнового пакета в момент времени $t = 300$

***Задача* 16.13**

Проверьте точность интегрирования нестационарного уравнения Шредингера, используя следующий прием.

1. *Проинтегрируйте уравнение на некотором временном интервале, затем измените знак шага по времени и проинтегрируйте уравнение Шредингера в обратном направлении на временном интервале аналогичной длины.*

2. *Сравните начальную форму волнового пакета и волнового пакета, полученного интегрированием уравнения Шредингера описанным выше способом.*

3. *Дополните описанный выше документ функцией, возвращающей мгновенные значения энергии пакета.*

4. *Проверьте, что точность сохранения энергии не зависит от шага интегрирования.*

***Задача* 16.14**

Исследуйте движение квантового волнового пакета в отсутствие потенциала. Сравните полученные результаты с результатами, полученными при описании движения волнового пакета в среде с дисперсией в главе 9.

***Задача* 16.15**

Исследуйте рассеивание волновых пакетов с различной формой, шириной и энергией на потенциальных ямах и барьерах различной формы и размера.

1. *Объясните особенности наблюдаемой эволюции пакетов во времени.*

2. *Для прямоугольной ямы и ступенчатого потенциала сравните относительные вероятности прохождения и отражения волнового пакета с соответствующими аналитическими выражениями, приведенными в [1, задачи 23—25].*

3. *Рассмотрите резонансный случай при рассеивании пакета на потенциальной яме. Пронаблюдайте распад волнового пакета на быструю и затухающую компоненты.*

***Задача* 16.16**

Исследуйте поведение волнового пакета в двойной яме, представленной на рис. 16.13.

***Задача* 16.17**

Замените вычислительную формулу (16.47) на неустойчивый явный метод или на неунитарный неявный метод и исследуйте получаемые решения нестационарного уравнения Шредингера.

***Задача* 16.18**

Замените граничное условие Дирихле на одной из границ сетки на условие Неймана с нулевой производной на этой же границе. Опишите эволюцию волнового пакета при его приближении к данной границе.

16.5. ОЦЕНКА ЭНЕРГИИ ОСНОВНОГО СОСТОЯНИЯ КВАНТОВОЙ СИСТЕМЫ МЕТОДОМ МОНТЕ-КАРЛО

В основу метода нахождения энергии и волновой функции основного состояния положена связь уравнения Шредингера с процессом диффузии в мнимом времени [2–4]. Продемонстрируем основные идеи данного метода на примере движения частицы в одномерном потенциале, описываемом одномерным уравнением Шредингера (16.6), которое для свободной частицы имеет вид

$$i\hbar\frac{\partial\psi\left(x,t\right)}{\partial\,t}=-\frac{\hbar^2}{2m}\frac{\partial^2}{\partial\,x^2}\psi\left(x,t\right). \tag{16.62}$$

Выполнив в (16.62) замену переменной

$$t\to -i\cdot\tau\hbar,$$

получим

$$\frac{\partial\psi\left(x,\tau\right)}{\partial\tau}=\frac{\hbar^2}{2m}\frac{\partial^2\psi\left(x,\tau\right)}{\partial x^2}. \tag{16.63}$$

Уравнение (16.63) по свой форме совпадает с уравнением диффузии

$$\frac{\partial\,P\left(x,t\right)}{\partial\,t}=D\frac{\partial^2 P\left(x,t\right)}{\partial\,x^2}, \tag{16.64}$$

где $P(x,t)$ — плотность вероятности обнаружить частицу в момент времени t в точке с координатой x.

Следовательно, можно интерпретировать волновую функцию $\psi(x,\tau)$ как плотность вероятности диффузионного процесса с коэффициентом диффузии $D=\hbar^2/2m$. Вспомним, что ранее в главе 12 мы продемонстрировали возможность использования метода случайных блужданий для моделирования решений уравнения диффузии. Таким образом, указанная аналогия между уравнениями (16.63), (16.64) позволяет вместо непосредственного решения уравнения Шредингера искать решение задачи о случайных блужданиях.

Уравнение Шредингера (16.6) относится к классу линейных дифференциальных уравнений в частных производных, поэтому при решении задачи можно рассмотреть движение не одного, но большого количества пешеходов, движущихся независимо

друг от друга. Для понимания роли потенциальной энергии, перепишем одномерное уравнение Шредингера (16.6) в следующем виде:

$$\frac{\partial\,\psi\,(x,\tau)}{\partial\,\tau} = \frac{\hbar^2}{2m}\frac{\partial^2\psi\,(x,\tau)}{\partial\,x^2} - U\,(x)\,\psi\,(x,\tau)\,. \tag{16.65}$$

При отсутствии члена $U(x)\psi(x,\tau)$ в правой части уравнения (16.65), оно превращается в обыкновенное уравнение диффузии. С другой стороны, при отсутствии в правой части члена

$$\frac{\hbar^2}{2m}\frac{\partial^2\psi\,(x,\tau)}{\partial\,x^2}$$

уравнение (16.65) можно трактовать как кинетическое уравнение, описывающее ветвящиеся процессы, такие как, например, радиоактивный распад, динамика численности биологической популяции и т. д. Следовательно, уравнение (16.63) можно трактовать как комбинацию диффузионного и ветвящегося процессов, в которых число пешеходов увеличивается или уменьшается в зависимости от знака функции $U(x)$. Отметим, что диффузионной интерпретации требуется, чтобы амплитуда вероятностей была неотрицательной функцией.

Общее решение уравнения (16.6) находится в соответствии с формулой (16.12). Выполнив в (16.12) замену переменной

$$t \to -i\cdot\tau\hbar,$$

получим общее решение уравнения (16.63) в виде

$$\psi\,(x,\tau) = \sum_n c_n\varphi_0\,(x)\;\,e^{-E_n\tau}. \tag{16.66}$$

Из (16.66) видно, что для достаточно больших τ основной вклад в сумму дает член, соответствующий состоянию с минимальной энергией E_0 (основному состоянию):

$$\psi\,(x,\tau) \approx c_0\varphi_0\,(x)\,e^{-E_0\tau}, \tag{16.67}$$

а пространственная зависимость плотности распределения вероятности пропорциональна собственной функции основного состояния $\varphi_0(x)$.

В то же время решение в виде (16.67) имеет существенный недостаток, заключающийся в том, что $\lim\limits_{\tau\to\infty}\psi\,(x,\tau) = 0$ для всех x.

Для устранения отмеченного недостатка можно отсчитывать E_0 от произвольного значения начальной энергии U_{ref}, которая выбирается из условия достижения стационарного распределения.

Для нахождения формулы для оценки E_0 перепишем (16.65) и (16.67) в явном виде, введя начало отсчета энергии в точке U_{ref}:

$$\frac{\partial\,\psi\,(x,\tau)}{\partial\,\tau} = \frac{\hbar^2}{2m}\frac{\partial^2\psi\,(x,\tau)}{\partial\,x^2} - [U\,(x) - U_{ref}]\,\psi\,(x,\tau)\,, \tag{16.68}$$

$$\psi\,(x,\tau) \approx c_0\varphi_0\,(x)\,e^{-(E_0-U_{ref})\tau}. \tag{16.69}$$

Проинтегрировав (16.68) по x и учтя, что

$$\lim_{|x|\to\infty} \frac{\partial \psi}{\partial x} = 0,$$

получаем

$$\int \frac{\partial \psi(x,\tau)}{\partial \tau}\, dx = -\int U(x)\,\psi(x,\tau)\, dx + U_{ref} \int \psi(x,\tau)\; dx. \tag{16.70}$$

Продифференцировав (16.70) по x, найдем

$$\frac{\partial \psi(x,\tau)}{\partial \tau} = (U_{ref} - E_0)\,\psi(x,\tau)\,. \tag{16.71}$$

Подставив (16.71) в (16.69), получаем

$$\int (U_{ref} - E_0)\,\psi(x,\tau)\; dx = -\int U(x)\,\psi(x,\tau)\, dx + U_{ref} \int \psi(x,\tau)\; dx. \tag{16.72}$$

Сократив в (16.72) члены, пропорциональные U_{ref}, получаем

$$E_0 \int \psi(x,\tau)\, dx = \int U(x)\,\psi(x,\tau)\; dx, \tag{16.73}$$

откуда

$$E_0 = \frac{\int U(x)\,\psi(x,\tau)\; dx}{\int \psi(x,\tau)\; dx}. \tag{16.74}$$

Так как $\psi(x,\tau)$ является аналогом плотности пешеходов $n(x,\tau)$ в точке x в момент времени τ, для случая случайных блужданий уравнение (16.74) можно окончательно записать в виде

$$E_0 = \langle U \rangle = \frac{\sum\limits_{i=1}^{N} n(x_i,\tau)\,U(x_i)}{\sum\limits_{i=1}^{N} n(x_i,\tau)}, \tag{16.75}$$

где N — число пешеходов.

Оценка E_0 получается, как среднее по нескольким значениям τ, отвечающим стационарному распределению.

Волновая функция основного состояния $\varphi_0(x)$ есть функция распределения плотности вероятности пешеходов по узлам сетки $n(x_i)$, вычисляемая как среднее по нескольким значениям τ, отвечающим стационарному распределению.

Метод случайных блужданий для нахождения энергии основного состояния квантовой частицы реализуется следующей последовательностью действий:

1. Размещение N_0 пешеходов случайным образом в начальные точки x_i.

2. Вычисление опорного потенциала

$$U_{ref} = \frac{1}{N_0} \sum_{i}^{N} U(x_i).$$

3. Перемещение случайным образом пешехода вправо или влево на заданный шаг Δs. (Здесь длина шага Δs связана с шагом по времени соотношением

$$(\Delta s)^2 = D\tau,$$

где D — коэффициент диффузии. В системе единиц $\hbar = m = 1$ $D = 1/2$.)

4. Вычисление

$$\Delta U = [U(x) - U_{ref}]\,\Delta\tau$$

и случайного числа r, равномерно распределенного на отрезке $[0, 1]$.

5. Если $\Delta U > 0$ и $r < \Delta U$, удалить пешехода.

6. Если $\Delta U \leq 0$ и $r \leq -\Delta U$, добавить пешехода в точку x.

7. Повторить п. 3–6 для каждого из N_0 пешеходов.

8. Заменить опорный потенциал на

$$U_{ref} = \langle U \rangle - \frac{1}{N_0}(N - N_0)\,\Delta\tau,$$

где N — новое число пешеходов, $\langle U \rangle$ — средняя потенциальная энергия N пешеходов, представляющая собой оценку энергии основного состояния.

9. Повторить п. 3–8 до достижения энергией стационарного значения.

10. Повторить п. 3–9 для получения нескольких стационарных значений энергии.

11. Получить усредненное значения энергии по нескольким стационарным значениям.

Алгоритм, вычисления волновой функции основного состояния реализуется следующей последовательностью действий:

1. Размещение N_0 пешеходов случайным образом в начальные точки x_i.

2. Вычисление опорного потенциала

$$U_{ref} = \frac{1}{N_0}\sum_{i}^{N} U(x_i).$$

3. Перемещение случайным образом пешехода вправо или влево на заданный шаг Δs. (Здесь длина шага Δs связана с шагом по времени соотношением

$$(\Delta s)^2 = D\tau,$$

где D — коэффициент диффузии. В системе единиц $\hbar = m = 1$ $D = 1/2$.)

4. Вычисление
$$\Delta U = [U(x) - U_{ref}]\,\Delta\tau$$
и случайного числа r, равномерно распределенного на отрезке $[0, 1]$.

5. Если $\Delta U > 0$ и $r < \Delta U$, удалить пешехода.

6. Если $\Delta U \leq 0$ и $r \leq -\Delta U$, добавить пешехода в точку x.

7. Повторить п. 3–6 для каждого из N_0 пешеходов.

8. Заменить опорный потенциал на
$$U_{ref} = \langle U \rangle - \frac{1}{N_0}(N - N_0)\,\Delta\tau,$$
где N — новое число пешеходов, $\langle U \rangle$ — средняя потенциальная энергия N пешеходов, представляющая собой оценку энергии основного состояния.

9. Повторить п. 3–8 до достижения энергией стационарного значения.

10. Вычисление функции распределения плотности пешеходов.

11. Повторение п. 3–10 для получения нескольких стационарных значений энергии.

12. Получение усредненной функции распределения плотности вероятности по нескольким стационарным значениям.

Для реализации описанного алгоритма создадим файл `WalkSh.m`, содержащий описание функции, возвращающей среднее значение энергии основного состояния и мгновенные (на каждом шаге Монте-Карло) значения числа пешеходов в заданных интервалах.

```
% листинг функции WalkSh.m
function [Em,h]=WalkSh(N,Srange,K,ds,NTrial,Interval)
% функция, возвращающая среднее значение
% энергии основного состояния
% и мгновенные (на каждом шаге Монте-Карло)
% значения числа
% пешеходов в заданных интервалах
% N - число пешеходов
% Srange - размер начальной области
% размещения пешеходов
% K - доля неиспользуемых нестационарных состояний
% ds - размер шага пешехода
% NTrial - число испытаний
% Interval - вектор, содержащий координаты
% концов отрезка, используемых для
% вычисления плотности
% распределения пешеходов
```

```
Vref=0;
% инициализация начальной
% конфигурации пешеходов
for i=1:N
     Xsite(i)=Srange*(0.5-rand(1));
     Vref=Vref+V(Xsite(i));
end;
Vref=Vref/N;
Ninitial=N; % число пешеходов
            % в начале испытания
Nwalker=N; % число пешеходов
           % в ходе испытания
dt=ds.^2; % шаг по времени
Nequial=K*NTrial; % номер первого
                  % стационарного состояния
m=1;
% реализация метода Монте-Карло
for i=1:NTrial
     Vsum=0; % полная потенциальная энергия
     for j=Ninitial:-1:1
           if rand(1)<0.5
          Xsite(j)=Xsite(j)+ds;
    else
          Xsite(j)=Xsite(j)-ds;
    end;
    Potential=V(Xsite(j));
    dv=Potential-Vref;
    if dv<0 % проверка на добавление пешехода
      if rand(1)<-dv*dt
        Nwalker=Nwalker+1;
        Xsite(Nwalker)=Xsite(j); % новый пешеход
        Vsum=Vsum+2*Potential;
      else
        Vsum=Vsum+Potential; % добавляем только
                             % энергию старого
                             % пешехода
      end;
    else
      if rand(1)<dv*dt % проверка на удаление
                       % пешехода
         Xsite(j)=Xsite(Nwalker);
         Nwalker=Nwalker-1;
      else
         Vsum=Vsum+Potential;
```

```
      end;
    end;
  end;
  Vave=Vsum/Nwalker; % средний потенциал
  if i>Nequial
    E(m)=Vave;
    for n=1:Nwalker
      X(n)=Xsite(n);
    end;
    h1=hist(X,Interval);
    if m==1
      h=h1';
    else
      h=cat(2,h,h1');
    end;
    m=m+1;
  end;
  Vref=Vave-(Nwalker-N)/(N*dt); % новый опорный
                                % потенциал
end;
Em=mean(E);

function z=V(x);
% функция, описывающая потенциал
z=0.5*x.^2;
```

Далее необходимо выполнить следующую последовательность команд, сохраненную нами в файле `Glava16_7.m`:

```
% Листинг файла Glava16_7.m
N=1000; % начальное число пешеходов
ds=0.1; % длина шага пешехода
NTrial=1000; % число испытаний
Srange=2; % размер начальной области
                  % размещения пешеходов
K=0.2; % доля неиспользуемых
       % нестационарных состояний
Ni=40; % число интервалов, используемых
       % для вычисления плотности
       % распределения пешеходов

% вычисление границ интервалов
i=1:Ni+1;
Interval(i)=-2*Srange+(4*Srange/Ni*(i-1));

% вычисление усредненной энергии
```

```
% основного состояния и вычисление
% мгновенных (на каждом шаге Монте-Карло)
% волновых функций
[A h]=WalkSh(N,Srange,K,ds,NTrial,Interval);
A % энергия основного состояния

% вычисление значений усредненной
% волновой функции
Nx=size(h,1);
Ny=size(h,2);
for i=1:Nx
    for j=1:Ny
        A(j)=h(i,j);
    end;
    A1(i)=sum(A);
end;
% визуализация потенциала и
% усредненной волновой функции
Ni=500;
dx=(max(Interval)-min(Interval))/(Ni-1);
i=1:Ni;
x(i)=min(Interval)+dx*(i-1);
plot(x,0.5*x.^2,'k',Interval,A1/max(A1),'k')
```

Ниже представлены результаты выполнения описанной последовательности команд.

1. Значение энергии основного состояния:

```
A =
0.4903
```

2. Нормированная волновая функция основного состояния (рис. 16.28).

Задача 16.19

1. *Сравните найденное значение энергии основного состояния с его точным значением [1, задача 30].*
2. *Исследуйте зависимость точности численного значения энергии основного состояния, найденного методом Монте-Карло от числа пешеходов и числа шагов Монте-Карло.*

Задача 16.20

1. *Дополните файл* `Glava16_7.m` *блоком, позволяющим нормировать волновую функцию так, чтобы она удовлетворяла условию (16.2).*

Рис. 16.28. Волновая функция основного состояния, нормированная на ее максимальное значение, и график зависимости потенциальной энергии от координаты

2. *Сравните численное значение нормированной волновой функции с соответствующими значениями известной аналитической функции [1, задача 30].*

Указание: *для вычисления интеграла (16.2) проведите интерполяцию усредненной зависимости* $n(x_i)$.

***Задача* 16.21**

Получите численное решение одномерного уравнения Шредингера для ангармонического осциллятора с потенциалом

$$U(x) = \frac{1}{2}x^2 + bx^3.$$

Рассмотрите следующие значения параметра: $b = 0{,}1, 0{,}2, 0{,}5$.

В чем проявляется увеличение нелинейности потенциала?

***Задача* 16.22**

Внесите необходимые изменения в файл `Glava16_7.m` *и вычислите энергию и волновую функцию основного состояния прямоугольной потенциальной ямы.*

Найдите зависимость энергии основного состояния от U_0.

Сравните полученные результаты с аналогичными результатами, полученными методами, описанными в разделах 16.2, 16.4.

***Задача* 16.23**

Вычислите изменение значений энергии и волновых функций волнового состояния, частицы, находящейся в прямоугольной потенциальной яме, обусловленные малым возмущением:

$$U(x) = \begin{cases} U_b, & \text{если} \quad |x| \leqslant 0,2, \\ 0, & \text{если} \quad 0,2 < |x| \leqslant 1, \\ U_0, & \text{если} \quad |x| > 1. \end{cases}$$

16.6. ОЦЕНКА ЭНЕРГИИ ОСНОВНОГО СОСТОЯНИЯ КВАНТОВОЙ СИСТЕМЫ ВАРИАЦИОННЫМ МЕТОДОМ МОНТЕ-КАРЛО

Альтернативный подход к оценке энергии основного состояния квантовых систем состоит в использовании вариационного метода, позволяющего получить оценки верхних границ значений энергии основного состояния. Данный вариационный метод находит многочисленные приложения в атомной физике, молекулярной физике, ядерной физике и физике конденсированных сред. Подробное рассмотрение многочисленных примеров, относящихся к перечисленным выше разделам физики, выходит за рамки нашей книги, поэтому далее мы ограничимся обсуждением основных идей вариационного метода Монте-Карло.

Рассмотрим квантовую систему, описываемую оператором Гамильтона $\hat{H}$ вида (16.10). Вариационный принцип утверждает, что для произвольной функции $\phi(x)$ выполняется неравенство

$$\langle H \rangle = \frac{\int \phi^*(x)\,\hat{H}\phi(x)\,dx}{\int \phi^*(x)\,\phi(x)\,dx} \geqslant E_0, \tag{16.76}$$

где E_0 — энергия основного состояния системы.

Неравенство (16.76) является основой вариационного метода. Механизм реализации данного метода состоит в физически обоснованном выборе пробной функции, зависящей от нескольких параметров, вычислении $\langle H \rangle$ в соответствии с (16.74), варьировании одного или более параметров до тех пор, пока не будет достигнуто минимальное значение $\langle H \rangle$. Данное значения является верхней границей истинного значения энергии основного состояния.

Рассмотрим реализацию вариационного метода Монте-Карло на примере одномерной системы. Введем пространственную временную сетку $x_i = i\Delta x$, где Δx — шаг сетки, в узлах которой будем вычислять значения функции. Алгоритм вариационного метода Монте-Карло реализуется следующей последовательностью действий:

1. Выбрать физически обоснованные значения пробной функции $\phi(x)$ в точках x_i.
2. Выбрать случайную точку x_i и изменить значение $\phi_i = \phi(x_i)$ на случайную величину $\Delta\varphi \in [-\delta, \delta]$.
3. Вычислить изменение энергии ΔE для $\langle H \rangle$.
4. Если $\Delta E \leq 0$, принять выполненное изменение ϕ_i.
5. Если $\Delta E > 0$, отклонить выполненное изменение $\Delta E \leq 0\phi_i$.
6. Повторять п. 2—5 до достижения постоянного (с заданной точностью) значения $\langle H \rangle$.

Рассмотрим реализацию описанного алгоритма в MATLAB для случая, когда волновая функция является симметричной. Далее приводится листинг m-файла `Variation.m`, содержащего описание функции, возвращающей верхнюю границу значения энергии основного состояния, соответствующую волновую функцию и мгновенные (на каждом шаге итерационного процесса) значения энергии.

```
% листинг файла Variation.m
function [Em,psi,W] = Variation(Xmax,delta,N)
% функция, возвращающая верхнюю границу
% значения энергии основного состояния,
% соответствующая волновая функция,
% и мгновенные (на каждом шаге)
% итерационного процесса значения энергии
% Xmax - максимальное значение координаты
% delta - максимальное изменение
% волновой функции
% N - число узлов координатной сетки
% в каждом направлении

dx=Xmax/N; % расстояние между
           % узлами координатной сетки
delta=2*delta;
Nmove=1000*N; % число шагов
              % итерационного процесса
Norm=0;
k=1;
% вычисление значений начального приближения
for i=1:2*N+1
     psi(i)=f(-Xmax+dx*(i-1));
end;
Norm=sum(psi.^2); % нормировочный множитель

% вычисление начальной энергии
E=0;
for i=N+2:2*N
   x=-Xmax+(i-1)*dx;
   pe=psi(i)*psi(i)*V(x); % потенциальная энергия
   % кинетическая энергия
   ke=0.5*psi(i)*(2*psi(i)-psi(i+1)-psi(i-1))/dx^2;
   E=E+2*(pe+ke);
end;
E=E+E*V(0)*psi(N+1)^2;
E=E+0.5*psi(N+1)*(2*psi(N+1)-psi(N+2)-psi(N))/dx^2;
% варьирование волновой функции
Ecum=0;
for Imove=1:Nmove
      i=floor((N-1)*rand(1))+N+2; % координата точки
      dpsi=delta*(rand(1)-0.5); % изменение волновой
                                % функции
       Psi_Trial=psi(i)+dpsi; % пробное значение
                              % волновой функции в i-м узле
```

```
      % вычисление изменения энергии,
      % обусловленное изменением волновой
      % функции в i-м узле
      dpsi2=Psi_Trial*Psi_Trial-psi(i)*psi(i);
      dV=dpsi2*V(-Xmax+(i-1)*dx); % изменение
                                  % потенциальной энергии
     % изменение кинетической энергии
      if i==N+1
        dK=(dpsi2-dpsi*(psi(i+1)+psi(i-1)))/dx^2;
        Etrial=E+dK+dV;
        Norm_Trial=Norm+dpsi2;
      else
        Etrial=E+2*(dV+dK);
        Norm_Trial=Norm+2*dpsi2;
   end;
   dE=Etrial/Norm_Trial-E/Norm;
   if dE<0 % принятие изменения
           % волновой функции
      psi(i)=Psi_Trial;
      psi(2*N+2-i)=Psi_Trial;
      Norm=Norm_Trial;
      E=Etrial;
   end;
    Ecum=Ecum+E/Norm;
    W(Imove)=E/Norm;
end;
Em=Ecum/Nmove;

function z = f(x)
% функция, описывающая
% начальное приближение
z=exp(-x.^2);

function z = V(x)
% функция, описывающая потенциал
z=0.5*x.^2;
```

Далее необходимо выполнить следующую последовательность команд, сохраненную нами в файле `Glava18_8.m`.

```
% листинг файла Glava18_8.m
Xmax=4; % максимальное значение координаты
N=40; % число узлов в каждом
      % направлении оси OX
delta=0.1; % максимальное изменение
           % волновой функции
```

```
[Em psi W] = Variation(Xmax,delta,N);

% визуализация волновых функций и
% зависимости мгновенного значения
% энергии от номера шага итерации

Ni=1000;
i=1:Ni;
dx=Xmax/Ni;
x(i)=-Xmax+2*dx*i;
j=1:length(psi);
X(j)=-Xmax+2*Xmax/length(psi)*(j-1);

figure(1)
plot(X(j),psi(j),'k',...
     x,exp((-x.^2)/2),'-.k',...
     x,exp(-x.^2),'--k');

figure(2)
plot(W,'k');
axis([0 length(W) 0.5 0.65])
```

Результат выполнения описанной выше последовательности команд представлен на рис. 16.29, 16.30.

Рис. 16.29. Волновая функция основного состояния гармонического осциллятора (1 — начальное приближение, 2 — после 40 000 итераций, 3 — точное значение)

Рис. 16.30. Зависимость мгновенного значения энергии основного состояния гармонического осциллятора от номера шага итерации

***Задача* 16.24**

Сравните скорость сходимости итерационного процесса при выборе в качестве пробной волновой функции $\phi(x) = \cos(x)$, $\phi(x) = 1$, $\phi(x) = e^{-x^2/1.5}$. *Исследуйте влияние величины максимального изменения волновой функции на точность вычисления собственного значения энергии и собственной волновой функции основного состояния.*

***Задача* 16.25**

Модифицируйте функцию `Variation.m` *так, чтобы изменения волновой функции проводились последовательно для каждого узла пространственной сетки. Приводит ли данная модификация программы к ускорению сходимости итерационного процесса?*

***Задача* 16.26**

Вычислите энергию основного состояния нелинейного гармонического осциллятора с потенциалом

$$U(x) = \frac{1}{2}x^2 + bx^4.$$

Сравните полученные значения собственной энергии и волновой функции основного состояния, полученные вариационным методом Монте-Карло, с соответствующими значениями, полученными в квазиклассическом приближении, методом пристрелки и методом случайных блужданий. Рассмотрите случаи $b > 0$, $b < 0$. *Как зависит энергия основного состояния от коэффициента* b, *определяющего степень нелинейности системы?*

***Задача* 16.27**

Используйте описанный выше документ для расчета энергии основного состояния прямоугольной потенциальной ямы с параметрами $U_0 = -2, -20$, $a = 1$. *Положите* $X\max = 2$, $N = 40$, $\Delta = 0,1$. *Рассмотрите начальные пробные волновые функции* $\phi = \cos(x)$, $\phi(x) = 1$, $\phi(x) = e^{-x^2/1.5}$.

1. *Какая из перечисленных выше функций дает наиболее быструю сходимость к искомому результату?*
2. *Чему равна глубина подбарьерного проникания волновой функции?*
3. *Сравните полученные значения собственной энергии и волновой функции основного состояния, полученные вариационным методом Монте-Карло, с соответствующими значениями, полученными в квазиклассическом приближении, методом пристрелки и методом случайных блужданий.*

***Задача* 16.28**

Вычислите вариационным методом Монте-Карло изменение значений энергии и волновых функций волнового состояния, частицы, находящейся в прямоугольной потенциальной яме, обусловленные малым возмущением:

$$U(x) = \begin{cases} U_b, & \text{если} \quad |x| \leqslant 0,2, \\ 0, & \text{если} \quad 0,2 < |x| \leqslant 1, \\ U_0, & \text{если} \quad |x| > 1. \end{cases}$$

Сравните полученные значения собственной энергии и волновой функции основного состояния, полученные вариационным методом Монте-Карло, с соответствующими значениями, полученными в квазиклассическом приближении, методом пристрелки и методом случайных блужданий.

***Задача* 16.29**

Сравните скорость сходимости итерационного процесса при выборе в качестве пробной волновой функции $\phi = \cos(x)$, $\phi(x) = 1$, $\phi(x) = e^{-x^2/1.5}$.

Исследуйте влияние величины максимального изменения волновой функции на точность вычисления собственного значения энергии и собственной волновой функции основного состояния.

***Задача* 16.30**

Модифицируйте функцию `Variation.m` *так, чтобы изменения волновой функции проводились последовательно для каждого узла пространственной сетки.*

Приводит ли данная модификация программы к ускорению сходимости итерационного процесса?

***Задача* 16.31**

Используйте файлы, описанные в настоящем разделе, для вычисления энергии основного состояния нелинейного гармонического осциллятора с потенциалом

$$U(x) = \frac{1}{2}x^2 + bx^4.$$

1. *Сравните полученные значения собственной энергии и волновой функции основного состояния, полученные вариационным методом Монте-Карло, с соответствующими значениями, полученными в квазиклассическом приближении, методом пристрелки и методом случайных блужданий. Рассмотрите случаи* $b > 0$, $b < 0$.

2. *Как зависит энергия основного состояния от коэффициента* b, *определяющего степень нелинейности системы?*

Для вычисления собственного значения энергии основного состояния можно использовать принцип вариации постоянных коэффициентов разложения волновой функции по полному набору ортогональных базисных функций f_n:

$$\phi = \sum_n c_n f_n, \tag{16.77}$$

Подставив (16.77) в (16.76), получим

$$\langle H \rangle = \frac{\int \sum\limits_m c_m^* f_m \hat{H} \sum\limits_n c_n f_n dx}{\int \sum\limits_m c_m^* f_m \sum\limits_n c_n f_n dx} = \frac{\sum\limits_{m,n} c_m^* c_n \int f_m^* \hat{H} f_n dx}{\sum\limits_n |c_n|^2} \geqslant E_0. \tag{16.78}$$

Неравенство (16.78) превращается в точное равенство только в том случае, когда функция $\phi(x)$ является собственной функцией оператора $\hat{H}$, соответствующей собственному значению E_0. Для вычисления математического ожидания $\langle H \rangle$ необходимо вычислить матричные элементы

$$\int f_m^* \hat{H} f_n dx.$$

При необходимости вычисления матричных элементов в пространстве больших размерностей можно использовать для вычисления интегралов метод Монте-Карло.

После вычисления матричных элементов проводится вариация коэффициентов c_n^*, c_m до тех пор, пока не будет достигнуто минимальное значение $\langle H \rangle$. Данное значение является верхней границей истинного значения энергии основного состояния.

Рассматриваемый алгоритм вариационного метода Монте-Карло реализуется следующей последовательностью действий:

1. Выбрать ортонормированный набор базисных функций.
2. Выбрать числа используемых базисных функций Nb.
3. Вычислить значения матричных коэффициентов

$$\int f_m^* \hat{H} f_n dx$$

4. Задать начальные значения коэффициентов c_i.
5. Вычислить начальное значение энергии E.
6. Выбрать случайным образом коэффициент c_i и изменить его значение на случайную величину $\Delta c \in [-\delta; \delta]$.
7. Вычислить изменение энергии $\langle H \rangle$, обусловленное изменением значения коэффициента.
8. Если $\Delta E \leq 0$, принять новое значение коэффициента.
9. Если $\Delta E > 0$, отклонить новое значение коэффициента.
10. Повторять п. 6—9 до достижения постоянного (с заданной точностью) значения $\langle H \rangle$.

***Задача* 16.32**

Решите задачу 16.29, используя метод вариации постоянных разложения собственной функции по ортонормированному базису. В качестве ортонормированного базиса выберите собственные волновые функции частицы, находящейся в бесконечно глубокой потенциальной яме.

***Задача* 16.33**

Определите, используя метод вариации постоянных, разложения собственной функции по ортонормированному базису, собственное значение энергии основного состояния частицы, находящейся в потенциале Морза

$$U(x) = U_0 \left(e^{-2\alpha x} - 2e^{-\alpha x}\right), \tag{16.79}$$

используя в качестве набора ортонормированного набора базисных функций базисные функции линейного гармонического осциллятора:

$$f_n\left(x\right) = N_n H_n\left(x\right) e^{-\frac{x^2}{2}}, \tag{16.80}$$

где $H_n(x)$ — полиномы Эрмита, N_n — нормировочная постоянная,

$$H_n\left(x\right) = (-1)^n e^{x^2} \frac{d^n}{d\,x^n} e^{-x^2}, \tag{16.81}$$

$$N_n = \left[\frac{1}{\pi^{\frac{1}{2}} 2^n n!}\right]. \tag{16.82}$$

1. *Создайте функцию, позволяющую для заданных волновых функций вычислять матричные элементы*

$$\int f_m^* \hat{H} f_n dx,$$

вычислите значения элементов матрицы для $n, m = 2, 3, \ldots, 10$.

2. *Создайте функцию, реализующую описанный выше алгоритм вариации постоянных разложения волновой функции по ортонормированному базису.*

3. *Проведите вычисления для различного числа используемых базисных функций* n, *выбирая в качестве начальных значений* c_0, $c_n = 0$ *для* $n \neq 0$.

   Как зависят получаемые значения энергии от числа используемых базисных функций?

   Как зависят получаемые значения энергии от максимальной величины изменения шага?

   Сравните полученные численные значения с известными точными значениями энергии [1, задача 70].

ЛИТЕРАТУРА К ГЛАВЕ 16

1. *Флюгге З.* Задачи по квантовой механике. Т. 1. М.: Мир, 1974.
2. *Anderson J. B.* A random walk simulation of Schrodinger equation: H_3^+, J. Chem. Phys, 63, 1499. 1975.
3. *Anderson J. B.* Quantum chemistry by random walk. H^2P, $H_3^+ D_{3h}{}^1A_1'$, $H_2{}^3\sum_u^+$, $H_4{}^1\sum_g^+$, Be^1S, J. Chem. Phys, 65, 4121. 1976.
4. *Anderson J. B.* Quantum chemistry by random walk: Higher accuracy, J. Chem. Phys, 73, 3897. 1980.
5. *Ландау Л. Д., Лифшиц Е. М.* Квантовая механика. Нерелятивистская теория. М.: Наука, 2001.
6. *Давыдов А. С.* Квантовая механика. М.: Наука, 1973.
7. *Крайнов А. В., Мигдал А. Б.* Качественные методы в квантовой механике. М.: Наука, 1976.
8. *Мигдал А. Б.* Качественные методы в квантовой теории. М.: Наука, 1975.
9. Методы Монте-Карло в статистической физике / Под ред. К. Биндера. М.: Мир, 1982.
10. *Гулд Х., Тобочник Я.* Компьютерное моделирование в физике. Т. 2. М.: Мир, 1990.
11. *Кунин С.* Вычислительная физика. М.: Мир, 1992.

Глава 17

МОДЕЛИРОВАНИЕ ФРАКТАЛЬНЫХ ОБЪЕКТОВ

17.1. ВВЕДЕНИЕ

Фракталы — математические объекты дробной размерности, название которых было введено в математику Б. Мандельбротом [1], являются в настоящее время как предметом самостоятельных математических исследований, так и инструментарием, используемым в целом ряде прикладных задач нелинейной динамики [2, 3], теории хаоса [4], обработки сигналов [5]. Однако только относительно недавно появилось первое полноценное учебное пособие по новой быстро развивающейся математической дисциплине [6], основой которого стал учебный курс, преподававшийся автором в течение ряда лет в университете Миссури, Колумбия.

Так как при изучении фракталов и хаоса большую роль играет компьютерное моделирование, в курсе предусмотрено параллельное изучение теоретических вопросов и проведение компьютерных экспериментов. Это отличает его структуру от традиционной структуры большинства математических курсов: теорема—доказательство—пример—задача. В [6] в обобщенном виде подробно описаны известные алгоритмы построения фрактальных объектов (L-системы и терл-графика, аффинные преобразования, системы итерированных функций, случайные системы итерированных функций и др.). Однако соответствующих программ, созданных на каком-либо языке программирования или в математическом пакете, не приводится. В то же время опыт практической реализации алгоритмов построения фрактальных объектов, описанных в [6], в каком-либо из современных математических пакетов (MATLAB, Mathcad, Maple, Matematica и т. д.), широко используемых в настоящее время в преподавании целого ряда физико-математических дисциплин (см., например, [7]), показывает, что существует необходимость внесения в них определенных корректировок, учитывающих особенности выбранного пакета (в первую очередь графические).

В данной главе обсуждаются алгоритмы построения классических фракталов и их программные реализации, созданные в MATLAB.

17.2. ИЗБРАННЫЕ ПОНЯТИЯ ФРАКТАЛЬНОЙ ГЕОМЕТРИИ И ХАОТИЧЕСКОЙ ДИНАМИКИ

Обсудим отдельные понятия фрактальной геометрии и хаотической динамики, следуя [6].

Самоподобие. Разделим отрезок прямой на N равных частей. Как очевидно, каждую полученную часть отрезка можно считать копией всего отрезка, уменьшенной в r раз (рис. 17.1).

Рис. 17.1. Связь размерности куба и самоподобия

При этом N и r связаны соотношением

$$N \cdot r = 1. \tag{17.1}$$

Одномерный случай легко обобщается для объектов больших размерностей. Если квадрат разбить на N равных квадратов, площадь которых в $1/r^2$ раз меньше площади исходного квадрата, то соотношение (17.1) примет вид

$$N \cdot r^2 = 1. \tag{17.2}$$

Если разбить куб на N равных кубов с объемом в $1/r^3$ раз меньше исходного, то соотношение, аналогичное соотношениям (17.1), (17.2), запишется в виде

$$N \cdot r^3 = 1. \tag{17.3}$$

Из (17.1)–(17.3) видно, что степень для каждого из рассмотренных объектов (отрезок, двумерный квадрат, трехмерный куб), связаны как степень r в соотношении с числом равных подобъектов N и коэффициентом подобия r:

$$N \cdot r^d = 1. \tag{17.4}$$

Множества, представленные на рис. 17.1, обладают целой размерностью.

Фрактальная размерность. Б. Мандельброт показал, что возможно построение таких геометрических объектов, у которых показатель степени в равенстве (17.4) не является целым, то есть при разбиении исходного множества на N непересекающихся подмножеств, полученных масштабированием оригинала с коэффициентом r, значение d не будет выражаться целым числом. Данные множества называются самоподобными фракталами. Величину d называют фрактальной размерностью или размерностью подобия.

Явное выражение для d через N находится логарифмированием обеих частей (17.4):

$$d = \frac{\log N}{\log 1/r}. \tag{17.5}$$

Классическими примерами фрактальных объектов являются подробно обсуждаемые далее снежинка Коха

$$d \approx 1,2618,$$

и ковер Серпинского

$$d \approx 1,5850.$$

17.3. РЕКУРСИВНЫЙ АЛГОРИТМ ПОСТРОЕНИЯ ФРАКТАЛЬНЫХ ОБЪЕКТОВ

Обсуждение алгоритма построения фрактальных объектов, основанного на использовании рекурсивной функции, проведем на примере простого самоподобного фрактала — ковра Серпинского. В рассматриваемом в данном разделе алгоритме используется способ построения, основанный на последовательном удалении из начальной области внутренних подобластей в соответствии с заданными правилами. Выберем в качестве начального множества S_0 — равносторонний треугольник вместе с областью, которую он замыкает. Удалим внутренность центральной треугольной области и обозначим оставшееся множество S_1 (рис. 17.2).

Рис. 17.2. Последовательность построения ковра Серпинского

Затем повторим описанный процесс для каждого из трех оставшихся треугольников и получим следующее приближение S_2. Продолжая таким образом, получим

последовательность вложенных множеств S_n, пересечение которых и образует ковер Серпинского (рис. 17.2). Из построения видно, что ковер является объединением $N = 3$ существенно непересекающихся уменьшенных в два раза копий (коэффициенты подобия по горизонтали и вертикали в данном случае оказываются одинаковыми $r = 1/2$).

Фрактальная размерность ковра Серпинского d равна

$$d = \frac{\log(3)}{\log(2)} \approx 1,5850.$$

Для построения рассматриваемого фрактала, как очевидно из вышеизложенного, можно использовать следующий алгоритм:

1. Задать порядок ковра N.
2. Задать координаты вершин исходного треугольника ABC: (X_A, Y_A), (X_B, Y_B), (X_C, Y_C).
3. Построить равносторонний треугольник ABC и залить его черным цветом.
4. Вычислить координаты середин сторон треугольника ABC.

$$dx = \frac{X_B - X_A}{2}, \quad dy = \frac{Y_B - Y_A}{2},$$

$$X_{A'} = X_A + dx, \quad Y_{A'} = Y_A + dy,$$

$$X_{B'} = X_B + dx + dx/2, \quad Y_{B'} = Y_B + dy,$$

$$X_{C'} = X_C + dx/2, Y'_C = Y_c + dy.$$

5. Построить треугольник $A'B'C'$ и залить его белым цветом.
6. Повторить N раз действия, описанные в п. 4, 5 для треугольников $AA'C'$, $A'BB'$, $C'B'C$ соответственно.

Наиболее просто описанный выше алгоритм можно реализовать при использовании рекурсивной процедуры, выполняющей последовательность действий, описанных в п. 4, 5. Для реализации алгоритма в MATLAB следует создать специальную функцию, возвращающую изображение ковра Серпинского, которая сохранена нами в файле `Serpinsky.m`.

```
% Листинг файла Serpinsky.m
function z = Serpinsky(Lmax)
% функция, возвращающая
% изображение ковра Серпинского
% Lmax - порядок ковра

% задание координат вершин
% равнобедренного треугольника
```

```
x1=0;
y1=0;

x2=1;
y2=0;

x3=0.5;
y3=sin(pi/3);

h=figure(1); % инициализация  графического окна
hold on; % включение режима
         % рисования фигур
         % в одном графическом окне
fill([x1 x2 x3],[y1 y2 y3],'k'); % прорисовка равностороннего
                                 % треугольника
set(gca,'xtick',[],'ytick',[]); % отключение режима
                                % оцифровки осей
set(gca,'XColor','w','YColor','w'); % установка цвета
                                    % рисования осей

% обращение к функции,
% прорисовывающей
% равносторонние треугольники
% белого цвета
Simplex(x1,y1,x2,y2,x3,y3,0,Lmax);

hold off % отключение режима
         % рисования фигур в одном графическом окне

function z=Simplex(x1,y1,x2,y2,x3,y3,n,Lmax)
% рекурсивная функция,
% прорисовывающая
% равносторонние треугольники
% белого цвета

if n<Lmax
% задание координат вершин текущего
% равностороннего треугольника
     dx=(x2-x1)/2;
     dy=(y3-y1)/2;
     x1n=x1+dx;
     y1n=y1;
     x2n=x1+dx+dx/2;
     y2n=y1+dy;
     x3n=x1+dx/2;
```

```
    y3n=y1+dy;

    % прорисовка текущего
    % равностороннего треугольника
    n=n+1;
    fill([x1n x2n x3n],[y1n y2n y3n],'w');

    % применение функции Simplex
    % к треугольнику с вершинами
    % (Xa',Ya); (Xa',Ya); (Xc',Yc')
    Simplex(x1,y1,x1n,y1n,x3n,y3n,n,Lmax);

    % применение функции Simplex
    % к треугольнику с вершинами
    % (Xa',Ya); (Xb',Yb'); (Xb',Yb')
    Simplex(x1n,y1n,x2,y2,x2n,y2n,n,Lmax);

    % применение функции Simplex
    % к треугольнику с вершинами
    % (Xc',Yc'); (Xb',Yb'); (Xc,Yc)
    Simplex(x3n,y3n,x2n,y2n,x3,y3,n,Lmax);
end;
```

Для вывода изображения ковра Серпинского, например, седьмого порядка, следует ввести в командной строке MATLAB имя функции с соответствующим значением:

```
>> Serpinsky(7);
```

Результат, возвращаемый функцией `Serpinsky.m` (множество S_7), представлен на рис. 17.3 .

Рис. 17.3. Ковер Серпинского 7-го порядка

Описанный выше алгоритм легко обобщается для фрактальных объектов, правила построения которых аналогичны правилам построения треугольного ковра Сер-

пинского. Например, алгоритм визуализации фрактального объекта, основные этапы построения которого представлены на рис. 17.4, реализуется следующей последовательностью действий:

Рис. 17.4. Квадрат Серпинского

1. Задать порядок ковра N.
2. Задать координаты вершин исходного квадрата $ABCD$: (X_A, Y_A), (X_B, Y_B), (X_C, Y_C), (X_D, Y_D).
3. Вычислить координаты точек, делящих стороны квадрата на три равные части:

$$dx = \frac{X_B - X_A}{3}, \quad dy = \frac{Y_B - Y_A}{3},$$
$$X'_A = X_A + dx, \quad Y'_A = Y_A + dy,$$
$$X_{B'} = X_A + dx + dx, \quad Y_{B'} = Y_A + dy,$$
$$X_{C'} = X_A + dx + dx, \quad Y_{C'} = Y_A + dy + dy,$$
$$X_{D'} = X_A + dx, \quad Y_{D'} = Y_A + dy + dy.$$

4. Построить квадрат $A'B'C'D'$ и залить его белым цветом.
5. Повторить N раз действия, описанные в п. 4, 5 для квадратов с вершинами, имеющими следующие координаты:

$$\begin{array}{llll}
(X_A, Y_A), & (X_{A'}, Y_A), & (X_{A'}, Y_{A'}), & (X_A, Y_{A'}); \\
(X_{A'}, Y_A), & (X_{B'}, Y_A), & (X_{B'}, Y_{B'}), & (X_{A'}, Y_{B'}); \\
(X_{B'}, Y_A), & (X_B, Y_B), & (X_B, Y_{B'}), & (X_{B'}, Y_{B'}); \\
(X_{B'}, Y_{B'}), & (X_B, Y_{B'}), & (X_B, Y_{C'}), & (X_{C'}, Y_{C'}); \\
(X_{C'}, Y_{C'}), & (X_B, Y_{C'}), & (X_C, Y_C), & (X_{C'}, Y_{C'}); \\
(X_{D'}, Y_{D'}), & (X_{C'}, Y_{C'}), & (X_C, Y_C), & (X_{D'}, Y_{C'}); \\
(X_A, Y_{D'}), & (X_{D'}, Y_{D'}), & (X_{D'}, Y_C), & (X_D, Y_D); \\
(X_A, Y_{A'}), & (X_{A'}, Y_{D'}), & (X_{A'}, Y_{D'}), & (X_A, Y_{D'}), \quad \text{соответственно.}
\end{array}$$

Ниже приведен листинг файла `Serpinsky2.m`, который содержит описание функции, возвращающей изображение квадратного ковра Серпинского, представленного на рис. 17.4.

```
% листинг фалайла Serpinsky2.m
function z=Serpinsky2(Lmax)
% функция, возвращающая
% изображение квадратного ковра Серпинского
% Lmax - порядок ковра

% Задание координат вершин исходного квадрата
x1=0;
y1=0;

x2=1;
y2=0;

x3=1;
y3=1;

x4=0;
y4=1;

% инициализация графического окна
% установка параметров рисования ковра Серпинского
figure(1);
hold on;
set(gca,'xtick',[],'ytick',[]);
set(gca,'XColor','b','YColor','b');
fill([x1 x2 x3 x4],[y1 y2 y3 y4],'b');

% обращение к рекурсивной функции,
% прорисовывающей квадраты белого цвета на синем фоне
Quadrate(x1,y1,x2,y2,x3,y3,x4,y4,0,Lmax);

function z=Quadrate(x1,y1,x2,y2,x3,y3,x4,y4,n,Lmax)
% рекурсивная функция,
% прорисовывающая квадраты белого цвета
if n<Lmax
   % вычисление новых длин сторон квадратов
   dx=(x2-x1)/3;
   dy=(y3-y1)/3;

   % вычисление координат вершин новых квадратов
   x1n=x1+dx;
   y1n=y1+dy;
```

```
x2n=x1+dx+dx;
y2n=y1+dy;
x3n=x1+dx+dx;
y3n=y1+dy+dy;
x4n=x1+dx;
y4n=y1+dy+dy;

% прорисовка нового квадрата
fill([x1n x2n x3n x4n],[y1n y2n y3n y4n],'w');
n=n+1;

% вызов функции Quadrate для квадрата
% с вершинами (Xa,Ya), (Xa',Ya), (Xa',Ya'), (Xa,Ya')
Quadrate(x1,y1,...
                  x1+dx,y1,...
                  x1+dx,y1+dy,...
                  x1,y1+dy,...
                  n,Lmax);

% вызов функции Quadrate для квадрата
% с вершинами (Xa',Ya), (Xb',Ya),(Xb',Yb'),(Xa',Yb')
Quadrate(x1+dx,y1,...
                  x1+2*dx,y1,...
                  x1+2*dx,y1+dy,...
                  x1+dx,y1+dy,...
                  n,Lmax);

% вызов функции Quadrate для квадрата
% с вершинами (Xb',Ya),(Xb,Yb),(Xb,Yb'),(Xb',Yb')
Quadrate(x1+2*dx,y1,...
                 x2,y1,...
                 x2,y1+dy,...
                 x1+2*dx,y1+dy,...
                 n,Lmax);

% вызов функции Quadrate для квадрата
% с вершинами (Xb',Yb'), (Xb,Yb'),(Xb,Yc'),(Xc',Yc')
Quadrate(x1+2*dx,y1+dy,...
                  x2,y1+dy,...
                  x2,y1+2*dy,...
                  x1+2*dx,y1+2*dy,...
                  n,Lmax);

% вызов функции Quadrate для квадрата
% с вершинами (Xc',Yc'), (Xb,Yc'), (Xc,Yc),(Xc',Yc')
```

```
    Quadrate(x1+2*dx,y1+2*dy,...
                      x2,y1+2*dy,...
                      x2,y3,...
                      x1+2*dx,y3,...
                      n,Lmax);

   % вызов функции Quadrate для квадрата
   % с вершинами (Xd',Yd'); (Xc',Yc'),(Xc,Yc),(Xd',Yc')
   Quadrate(x1+dx,y1+2*dy,...
                     x1+2*dx,y1+2*dy,...
                     x1+2*dx,y4,...
                     x1+dx,y4,...
                     n,Lmax);

   % вызов функции Quadrate для квадрата
   % с вершинами (Xa,Yd'),(Xd',Yd'),(Xd',Yc),(Xd,Yd)
   Quadrate(x1,y1+2*dy,...
                     x1+dx,y1+2*dy,...
                     x1+dx,y4,...
                     x1,y4,...
                     n,Lmax);

   % вызов функции Quadrate для квадрата
   % с вершинами (Xa,Ya'),(Xa',Yd'),(Xd',Yd'),(Xa,Yd')
   Quadrate(x1,y1+dy,...
                     x1+dx,y1+dy,...
                     x1+dx,y1+2*dy,...
                     x1,y1+2*dy,...
                     n,Lmax);
end;
```

Рис. 17.5. Построение кривой Коха

Еще одним примером фрактального объекта, для построения которого оказывается удобным использовать рекурсивный алгоритм, является кривая Коха. Построение данной кривой начинается с отрезка K_0 единичной длины. Удалим из отрезка K_0 отрезок длиной $1/3$ и добавим два новых отрезка такой же длины, как показано на рис. 17.5. Назовем полученное множество K_1. На следующем шаге разделим каждый отрезок длиной $1/3$ на три части длиной $1/9$ и повторим описанную процедуру, заменяя на каждом шаге среднюю ветвь двумя новыми отрезками. Обозначим через K_n фигуру,

получившуюся после n-го шага. Можно строго доказать [6], что последовательность кривых $\{K_n\}_{n=1}^{\infty}$ сходится к предельной кривой K бесконечной длины, фрактальная размерность которой равна

$$d = \frac{\log(4)}{\log(3)} \approx 1,2618$$

Ниже приводится листинг рекурсивной функции `Koch.m`, возвращающей изображение кривой Коха.

```
function z=Koch(N)
% функция, возвращающая изображение кривой Коха
% N - порядок кривой Коха

% задание координат левой точки
% начального отрезка
x1=0;
y1=0;

% задание координат правой точки
% начального отрезка
x2=1;
y2=0;

% инициализация графического окна
figure(1);
axis([0 1 0 1]);
set(gca,'XColor','w','YColor','w');
hold on;
set(gca,'xtick',[],'ytick',[]);
set(gca,'XColor','w','YColor','w')

Coord(x1,y1,x2,y2,N); % вызов рекурсивной функции,
                      % прорисовывающей кривую Коха

function z=Coord(x1,y1,x2,y2,n)
% рекурсивная функция,
% прорисовывающая кривую Коха

if n>0
   % вычисление координат концов отрезков на
   % очередном шаге рекурсии
   dx=(x2-x1)/3;
   dy=(y2-y1)/3;
   x1n=x1+dx;
   y1n=y1+dy;
   x2n=x1+2*dx;
   y2n=y1+2*dy;
```

```
   xmid=dx/2-dy*sin(pi/3)+x1n;
   ymid=dy/2+dx*sin(pi/3)+y1n;

% рекурсия
   Coord(x1,y1,x1n,y1n,n-1);
   Coord(x1n,y1n,xmid,ymid,n-1);
   Coord(xmid,ymid,x2n,y2n,n-1);
   Coord(x2n,y2n,x2,y2,n-1);
else
   r1=[x1 y1];
   r2=[x2 y2];
   R=cat(1,r1,r2);
   % визуализация кривой Коха
   plot(R(:,1),R(:,2),'Color','b');
end;
```

Изображение кривой Коха пятого порядка K_5, возвращенное описанной выше функцией, представлено на рис. 17.6.

Рис. 17.6. Кривая Коха пятого порядка

17.4. L-СИСТЕМЫ И ТЕРЛ-ГРАФИКА

Понятие «L-система» было введено А. Лидермайером в 1968 г. при изучении формальных языков. С помощью L-систем оказывается возможным строить как многие известные самоподобные фракталы, например снежинку Коха, ковер Серпинского, кривые Пеано, Гильберта, Серпинского и др., так и создавать бесконечное разнообразие новых фракталов, укладывающихся в данную схему. Изложение понятий «L-система» и «терл-графика» в данном разделе следует [6].

Терл-графика (от turtle — черепашка) является подсистемой вывода графического представления фрактального объекта. Основным исполнителем данной системы является черепашка (точка), которая перемещается по экрану дискретными шаги, прочерчивая или не прочерчивая свой след. «Мгновенное» положение черпашки задается тремя параметрами (x, y, α), где (x, y) — координаты черепашки, α — направление следующего шага (угол, отсчитываемый от положительного направления оси OX). Последовательность команд, определяющая направление перемещения и действия черепашки, задается кодовым словом, буквы которого читаются слева направо. Кодовое слово, представляющее собой результат работы L-системы, может включать в себя следующие буквы:

- F — переместиться на один шаг вперед, прорисовывая след;
- b — переместиться на один шаг вперед, не прорисовывая след;
- [— открыть ветвь;
-] — закрыть ветвь;
- + — увеличить угол α на величину Θ;
- − — уменьшить угол α на величину Θ;
- X, Y — вспомогательные переменные.

Размер шага и величина приращения по углу Θ задаются изначально и остаются неизменными для всех перемещений черепашки.

Формально, детерминированная L-система состоит из алфавита, слова инициализации, называемого аксиомой или инициатором, и набора порождающих правил, указывающих, как следует преобразовать слово при каждой следующей итерации. Например, L-система, соответствующая снежинке Коха, представленной на рис. 17.7, задается следующим образом:

Рис. 17.7. Снежинка Коха

Аксиома: $F++F++F$
Порождающее правило: $Newf = F-F++F-F$
$\Theta = \pi/3$.

Как очевидно, графическим представлением аксиомы $F++F++F$ является равносторонний треугольник. Черепашка делает один шаг вперед, затем угол увеличивается на $2\pi/3$ и черепашка делает еще один шаг вперед, далее угол вновь увеличивается на $2\pi/3$ и черепашка делает еще один шаг.

На первом шаге каждая буква F в слове-инициаторе заменяется на слово *Newf*:

$$(F-F++F-F)++(F-F++F-F)++(F-F++F-F),$$

из которого, опустив скобки, получаем:

$$F-F++F-F++F--F++F-F++F-F++F-F.$$

Повторяя этот процесс, на втором шаге получим

$$\begin{array}{l} F-F++F-F-F-F++F-F++ \\ F-F++F-F-F-F++F-F++ \\ F-F++F-F-F-F++F-F++ \\ F-F++F-F-F-F++F-F++ \\ F-F++F-F-F-F++F-F. \end{array}$$

В MATLAB наиболее просто реализовать L-систему, используя рекурсивную функцию. Ниже приведен пример такой функции `RuleKoch.m`, возвращающей снежинку Коха, представленную на рис. 17.7.

```
function z=RuleKoch(Lmax,Axiom,Newf,n,tmp);
% функция, возвращающая
% L-систему для снежинки Коха
% Lmax - порядок снежинки
% Axiom - строка, содержащая аксиому
% Newf - строка, содержащая
%        порождающее правило
% tmp - входная L-система
while n<=Lmax
   if n==1
      tmp=Axiom; n=n+1;
   else
       % замена всех букв F на порождающее
       % правило
       tmp=strrep(tmp,'F',Newf);
       n=n+1;
       % рекурсия
       tmp=RuleKoch(Lmax,Axiom,Newf,n,tmp);
   end;
end;
z=tmp;
```

Для вывода изображения снежинки Коха можно использовать следующую m-функцию, реализующую описанный выше алгоритм терл-графики:

```
% листинг функции Koch1
function [X,Y]=Koch1(Lmax)
% функция, возвращающая
% изображение кривой Коха

% задание аксиомы
Axiom='F++F++F';

% порождающие правила
Newf='F-F++F-F';
teta=pi/3;
alpha=0;

% начальная точка кривой Коха
p=[0;0];

% вычисление координат кривой Коха
p=Coord(Lmax,Axiom,Newf,alpha,teta,p);

% визуализация кривой Коха
M=size(p,2);
X=p(1:1,1:M);
Y=p(2:2,1:M);
figure(1);
axis([0 1 0 1]);
plot(X,Y,'Color','b');
set(gca,'xtick',[],'ytick',[]);
set(gca,'XColor','w','YColor','w');

function z=Coord(Lmax,Axiom,Newf,alpha,teta,p)
% функция, возвращающая значения
% координат кривой Коха

 % инициализация порождающего правила
 Rule= RuleKoch(Lmax,Axiom,Newf,1,' ');

 % вычисление координат кривой Коха
 % в соответствии с порождающим правилом
M=length(Rule);
  for i=1:M
      Tmp=p(1:2,size(p,2):size(p,2));
      if Rule(i)=='F'
        R=[cos(alpha);sin(alpha)];
        R=R/(4^Lmax);
        Tmp=Tmp+R;
```

```
        p=cat(2,p,Tmp);
      end;
      if Rule(i)=='+'
        alpha=alpha+teta;
      end;
      if Rule(i)=='-'
        alpha=alpha-teta;
      end;
end;
z=p;
```

Для построения снежинки Коха 5-го порядка с помощью терл-графики следует выполнить команду

```
>> Koch1(5)
```

Однако при построении других фракталов (например, представленного на рис. 17.8 дракона Хартера—Хейтуэя) на некоторых шагах возникает необходимость в изменении направления чтения правила (не слева направо, а справа налево).

Для решения данной проблемы вводят две дополнительные команды, обозначаемые X и Y, которые используются для создания соответствующей L-системы, но игнорируются черепашкой при перемещении. При использовании этих команд порождающее правило для дракона записывается в виде:

Аксиома: FX
Порождающие правила:
$Newf = FX$
$Newx = X + YF+$
$Newy = -FX - Y$
в соответствии с данными правилами L-система имеет следующий вид:
1-й шаг: $FX + YF+$
2-й шаг: $FX + YF + + \ - FX - YF+$
3-й шаг: $FX + YF + + - FX - YF + + \ - FX + YF + - - FX - YF+$
4-й шаг:

$$FX + YF + + \ - FX - YF + + \ - FX + YF + - - FX - YF + +$$
$$-FX + YF + + - FX - YF + - - FX + YF + - - FX - YF + \ .$$

Ниже приводится листинг файла `Dracon.m`, содержащего описание функции, возвращающей изображение дракона, в соответствии с описанной выше L-системой.

```
function [X,Y]=Dracon(Lmax)
% функция, возвращающая
% изображение дракона
% Lmax - порядок дракона

% порождающее правила
Axiom='FX';
```

Рис. 17.8. Дракон Хартера–Хейтуэя

```
Newf='F';
Newx='X+YF+';
Newy='-FX-Y';
teta=pi/2;
alpha=0;

% задание координаты стартовой точки
p=[0;0];

% обращение к функции, возвращающей
% координаты угловых точек дракона
p=Coord(p,Lmax,Axiom,Newf,Newx,Newy,...
                alpha,teta);

% визуализация дракона
M=size(p,2);
```

```
X=p(1:1,1:M); % создание вектора, содержащего
              % X-е координаты угловых точек
              % дракона
Y=p(2:2,1:M); % создание вектора, содержащего
              % Y-е координаты угловых точек
              % дракона
figure(1);
plot(X,Y,'Color','k');
set(gca,'xtick',[],'ytick',[]);
set(gca,'XColor','w','YColor','w');

function z=Coord(p,Lmax,...
                 Axiom,Newf,Newx,Newy,alpha,teta)
% функция, возвращающая координаты
% угловых точек дракона

% задание L-системы
Rule=DraconString(Lmax,...
                  Axiom,Newf,Newx,Newy,1,'');

% вычисление координат угловых точек дракона
M=length(Rule);
for i=1:M
   Tmp=p(1:2,size(p,2):size(p,2));
    if Rule(i)=='F' % шаг вперед
       R=[cos(alpha);sin(alpha)];
       R=R/(2^Lmax);
       Tmp=Tmp+R;
       p=cat(2,p,Tmp);
    end;
    if Rule(i)=='+' % увеличение угла, задающего
                    % направление движения
       alpha=alpha+teta;
    end;
    if Rule(i)=='-' % уменьшение угла, задающего
                    % направление движения
       alpha=alpha-teta;
    end;
 end;
z=p;

function z=DraconString(Lmax,...
                        Axiom,Newf,Newx,Newy,...
                        n,tmp);
% функция, возвращающая L-систему
```

```
a=1;
if n<=Lmax
  if n==1
    tmp=Axiom;
  end;
  M=length(tmp);
  tmp1='';
  for i=1:M
    if tmp(i)=='F'
      tmp1=strcat(tmp1,Newf);
    end;
    if tmp(i)=='X'
      tmp1=strcat(tmp1,Newx);
    end;
    if tmp(i)=='Y'
      tmp1=strcat(tmp1,Newy);
    end;
    if not(tmp(i)=='X') & not(tmp(i)=='Y') & not(tmp(i)=='F')
      tmp1=strcat(tmp1,tmp(i));
    end;
  end;
  tmp=tmp1;
  n=n+1;
  tmp=DraconString(Lmax,Axiom,Newf,Newx,Newy,n,tmp);
end;
z=tmp;
```

Далее следует выполнить в командной строке MATLAB команду:

```
>> Dracon(12);
```

Результат ее выполнения представлен на рис. 17.8.

Заменяя в описанной функции порождающие правила, можно получить и другие фрактальные кривые, например кривую Гильберта (рис. 17.9):

Рис. 17.9. Кривая Гильберта

Аксиома: X
Порождающие правила:
$Newf = F$
$Newx = -YF + XFX + FY-$
$Newy = +XF - YFY - FX+$
$\alpha = 0$
$\Theta = \pi/2.$
Кривая Госпера (рис. 17.10):

Рис. 17.10. Кривая Госпера

Аксиома: XF
Порождающие правила:
$Newf = F$
$Newx = X + YF + +YF - FX - -FXFX - YF+$
$Newy = -FX + YFYF + +YF + FX - -FX - Y$
$\alpha = 0$
$\Theta = \pi/3.$

Кривая Серпинского (рис. 17.11):

Рис. 17.11. Кривая Серпинского

Аксиома: $F + XF + F + XF$
Порождающие правила:
$Newf = F$
$Newx = XF - F + F - XF + F + XF - F + F - X$
$Newy =''$
$\alpha = \pi/4$
$\Theta = \pi/2.$
Кривая Остров (рис. 17.12):

Рис. 17.12. Кривая Остров 2-го порядка

Аксиома: $F + F + F + F$
Порождающие правила: $Newf = F + F - F - FFF + F + F - F'$
$Newx =''$
$Newy =''$
$\alpha = 0$
$\Theta = \pi/2.$

***Задача* 17.1**

Модифицируйте функцию `Dracon.m` *и постройте кривую Гильберта.*

***Задача* 17.2**

Модифицируйте функцию `Dracon.m` *и постройте кривую Госпера.*

***Задача* 17.3**

Модифицируйте функцию `Dracon.m` *и постройте кривую Серпинского.*

***Задача* 17.4**

Модифицируйте функцию `Dracon.m` *и постройте кривую Остров.*

В заключение остановимся на операции ветвления. Когда в L-системе встречается символ [— открыть ветвь, необходимо запомнить координаты точки нахождения черепашки и направление ее движения, то есть переменные (x, y, α). К сохраненным переменным следует вернуться после обнаружения символа] — закрыть ветвь. Для хранения триплетов (x, y, α) в [6] предлагается использовать стек:

$$\begin{bmatrix} x_1 & y_1 & \alpha_1 \\ x_2 & y_2 & \alpha_2 \\ \vdots & \vdots & \vdots \\ x_n & y_n & \alpha_n \end{bmatrix},$$

в конец которого записываются новые данные.

При закрытии ветви, переменным (x, y, α) присваиваются значения, считанные из конца стека, затем эти значения из стека удаляются. В MATLAB оказывается более удобным использовать матрицу с переменным числом столбцов:

$$M = \begin{bmatrix} x_1 & x_2 & \dots & x_n \\ y_1 & y_2 & \dots & y_n \\ \alpha_1 & \alpha_2 & \dots & \alpha_n \end{bmatrix},$$

причем координаты каждой новой точки ветвления добавляются в новый столбец матрицы M.

После закрытия ветви, переменным (x, y, α) присваиваются значения, считанные из последнего столбца матрицы M, затем этот столбец удаляется.

Таким образом, процедура ветвления реализуется следующими действиями:

[— открыть ветвь: добавить вектор

$$\begin{bmatrix} x \\ y \\ \alpha \end{bmatrix}$$

новым столбцом матрицы M,

] — закрыть ветвь: присвоить значения переменным (x, y, α) координаты вектора, являющегося последним столбцом матрицы M.

Ниже приводится листинг файла `Flower.m`, содержащего описание функции, возвращающей изображение цветка, в соответствии с описанной выше L-системой.

```
function [X,Y]=Flower(Lmax)
% функция, возвращающая изображение цветка
% Lmax - порядок цветка

% порождающие правила
Axiom='F[+F+F][-F-F][++F][--F]F';
Newf='FF[++F][+F][F][-F][--F]';
teta=pi/16;
alpha=pi/2;
p=[0;0];

% обращение к функции,
% возвращающей изображение цветка
Coord(p,Lmax,Axiom,Newf,alpha,teta);

function z=Coord(p,Lmax,Axiom,Newf,alpha,teta)
% функция, возвращающая изображение
% цветка

   % задание L-системы
   Rule=FlowerString(Lmax,Axiom,Newf,1,'');
   figure(1);
   hold on;
   M=length(Rule);
   L=0;
   x0=p(1);
   y0=p(2);
   for i=1:M
     if Rule(i)=='F'
        x1=x0+cos(alpha);
        y1=y0+sin(alpha);
        X=[x0,x1];
        Y=[y0,y1];
        x0=x1;
        y0=y1;
        plot(X,Y,'Color','g');
      end;
     if Rule(i)=='+'
        % увеличение угла, задающего
        % направление движения
        alpha=alpha+teta;
      end;
      if Rule(i)=='-'
        % уменьшение угла, задающего
        % направление движения
```

```
      alpha=alpha-teta;
      end;
      if Rule(i)=='['; % открыть ветвь
       if L==0;
         St=[x0;y0;alpha];
         L=1;
       else
         St=cat(2,St,[x0;y0;alpha]);
       end;
      end;
      if Rule(i)==']' % закрыть ветвь
        M=size(St,2);
        R=St(1:3,M:M);
        x0=R(1);
        y0=R(2);
        alpha=R(3);
        St=St(1:3,1:M-1);
      end;
    end;
    set(gca,'xtick',[],'ytick',[]);
    set(gca,'XColor','w','YColor','w');
    hold off

function z=FlowerString(Lmax,Axiom,Newf,n,tmp);
% функция, возвращающая L-систему
while n<=Lmax
  if n==1
    tmp=Axiom;
    n=n+1;
  else
    tmp=strrep(tmp,'F',Newf);
    n=n+1;
    % рекурсия
    tmp=FlowerString(Lmax,Axiom,Newf,n,tmp);
  end;
end;
z=tmp;
```

Пример фрактала третьего порядка, построенного с помощью функции `Flower.m` представлен на рис. 17.13.

Аксиома: F

Порождающие правила:

$Newf = -F + F + [+F - F-] - [-F + F + F]$

$\alpha = \pi/2$

$\Theta = \pi/2.$

Рис. 17.13. Фрактал Цветок

Заменяя в описанной программе порождающие правила, можно получить и другие ветвящиеся фрактальные объекты, например фрактал Куст (рис. 17.14):

Рис. 17.14. Фрактал Куст

Аксиома: $[F] + [F] + [F] + [F] + [F] + [F]$
Порождающие правила:
$New f = F[++F][-FF]FF[+F][-F]FF$
$\alpha = 0$
$\Theta = pi/3.$

Фрактал Снежинка (рис. 17.15):

Рис. 17.15. Фрактал Снежинка

и др.

***Задача* 17.5**

Модифицируйте функцию `Flower.m` *и постройте фрактал Куст.*

***Задача* 17.6**

Модифицируйте функцию `Flower.m` *и постройте кривую Снежинка.*

17.5. СИСТЕМЫ ИТЕРИРОВАННЫХ ФУНКЦИЙ

Напомним, следуя [6], что в общем случае для построения системы итерированных функций (СИФ) в рассмотрение вводится совокупность сжимающих отображений:

$$T_1, \ s_1,$$

$$T_2, \ s_2,$$

$$\vdots$$

$$T_m, \ s_m,$$

действующих в $\mathbf{R}^n$.

Эти m отображений используются для построения одного сжимающего отображения T в пространстве Ω всех непустых компактов из $\mathbf{R}^n$.

Здесь преобразование Хатчинсона $T : \Omega \to \Omega$ определяется следующим образом:

$$T(E) = T_1(E) \cup T_2(E) \cup \ldots \cup T_m(E), E \in \Omega.$$

Данное преобразование ставит в соответствии «точкам» из Ω, под которыми здесь понимаются компактные множества, также «точки» из Ω.

Системой итерированных функций называется совокупность определенных выше отображений вместе с итерационной схемой:

$$E_1 = T(E_0), E_2 = T(E_1), \ldots, E_n = T(E_{n-1}), \ldots,$$

где E_0 — произвольное компактное множество.

Существование предельного множества E:

$$E = \lim_{n\to\infty} E_n$$

системы итерированных функций в смысле сходимости в метрике Хаусфдорфа:

$$H(E, F) = \min\{\varepsilon > 0 : E \subset F + \varepsilon \;\&\; F \subset E + \varepsilon\},$$

где E и F — непустые компактные подмножества в $\mathbf{R}^n$, доказывает соответствующая теорема [6].

Например, СИФ при построении ковра Серпинского (см. рис.17.2), задается тремя аффинными преобразованиями, которые в матричной форме имеют следующий вид:

$$T_1\left(\begin{bmatrix} x_1 \\ x_2 \end{bmatrix}\right) = \begin{bmatrix} 1/2 & 0 \\ 0 & 1/2 \end{bmatrix}\begin{bmatrix} x_1 \\ x_2 \end{bmatrix} + \begin{bmatrix} 0 \\ 0 \end{bmatrix},$$

$$T_2\left(\begin{bmatrix} x_1 \\ x_2 \end{bmatrix}\right) = \begin{bmatrix} 1/2 & 0 \\ 0 & 1/2 \end{bmatrix}\begin{bmatrix} x_1 \\ x_2 \end{bmatrix} + \begin{bmatrix} 1/2 \\ 0 \end{bmatrix},$$

$$T_3\left(\begin{bmatrix} x_1 \\ x_2 \end{bmatrix}\right) = \begin{bmatrix} 1/2 & 0 \\ 0 & 1/2 \end{bmatrix}\begin{bmatrix} x_1 \\ x_2 \end{bmatrix} + \begin{bmatrix} 1/4 \\ \sqrt{3}/4 \end{bmatrix}.$$

Различают два подхода к реализации СИФ:

1) детерминированный (ДСИФ) (в котором аффинные преобразования применяются последовательно к каждой точке начальной конфигурации);

2) рандомизированный (РСИФ) (в котором случайно выбираемые аффинные преобразования применяют к единственной начальной точке).

Известный детерминированный алгоритм вычисления СИФ, ориентированный на реализацию в виде компьютерной программы в каком-либо языке программирования, допускающем компиляцию, описан в [6].

К недостаткам данного алгоритма можно отнести:

1. Зависимость качества изображения от размера графического окна (удовлетворительное качество изображения достигается для $m \geq 256$).

2. Привязка алгоритма к размеру графического окна, и, как следствие, большой объем вычислений (число операций прямо пропорционально числу точек m^2 и числу итераций).

3. Возникновение аварийных остановок программы с сообщением об ошибке «индекс вышел за пределы» при попадании точки за пределы окна $m \times m$.

Рис. 17.16. Построение ковра Серпинского с помощью алгоритма ДСИФ

Проведем анализ модификации алгоритма ДСИФ, позволяющей реализовать его в MATLAB, на примере уже рассмотренного в первом разделе ковра Серпинского. Во-первых, заметим, что, как видно из рис. 17.16, для получения изображения ковра Серпинского необходимо на каждую точку $\left(x_i^{(0)}, y_i^{(0)}\right)$, находящуюся внутри исходного треугольника S_0, раздельно подействовать каждым из аффинных преобразований T_1, T_2, T_3:

$$T_1 \begin{bmatrix} x_i^{(0)} \\ y_i^{(0)} \end{bmatrix} = \begin{bmatrix} x_{i1}^{(1)} \\ y_{i1}^{(1)} \end{bmatrix},$$

$$T_2 \begin{bmatrix} x_i^{(0)} \\ y_i^{(0)} \end{bmatrix} = \begin{bmatrix} x_{i2}^{(1)} \\ y_{i2}^{(1)} \end{bmatrix},$$

$$T_3 \begin{bmatrix} x_i^{(0)} \\ y_i^{(0)} \end{bmatrix} = \begin{bmatrix} x_{i3}^{(1)} \\ y_{i3}^{(1)} \end{bmatrix}.$$

Таким образом, каждая точка $\left(x_i^{(0)}, y_i^{(0)}\right)$ на первом шаге итерации порождает три новые точки: $\left(x_{i1}^{(0)}, y_{i1}^{(0)}\right)$, $\left(x_{i2}^{(0)}, y_{i2}^{(0)}\right)$, $\left(x_{i3}^{(0)}, y_{i3}^{(0)}\right)$, на каждую из этих точек на втором шаге итерации вновь следует подействовать аффинными преобразованиями T_1, T_2, T_3.

В результате каждая из трех точек ковра S_1 вновь породит три точки ковра S_2 и т. д. Описанный процесс удобно изобразить в виде следующего графа (рис. 17.17).

Рис. 17.17. Построение ковра Серпинского с помощью алгоритма ДСИФ

Из рис. 17.17. видно, что на втором шаге итерации существует 9 ($3^n = 9$, где $n = 2$) правил, по которым каждой начальной точке $\left(x_i^{(0)}, y_i^{(0)}\right)$ ставятся в соответствии 9 точек ковра Серпинского S_1.

Соответственно, на третьем шаге итерации таких правил будет $3^3 = 27$:

$$\begin{array}{lll}
T_1T_1T_1 & T_1T_2T_1 & T_1T_3T_1 \\
T_1T_1T_2 & T_1T_2T_2 & T_1T_3T_2 \\
T_1T_1T_3 & T_1T_2T_3 & T_1T_3T_3 \\
T_2T_1T_1 & T_2T_2T_1 & T_2T_3T_1 \\
T_2T_1T_2 & T_2T_2T_2 & T_2T_3T_2 \\
T_2T_1T_3 & T_2T_2T_3 & T_2T_3T_3 \\
T_3T_1T_1 & T_3T_2T_1 & T_3T_3T_1 \\
T_3T_1T_2 & T_3T_2T_2 & T_3T_3T_2 \\
T_3T_1T_3 & T_3T_2T_3 & T_3T_3T_3.
\end{array}$$

Таким образом, для построения ковра Серпинского n-го уровня с помощью ДСИФ необходимо научиться генерировать 3^n правил, по которым каждой точке $\left(x_i^{(0)}, y_i^{(0)}\right)$ ставится в соответствии 3^n точек $\left(x_{ij}^{(n)}, y_{ij}^{(n)}\right)$, $j = 1, 2, \ldots, 3^n$.

Введем обозначения: $T_1 \Leftrightarrow 0, T_2 \Leftrightarrow 1, T_2 \Leftrightarrow 2$. В выбранных обозначениях правила преобразования на третьем шаге итерации имеют вид

$$\begin{array}{lll}
000 & 010 & 020 \\
001 & 011 & 021 \\
002 & 012 & 022 \\
100 & 110 & 120 \\
101 & 111 & 121 \\
102 & 112 & 122 \\
200 & 210 & 220 \\
201 & 211 & 221 \\
202 & 212 & 222.
\end{array}$$

Анализ таблицы закодированных правил преобразований показывает, что названия правил являются не чем иным, как множеством натуральных чисел $1, 2, \ldots, 27$, записанных в троичной системе счисления. При этом для представления кода каждого правила используется число цифр, совпадающее с порядком ковра n.

Соответственно, для случая $n = 4$ имеем множество, состоящее из $3^4 = 81$ правила, названия которых есть множество чисел $1, 2, \ldots, 81$, записанных в троичной системе счисления, при этом для представления каждого числа используются 4 цифры. Очевидно, что для хранения названия правил наиболее удобно использовать массив строковых переменных длиной n, число элементов которого равно $3^4 = 81$.

Таким образом, для построения ковра Серпинского в MATLAB с помощью ДСИФ можно использовать следующий алгоритм:

1. Задать порядок ковра Серпинского n.

2. Задать число точек начальной конфигурации m.

3. Задать координаты i точек ($i = 1, 2, \ldots, m$), заполняющих начальное множество.

4. Перевести каждое из чисел $1, 2, \ldots, 3^n$ в троичную систему счисления.

5. Сформировать массив, состоящий из 3^n строк, длиной n символов.

6. Задать аффинные преобразования.

7. Для i-й точки начальной конфигурации последовательно применить каждое из $j = 1, 2, \ldots, 3^n$ итерационных правил и отобразить в графическом окне полученные образы каждой начальной точки.

Ниже приводится листинг файла `SerpDSIF.m`, содержащего описание соответствующей функции, возвращающей изображение ковра Серпинского.

```
% листинг файла SerpDSIF.m
function z=SerpDSIF(Niter,NPoints)
% функция, возвращающая
% изображение ковра Серпинского,
% создаваемого с помощью алгоритма DSIF
% Niter - порядок ковра
% NPoints - число точек начальной конфигурации

x=zeros(NPoints,1);
y=zeros(NPoints,1);
% задание координат точек начальной конфигурации
x1=0;
y1=0;
x2=1;
y2=0;
x3=1/2;
y3=sin(pi/3);
j=1;
while j<=NPoints
   tmpx=rand(1,1);
   tmpy=sqrt(3)/2*rand(1,1);
   if (-sqrt(3)*tmpx+tmpy<=0)&(sqrt(3)*tmpx+tmpy-sqrt(3)<=0)
     x(j)=tmpx;
     y(j)=tmpy;
     j=j+1;
   end;
end;
% Формирование массива, содержащего правила итерации
for i=1:3^Niter
   Tmp(i)=system3(i-1); % перевод числа из десятичной
                        % в троичную систему счисления
end;
n=1;s='0';
while n<Niter
```

```
   s=strcat(s,'0'); n=n+1;
end;
for i=1:3^Niter
   tmp=num2str(Tmp(i));tmp1=s;
   for m=1:length(tmp)
      tmp1(Niter-m+1)=tmp(length(tmp)-m+1);
   end;
   Cod(i,1:Niter)=tmp1;
end;
% задание аффинных преобразований
a1=[0;0];
a2=[1/2;0];
a3=[1/4;sqrt(3)/4];
A=[1/2,0;0,1/2];

% инициализация графического окна
figure(1);
hold on;
set(gca,'xtick',[],'ytick',[]);
set(gca,'XColor','w','YColor','w');
fill([x1 x2 x3],[y1 y2 y3],'w');
% вызов функции, осуществляющей
% построение ковра Серпинского
GosperDraw(Niter,NPoints,x,y,A,a1,a2,a3,Cod);

function z=GosperDraw(Niter,NPoints,x,y,...
                      A,a1,a2,a3,Cod)
% функция, создающая изображение ковра Серпинского
for m=1:3^Niter
   X=x; Y=y; Rule=Cod(m,:);
   for i=1:Niter
      tmp=Rule(Niter+1-i);
      if tmp=='0' % первое аффинное преобразование
          [X Y]=T(NPoints,X,Y,A,a1);
      end;
      if tmp=='1' % второе аффинное преобразование
         [X Y]=T(NPoints,X,Y,A,a2);
      end;
      if tmp=='2' % третье аффинное преобразование
         [X Y]=T(NPoints,X,Y,A,a3);
      end;
   end;
   % отображение результатов итерации
   plot(X,Y,'.','MarkerSize',1,'MarkerEdgeColor','k');
end;
```

```
function [X,Y]=T(NPoints,x,y,A,a)
% функция, возвращающая
% результат аффинного преобразования
X=zeros(NPoints,1); Y=zeros(NPoints,1);
for i=1:NPoints
    R=[x(i);y(i)];
    R=A*R+a;
    X(i)=R(1);
    Y(i)=R(2);
end;

function z=system3(D);
% функция, возвращающая значение
% целого числа в троичной системе счисления
% D - число в десятичной системе счисления
n=1;
while D>=3^n
     n=n+1;
end;
if n>1
   a=floor(D/3^(n-1))*10^(n-1); b=mod(D,3^(n-1));
   if b>=3
      b=system3(b); % рекурсия
    end;
   z=a+b;
else
   z=D;
end;
```

Результат выполнения команды `>> SerpDSIF(7,100)` представлен на рис. 17.18.

Рис. 17.18. Ковер Серпинского, построенный с помощью ДСИФ

Ниже приводятся листинги функций, осуществляющих построение других фракталов с помощью алгоритма ДСИФ.

```
% листинг функции MapleS.m
function z=MapleS(Niter,NPoints)
% функция, возвращающая
% изображение фрактала Лист,
% создаваемого с помощью алгоритма
% DSIF
% Niter - порядок ковра
% NPoints - число точек начальной конфигурации

% задание координат начальной конфигурации
x1=0;
y1=0;
x2=1;
y2=0;
x3=1/2;
y3=sin(pi/3);
x=zeros(NPoints,1);
y=zeros(NPoints,1);
j=1;
while j<=NPoints
  tmpx=rand(1,1);
  tmpy=sqrt(3)/2*rand(1,1);
  if (-sqrt(3)*tmpx+tmpy<=0)&(sqrt(3)*tmpx+tmpy-sqrt(3)<=0)
    x(j)=tmpx;
    y(j)=tmpy;
    j=j+1;
  end;
end;

% формирование массива, содержащего
% правила итерации
for i=1:2^Niter
     Tmp(i)=system2(i-1,2);
end;
n=1;
s='0';
while n<Niter
    s=strcat(s,'0');
    n=n+1;
end;
for i=1:2^Niter
      tmp=num2str(Tmp(i));
      tmp1=s;
      for m=1:length(tmp)
            tmp1(Niter-m+1)=tmp(length(tmp)-m+1);
```

```
      end;
      Cod(i,1:Niter)=tmp1;
end;

% задание аффинных преобразований
A1=[0.4,-0.3733;0.0600,0.6000];
A2=[-0.8000,-0.1867;0.1371,0.8000];
a1=[0.3533;0.000];
a2=[1.1000;0.1000];

% инициализация графического окна
figure(1);
hold on;
set(gca,'xtick',[],'ytick',[]);
set(gca,'XColor','w','YColor','w');
% обращение к функции Simplex,
% возвращающей изображение фрактала
Simplex(Niter,NPoints,x,y,A1,A2,a1,a2,Cod);
hold off

function z=Simplex(Niter,NPoints,x,y,A1,A2,a1,a2,Cod)
% функция, возвращающая изображение фрактала
for m=1:2^Niter
      X=x;
      Y=y;
      Rule=Cod(m,:);
      for i=1:Niter
            tmp=Rule(Niter+1-i);
            if tmp=='0'
               [X Y]=T(NPoints,X,Y,A1,a1);
            end;
            if tmp=='1'
               [X Y]=T(NPoints,X,Y,A2,a2);
            end;
      end;
     plot(X,Y,'.','MarkerSize',1,'MarkerEdgeColor','g');
end;

function [X,Y]=T(NPoints,x,y,A,a)
% функция, реализующая аффинные
% пребразования координат
X=zeros(NPoints,1);
Y=zeros(NPoints,1);
for i=1:NPoints
      R=[x(i);y(i)];
```

```
      R=A*R+a;
      X(i)=R(1);
      Y(i)=R(2);
end;

function z=system2(D,m);
% функция, реализующая преобразование
% числа D в систему счисления с основанием m
n=1;
while D>=m^n
  n=n+1;
end;
if n>1
  a=floor(D/m^(n-1))*10^(n-1);
  b=mod(D,m^(n-1));
  if b>=m
    b=system2(b,m);
  end;
  z=a+b;
else
  z=D;
end;
```

Результат выполнения команды `>> MapleS(11,100)` представлен на рис. 17.19.

Рис. 17.19. Фрактал Лист

```
% листинг файла Paporotnic.m
function z=Paporotnic(Niter,NPoints)
% функция, возвращающая
% изображение фрактала Лист,
% создаваемого с помощью алгоритма DSIF
% Niter - порядок ковра
% NPoints - число точек начальной конфигурации

% задание координат начальной конфигурации
x1=0;
y1=0;
x2=1;
y2=0;
x3=1/2;
y3=sin(pi/3);
x=zeros(NPoints,1);
y=zeros(NPoints,1);
j=1;
while j<=NPoints
  tmpx=rand(1,1);
  tmpy=sqrt(3)/2*rand(1,1);
  if (-sqrt(3)*tmpx+tmpy<=0)&(sqrt(3)*tmpx+tmpy-sqrt(3)<=0)
    x(j)=tmpx;
    y(j)=tmpy;
    j=j+1;
  end;
end;

% формирование массива, содержащего правила итерации
for i=1:4^Niter
     Tmp(i)=system2(i-1,4);
end;
n=1;
s='0';
while n<Niter
    s=strcat(s,'0');
    n=n+1;
end;
for i=1:4^Niter
    tmp=num2str(Tmp(i));
    tmp1=s;
    for m=1:length(tmp)
        tmp1(Niter-m+1)=tmp(length(tmp)-m+1);
    end;
    Cod(i,1:Niter)=tmp1;
```

```
end;

% задание аффинных преобразований

A1=[0.7000,0;0,0.7000];
A2=[0.1000,-0.4330;0.1732,0.2500];
A3=[0.1000,0.4330;-0.1732,0.2500];
A4=[0,0;0,0.3000];
a1=[0.1496;0.2962];
a2=[0.4478;0.0014];
a3=[0.4450;0.1559];
a4=[0.4987;0.0070];

% инициализация графического окна
figure(1);
hold on;
set(gca,'xtick',[],'ytick',[]);
set(gca,'XColor','w','YColor','w');

% обращение к функции Simplex,
% возвращающей изображение фрактала
Simplex(Niter,NPoints,x,y,...
                A1,A2,A3,A4,...
                a1,a2,a3,a4,Cod);
hold off

function z=Simplex(Niter,NPoints,x,y,...
                            A1,A2,A3,A4,...
                            a1,a2,a3,a4,Cod)
% функция, возвращающая изображение фрактала
for m=1:4^Niter
    X=x;
    Y=y;
    Rule=Cod(m,:);
    for i=1:Niter
        tmp=Rule(Niter+1-i);
        if tmp=='0'
            [X Y]=T(NPoints,X,Y,A1,a1);
        end;
        if tmp=='1'
            [X Y]=T(NPoints,X,Y,A2,a2);
        end;
        if tmp=='2'
            [X Y]=T(NPoints,X,Y,A3,a3);
        end;
```

```
        if tmp=='3'
            [X Y]=T(NPoints,X,Y,A4,a4);
        end;
    end;
    plot(X,Y,'.','MarkerSize',1,'MarkerEdgeColor','b');
end;

function [X,Y]=T(NPoints,x,y,A,a)
% функция, реализующая аффинные
% пребразования координат
X=zeros(NPoints,1);
Y=zeros(NPoints,1);
for i=1:NPoints
    R=[x(i);y(i)];
    R=A*R+a;
    X(i)=R(1);
    Y(i)=R(2);
end;

function z=system2(D,m);
% функция,
% реализующая преобразование
% числа D в систему счисления
% с основанием D
n=1;
while D>=m^n
  n=n+1;
end;
if n>1
  a=floor(D/m^(n-1))*10^(n-1);
  b=mod(D,m^(n-1));
  if b>=m
    b=system2(b,m);
  end;
  z=a+b;
else
  z=D;
end;
```

Результат выполнения команды `>> Paporotnic(6,100)` представлен на рис. 17.20.

В отличие от ДСИФ в рандомизированном алгоритме начальное множество S_0 состоит из одной точки x_0, y_0, а правило, по которому точке x_0, y_0 ставится в соответствии точка (x_i, y_i), где i — номер правила, выбирается случайным образом из набора, содержащего все возможные правила аффинных преобразований. Например, применительно к ковру Серпинского это означает, что при построении ковра 2-го порядка

Рис. 17.20. Фрактал Папоротник

преобразование должно случайным образом выбираться из следующего множества преобразований:

$$\{T_1, T_2, T_3, T_1T_1, T_1T_2, T_1T_3, T_2T_3, T_3T_1, T_3T_2, T_3T_3\},$$

число элементов N которого, как очевидно, равно $N = \sum\limits_{k=1}^{n} m^k$.

Таким образом, для построения ковра Серпинского в MATLAB с помощью РСИФ можно использовать следующий алгоритм:

1. Задать порядок ковра Серпинского .
2. Задать число испытаний $NTrial$.
3. Задать число аффинных преобразований $m = 3$.
4. Сформировать массив, содержащий набор правил для аффинных преобразований.
5. Задать координаты начальной точки (x_0, y_0).
6. Перевести каждое из чисел $1, 2, \ldots, 3^n$ в троичную систему счисления.
7. Задать аффинные преобразования.
8. Для заданного числа испытаний последовательно, начиная с начальной точки, в соответствии с правилами аффинных преобразований, выбираемых случайным образом, вычислить точки итерационной последовательности.

9. Отобразить вычисленное множество точек в графическом окне.

В общем случае для фракталов n-го порядка, изображение которых создается с помощью m аффинных преобразований, как и при использовании ДСИФ, необходимо переводить числа $1, 2, \ldots, m^n$ в m-ю систему счисления.

Ниже приводится листинг файла `Cristal.m`, содержащего описание соответствующей функции.

```
% листинг файла Cristal.m
function z=Cristal(Niter,NTrial)
% функция, возвращающая
% изображение кристалла, создаваемого
% в соответствии с РСИФ
% Niter - порядок кристалла
% NTrial - число испытаний

% задание координаты начальной точки
x1=0;
y1=0;
x2=1;
y2=0;
x3=1/2;
y3=sin(pi/3);
j=1;
Flag=0;
while Flag==0
  tmpx=rand(1,1);
  tmpy=sqrt(3)/2*rand(1,1);
  if (-sqrt(3)*tmpx+tmpy<=0)&(sqrt(3)*tmpx+tmpy-sqrt(3)<=0)
    x=tmpx;
    y=tmpy;
    Flag=1;
  end;
end;

% создание системы порождающих правил
for m=1:Niter
  for i=1:4^m
    Tmp(k)=system3(i-1,4);
    k=k+1;
  end;
end;
Q(1)=4;
for m=2:Niter
  Q(m)=Q(m-1)+4^m;
end;
```

```
n=1;
s='0';
M=1;
while n<=length(Tmp)
  m=1;
  while n>Q(m)
    m=m+1;
  end;
  if m==1
    S(n,1:1)=s;
  else
    S(n,1:1)=s;
    for i=2:m
      S(n,1:i)=strcat(S(n,:),s);
    end;
  end;
  n=n+1;
end;
for i=1:k-1
  tmp=num2str(Tmp(i));
  m=1;
  while i>Q(m)
    m=m+1;
  end;
  tmp1(1:m)=S(i,1:m);
  for m=1:length(tmp)
    tmp1(length(tmp1)-m+1:length(tmp1)-m+1)=...
            tmp(length(tmp)-m+1:length(tmp)-m+1);
  end;
  Cod(i,1:length(tmp1))=tmp1;
end;

% задание параметров аффинных
% преобразований
A1=[0.2550,0.0000;0.0000,0.2550];
A2=[0.2550,0.0000;0.0000,0.2550];
A3=[0.2550,0.0000;0.0000,0.2550];
A4=[0.3700,-0.6420;0.6420,0.3700];
a1=[0.3726;0.6714];
a2=[0.1146;0.2232];
a3=[0.6306;0.2232];
a4=[0.6356;-0.0061];

% инициализация графического окна
figure(1);hold on;
```

```
set(gca,'xtick',[],'ytick',[]);
set(gca,'XColor','w','YColor','w');
% вызов функции, реализующей построение
% фрактала
DrawFractal(Niter,NTrial,x,y,A1,A2,A3,A4,...
                             1,a2,a3,a4,Cod);
hold off

function z=DrawFractal(Niter,NTrial,x,y,...
                           A1,A2,A3,A4,...
                           a1,a2,a3,a4,Cod)
% функция, возвращающая
% изображение фрактала
X1=zeros(NTrial,1);
Y1=zeros(NTrial,1);
X=x;
Y=y;
for m=1:NTrial
  Np=1+round((size(Cod,1)-1)*rand(1,1));
  Rule=Cod(Np,:);
  for i=1:length(Rule)
    tmp=Rule(length(Rule)+1-i);
    if tmp=='0'
      [X Y]=T(X,Y,A1,a1);
    end;
    if tmp=='1'
      [X Y]=T(X,Y,A2,a2);
    end;
    if tmp=='2'
      [X Y]=T(X,Y,A3,a3);
    end;
    if tmp=='3'
      [X Y]=T(X,Y,A4,a4);
    end;
  end;
  X1(m)=X;
  Y1(m)=Y;
end;
plot(X1,Y1,'.','MarkerSize',1,'MarkerEdgeColor','b');

function [X,Y]=T(x,y,A,a)
% функция, возвращающая результат
% аффинного преобразования
R=[x;y];
R=A*R+a;
```

```
X=R(1);
Y=R(2);

function z=system3(D,m);
% функция, возвращающая
% значение числа
% в четверичной
% системе координат
n=1;
while D>=m^n
  n=n+1;
end;
if n>1
  a=floor(D/m^(n-1))*10^(n-1);
  b=mod(D,m^(n-1));
  if b>=m
    b=system3(b,m);
  end;
  z=a+b;
else
  z=D;
end;
```

Результат выполнения команды >> `Cristal(4,500000)` представлен на рис. 17.21.

Рис. 17.21. Фрактал Кристалл

***Задача* 17.7**

Модифицируйте, описанные выше т-функции и постройте фракталы, используя следующие аффинные преобразования:

$$A_1 = \begin{bmatrix} 0,5 & 0 \\ 0 & -0,5 \end{bmatrix}, A_2 = \begin{bmatrix} 0 & -0,5 \\ -0,5 & 0 \end{bmatrix}, A_3 = \begin{bmatrix} -0,5 & 0 \\ 0 & -0,5 \end{bmatrix},$$

$$a_1 = \begin{bmatrix} 0,5 \\ 0,5 \end{bmatrix}, a_2 = \begin{bmatrix} 0,5 \\ 0,5 \end{bmatrix}, a_3 = \begin{bmatrix} 0,5 \\ 1,0 \end{bmatrix}.$$

17.6. МЕТОДЫ ОЦЕНКИ ФРАКТАЛЬНОЙ РАЗМЕРНОСТИ

Фрактальную размерность множества можно определить многими способами. Далее мы рассматриваем различные способы введения понятия «фрактальная размерность» и соответствующие алгоритмы вычисления данной размерности.

Геометрический подход. Рассмотрим равномерное распределение точек вдоль некоторой линии, или одномерного многообразия в двумерном пространстве (рис. 17.22).

Рис. 17.22. Покрытия для линейного и плоского распределения точек

Вычислим минимальное число элементарных квадратов $N(\varepsilon)$ с длиной стороны ε, покрывающих рассматриваемое множество $(N(\varepsilon)) < N_0$. Если N_0 велико, то число квадратов, покрывающих линию, будет меняться в зависимости от ε как $N(\varepsilon) \approx 1/\varepsilon$.

Аналогично, если точки распределить по двумерной поверхности, то минимальное число квадратов, покрывающих множество, будет изменяться в зависимости от ε как $N(\varepsilon) \approx 1/\varepsilon^2$.

В самом общем случае зависимость $N(\varepsilon)$ естественно определить законом

$$N(\varepsilon) \approx \frac{1}{\varepsilon^d}. \tag{17.6}$$

Прологарифмировав выражение (17.6), и введя индекс C (от английского *capacity dimension* — емкостная размерность), получим

$$d_c = \lim_{\varepsilon \to 0} \frac{\log(N(\varepsilon))}{\log(1/\varepsilon)}. \tag{17.7}$$

Отметим, что неявно в данном определении содержится требование, согласно которому число точек в множестве должно быть большим $N_0 \to \infty$.

Существует два возражения против использования (17.7) для оценки фрактальной размерности. Во-первых, емкостная размерность — геометрическая мера, поэтому она не учитывает частоту, с которой траектория посещает элемент покрытия. Во-вторых, подсчет гиперкубов, образующих покрытие множества в фазовом пространстве, требуют очень больших вычислительных затрат. Отмеченные недостатки обусловили введение других способов оценки фрактальной размерности, которые, как оказывается [8], дают близкие значения.

Ниже приведен листинг m-функции `DimMinkovsky.m`, реализующей описанный выше алгоритм вычисления фрактальной размерности двумерных объектов.

```
% листинг функции DimMinkovsky.m
function [Ksi,Nu,d]=DimMinkovsky(S,Lmax);
% листинг функции, возвращающей
% вектор Ksi, содержащий значения  ,
% вектор Nu, содержащий значения  ,
% скаляр d - оценку фрактальной размерности
% S - бинарная матрица фрактала
% Lmax - максимальное число квадратов,
% покрывающих фрактал

Ly=size(S,1);
Lx=size(S,2);
S1=S;
% инвертирование матрицы S
for i=1:Ly
        for j=1:Lx
            if S(i,j)==255
               S1(i,j)=0;
            else
                S1(i,j)=1;
            end;
        end;
end;

for L=1:Lmax
  N(L)=0;
  Bx=fix(Lx/L);
  By=fix(Ly/L);
  tmp=zeros(By,Bx);
```

```
  for j=1:By
    Ny1=(j-1)*L+1;
    Ny2=j*L;
    for i=1:Bx
      Nx1=(i-1)*L+1;
      Nx2=i*L;
      tmp=S(Ny1:Ny2,Nx1:Nx2);
      cnt=sum(sum(tmp));
      if cnt>0
         N(L)=N(L)+1;
      end;
     end;
   end;
end;
tmp=1:Lmax;
Ksi(tmp)=log(tmp);
Nu=log(N);
B=[ones(size(Ksi')) Ksi'];
tmp=B\Nu';
d=-tmp(2);
```

Отметим, что в данной функции в качестве одного из входных параметров используется бинарная матрица фрактала, представляющая собой изображение фрактала в bmp-формате. Для создания данной переменной необходимо выполнить следующую последовательность действий:

1. Создать изображение фрактала в MATLAB.
2. Сохранить изображение фрактала на диск в формате bmp.
3. Считать с диска файл, созданный на предыдущем этапе, и занести считанную бинарную матрицу в выбранную переменную, выполнив в командной строке MATLAB команду:

   ```
   Имя_переменной=imread('Имя_файла')
   ```

4. Вызвать функцию `DimMinkovsky.m`

Ниже приведен пример вычисления фрактальной размерности ковра Серпинского.

```
>> S=imread('1.bmp'); % считывание изображения ковра Серпиского
>> [Ksi,Nu,d]=DimMinkovsky(S,30)
Ksi =
  Columns 1 through 10
         0    0.6931    1.0986    1.3863    1.6094    1.7918
                        1.9459    2.0794    2.1972    2.3026
  Columns 11 through 20
    2.3979    2.4849    2.5649    2.6391    2.7081    2.7726
                        2.8332    2.8904    2.9444    2.9957
```

```
  Columns 21 through 30
    3.0445    3.0910    3.1355    3.1781    3.2189    3.2581
                        3.2958    3.3322    3.3673    3.4012
Nu =
  Columns 1 through 10
    9.8373    8.7709    8.1206    7.6926    7.3271
    7.0596    6.8035    6.6080    6.4281    6.2672
  Columns 11 through 20
    6.1717    6.0113    5.8289    5.6802    5.5607    5.5491
                        5.4889    5.3423    5.2575    5.1475
  Columns 21 through 30
    5.0173    5.1059    4.9972    4.8828    4.7791    4.8040
                        4.6540    4.6728    4.5643    4.5109
d =
    1.5618
>> plot(Nu,Ksi) % построение графика
               % зависимости log(1/eps)=f(log(N(1/eps)))
```

Рис. 17.23. График зависимости $\log(1/\varepsilon) = f(\log(N(\varepsilon)))$

Отметим, что улучшить оценку значения фрактальной размерности можно, если использовать значения зависимости $\log(1/\varepsilon) = f(\log(N(\varepsilon)))$, у которых $\log(N(\varepsilon)) \geq$ $\geq 6,3$.

***Задача* 17.8**

Вычислите значения фрактальной размерности, используя значения зависимости $\log(1/\varepsilon) = f(\log(N(\varepsilon)))$, *у которых* $\log(N(\varepsilon)) \geq 6,3$.

Сравните вычисленное и точное значения.

Поточечная размерность. Рассмотрим какую-либо траекторию в фазовом пространстве на протяжении достаточно продолжительного интервала времени (рис. 17.24).

Во-первых, произведем некоторую выборку точек, с тем чтобы получить на траектории достаточное количество точек. Во-вторых, опишем вокруг произвольно выбранной точки куб с ребром r и подсчитаем число выборочных точек $N(r)$, попавших

Рис. 17.24. К объяснению понятия «поточечная фрактальная размерность»

внутрь сферы. Вероятность того, что выборочная точка окажется внутри сферы, можно найти, разделив $N(r)$ на полное число выборочных точек на траектории:

$$P\left(r\right)=\frac{N\left(r\right)}{N_0}. \tag{17.8}$$

Для одномерной орбиты, например замкнутой периодической орбиты, вероятность $P(r)$, вычисляемая в соответствии с (17.8), линейна по r при $r\to\infty$, $N_0\to\infty$:

$$P(r)\approx br.$$

Данный факт позволяет предложить использовать для определения размерности траектории долю времени, проводимого траекторией внутри рассматриваемого куба, центр которого находится в точке с радиусом-вектором x_i:

$$d_p=\lim_{r\to 0}\frac{\log P\left(r,x_i\right)}{\log r}. \tag{17.9}$$

В связи с тем, что фрактальная размерность d_p, вычисляемая в соответствии с (17.9), оказывается в общем случае зависящей от x_i, более точная оценка d_p может быть получена, если использовать усредненную поточечную размерность. Для получения поточечной размерности следует случайным образом выбрать множество точек $M<N_0$, в каждой его точке вычислить $d_p(x_i)$ и затем $\widehat{d}_p$:

$$\widehat{d}_p=\frac{1}{M}\sum_{i=1}^{M}d_p\left(x_i\right). \tag{17.10}$$

Альтернативный (17.10) вариант вычисления d_p основан на усреднении вероятности $P(r,x_j)$, которое проводится по случайному подмножеству из M точек, расположенному вокруг аттрактора $M\le N_0$. Здесь предполагается, что

$$\lim_{r\to 0}\frac{1}{M}\sum_{i=1}^{M}P\left(r,x_i\right)=ar^{d_p}$$

или

$$d_p = \lim_{r \to 0} \frac{\log\left(\frac{1}{M}\sum\limits_{i=1}^{M} P\left(r, x_i\right)\right)}{\log r}.$$

Ниже приведен листинг m-функции `DimPoints.m`, реализующей описанный выше алгоритм вычисления фрактальной размерности двумерных объектов.

```
% листинг функции DimPoints.m
function [Ksi,Nu,d]=DimPoints(S,Lmax);
% листинг функции, возвращающей
% вектор Ksi, содержащий значения log(1/eps),
% вектор Nu, содержащий значения  log(N(eps),
% скаляр d - оценка фрактальной размерности
% S - бинарная матрица фрактала
% Lmax - максимальное число квадратов,
% покрывающих фрактал

Ly=size(S,1);
Lx=size(S,2);
S1=S;
% инвертирование матрицы S
for i=1:Ly
        for j=1:Lx
            if S(i,j)==255
               S1(i,j)=0;
            else
                S1(i,j)=1;
            end;
        end;
end;

% вычисление фрактальной размерности
if mod(Lmax,2)==0
  Lmax=Lmax+1;
end;
LM2=fix(Lmax/2);
P=zeros(Lx*Ly,LM2);
for i=LM2+1:Ly-LM2
  for j=1:LM2+1:Lx-LM2
    if S1(i,j)==1
      for L=1:LM2
        Ny1=i-L;
        Ny2=i+L;
        Nx1=j-L;
        Nx2=j+L;
        tmp=S1(Ny1:Ny2,Nx1:Nx2);
```

```
        m=sum(sum(tmp));
        if m>0
          P(m,L)=P(m,L)+1;
        end;
      end;
    end;
  end;
end;
m=1:Lx*Ly;
for L=1:LM2
  N(L)=sum(P(:,L)./m');
end;
tmp=1:LM2;
Ksi(tmp)=log(2*tmp+1);
Nu=log(N);
B=[ones(size(Ksi')) Ksi'];
tmp=B\Nu';
d=-tmp(2);
% листинг функции DimPoints.m
function [Ksi,Nu,d]=DimPoints(S,Lmax);
% листинг функции, возвращающей
% вектор Ksi, содержащий значения log(1/eps),
% вектор Nu, содержащий значения  log(N(eps),
% скаляр d - оценка фрактальной размерности
% S -бинарная матрица фрактала
% Lmax - максимальное число квадратов,
% покрывающих фрактал

Ly=size(S,1);
Lx=size(S,2);
S1=S;
% инвертирование матрицы S
for i=1:Ly
        for j=1:Lx
            if S(i,j)==255
               S1(i,j)=0;
            else
                S1(i,j)=1;
            end;
        end;
end;

 % вычисление фрактальной размерности
if mod(Lmax,2)==0
  Lmax=Lmax+1;
```

```
end;
LM2=fix(Lmax/2);
P=zeros(Lx*Ly,LM2);
for i=LM2+1:Ly-LM2
  for j=1:LM2+1:Lx-LM2
    if S1(i,j)==1
      for L=1:LM2
        Ny1=i-L;
        Ny2=i+L;
        Nx1=j-L;
        Nx2=j+L;
        tmp=S1(Ny1:Ny2,Nx1:Nx2);
        m=sum(sum(tmp));
        if m>0
          P(m,L)=P(m,L)+1;
        end;
      end;
    end;
  end;
end;
m=1:Lx*Ly;
for L=1:LM2
  N(L)=sum(P(:,L)./m');
end;
tmp=1:LM2;
Ksi(tmp)=log(2*tmp+1);
Nu=log(N);
B=[ones(size(Ksi')) Ksi'];
tmp=B\Nu';
d=-tmp(2);
```

Ниже приведены результаты, возвращаемые функцией `DimPoints.m` и график зависимости $\log(1/\varepsilon) = f(\log(N(\varepsilon)))$.

```
Ksi =
  Columns 1 through 10
    1.0986    1.6094    1.9459    2.1972    2.3979    2.5649
                        2.7081    2.8332    2.9444    3.0445
  Columns 11 through 15
    3.1355    3.2189    3.2958    3.3673    3.4340
Nu =
  Columns 1 through 10
    5.1406    4.2857    3.7617    3.3681    3.0375    2.7734
                        2.5551    2.3605    2.1821    2.0161
  Columns 11 through 15
    1.8694    1.7406    1.6236    1.5152    1.4117
```

```
d =
    1.5908
```

Рис. 17.25. График зависимости $\log(1/\varepsilon) = f(\log(N(\varepsilon)))$

Корреляционная размерность. При вычислении корреляционной размерности, аналогично рассмотренному выше случаю вычисления поточечной размерности, непрерывная траектория заменяется множеством из N точек x_i в фазовом пространстве. Далее вычисляются расстояния между парами точек $s_{ij} = |x_i - x_j|$, используя при этом любую из известных метрик, например евклидову метрику. Корреляционная функция определяется как

$$C(r) = \lim_{N\to\infty} \frac{1}{N^2} \left(\begin{array}{c} \text{число пар } (i,j), \\ \text{для которых расстояние } s_{ij} < r \end{array} \right).$$

Оказывается, что для многих аттракторов данная функция зависит от r, при $r \to 0$ по степенному закону:

$$\lim_{r\to 0} C(r) = ar^d,$$

поэтому корреляционную размерность можно определять по наклону прямой в координатной плоскости $(\log C, \log r)$:

$$d_C = \lim_{r\to 0} \frac{\log C(r)}{\log r}. \tag{17.11}$$

Однако вычисления $C(r)$ оказываются более эффективными, если описать вокруг каждой точки в фазовом пространстве куб вокруг каждой точки x_i и подсчитать число точек, попавшую в каждую сферу, то есть

$$C(r) = \lim_{N\to\infty} \frac{1}{N^2} \sum_{i=1}^{N} \sum_{j=1}^{N} H(r - |x_i - x_j|), \tag{17.12}$$

где

$$H(s) = \begin{cases} 1, \text{if } s \geqslant 0, \\ 0, \text{ if } s < 0. \end{cases}$$

Процедура вычисления данной величины отличается от процедуры вычисления тем, что в данном случае суммирование проводится вокруг каждой точки.

Далее приведен листинг m-функции `DimCorrelation.m`, которая реализует описанный выше алгоритм вычисления корреляционной размерности.

```
% листинг файла DimCorrelation.m
function [X,Y,d] = DimCorrelation(data,Lmin,Lmax,Lstep,Nshift,m)
% функция, возвращающая
% вектор X, содержащий значения  log(r),
% вектор Y, содержащий значения  log(C(r))
% скаляр d, содержащий оценку
% корреляционной размерности
% data - вектор, содержащий мгновенные
% значения временного ряда
% Lmin, Lmax, Lstep - параметры,
% задающие размер текущей сферы,
% описанной вокруг точки временного ряда
% Nshift - шаг прореживания временного ряда
% m - число точек временного ряда,
% используемых для вычисления
% корреляционной размерности

M=length(data);
tmp=zeros(M-Nshift*m,m);
for i=1:M-Nshift*m
  for j=1:m
    tmp(i,j)=data(i+Nshift*(j-1));
  end;
end;
k=0;
for i=1:M-2*m*Nshift
  for j=i+1:M-m*Nshift
    k=k+1;
  end;
end;
C=zeros(k,1);
for i=1:M-2*m*Nshift
  v1=tmp(i,:);
  for j=i+1:M-m*Nshift
    v2=tmp(j,:);
    C(k)=norm(v1-v2);
  end;
end;
N=length(C);
x1=min(data);
x2=max(data);
```

```
for L=Lmin:Lstep:Lmax
  eps=(x2-x1)/10^L;
  S=0;
  for i=1:N
    if C(i)<=eps
      S=S+1;
    end;
  end;
  S=2*S/(N*(N-1));
  X(k)=L;
  Y(k)=log(S);
  k=k+1;
end;
B=[ones(size(X')) X'];
tmp=B\Y';
d=-tmp(2);
```

Для вычисления информационной размерности находят число точек N_i в каждой из N ячеек покрытия и оценивают вероятность P_i обнаружить точку в i-й ячейке:

$$P_i \equiv \frac{N_i}{N_0}, \quad \sum_{i=1}^{N} P_i = 1,$$

где N_0 — общее число точек множества. Подчеркнем, что $N_0 \neq N$.

Информационная энтропия определяется выражением

$$I(\varepsilon) = -\sum_{i=1}^{N} P_i \log P_i. \tag{17.13}$$

Известно [8], что при малых ε информационная энтропия ведет себя как

$$I \approx d_I \log\left(\frac{1}{\varepsilon}\right),$$

поэтому при малых ε информационную размерность можно определить как

$$d_I = \lim_{\varepsilon\to 0} \frac{I(\varepsilon)}{\log(1/\varepsilon)} = \lim_{\varepsilon\to 0} \frac{\sum P_i \log P_i}{\log \varepsilon}. \tag{17.14}$$

Информационная размерность (17.14) связанна с емкостной размерностью. Действительно, если вероятности P_i одинаковы для всех ячеек, то есть

$$P_i = \frac{N_i}{N_0} = \frac{1}{N},$$

то

$$I = \sum P_i \log P_i = -N\frac{1}{N}\log\frac{1}{N} = \log N$$

и $d_I = d_C$. В [9] показано, что в общем случае

$$d_i < d_C.$$

***Задача* 17.9**

Вычислите значения фрактальной размерности фракталов, рассмотренных в данной главе, проводя предварительный анализ зависимости $\log(1/\varepsilon) = f(\log(N(\varepsilon)))$.

Сравните получаемые значения фрактальной размерности, вычисляемой с помощью различных алгоритмов.

ЛИТЕРАТУРА К ГЛАВЕ 17

1. *Mandelbrot B. B.* Les object fractals: forme, hasard et dimantion. Paris: Flamarion, 1975.

2. *Каток А. Б., Хасселблат Б.* Введение в современную теорию динамических систем. М.: Факториал, 1999.

3. *Лихтенберг А., Либерман М.* Регулярная и стохастическая динамика. М.: Меркурий-ПРЕСС, 2000.

4. *Шустер Г.* Детерминированный хаос. М.: Мир, 1988.

5. *Кренкель Э. Т.* Сжатие сигналов с применением теории фракталов. М.: МТУСИ, 1996.

6. *Кроновер Р. М.* Фракталы и хаос в динамических системах. Основы теории. М.: Постмаркер, 2000.

7. http://www.exponenta.ru

8. *Мун Ф.* Хаотические колебания. М.: Мир, 1990. 312 с.

9. *Farmer J., D., Ott, J., and Yorke, J., A.* The Dimension of Chaotic Attractors. Physica 7D. 1983. P. 153–170.

Приложение А

ОСНОВНЫЕ ПРИЕМЫ РАБОТЫ С ПАКЕТОМ MATLAB

А.1. ВВЕДЕНИЕ В ПАКЕТ MATLAB

MATLAB является интерактивным, матрично-ориентированным пакетом, предназначенным для выполнения научных и инженерных расчетов. Имя пакета MATLAB является аббревиатурой двух слов MATrix LABoratory (МАТричная ЛАБоратория). Цель данного раздела — познакомить читателя с необходимыми сведениями для начала работы с пакетом MATLAB. Мы рекомендуем совместить изучение данного раздела с одновременным выполнением приведенных в разделе примеров. Отметим, что во время работы с системой можно пользоваться встроенной оперативной помощью, которая содержит сведения, необходимые для успешной работы с пакетом. После входа в систему команда help выведет список групп, в которые объединены функции:

```
>> help

  To get started, select "MATLAB Help" from the Help menu.

HELP topics:

matlab\general       -  General purpose commands.
matlab\ops           -  Operators and special characters.
matlab\lang          -  Programming language constructs.
matlab\elmat         -  Elementary matrices and matrix
                        manipulation.
matlab\elfun         -  Elementary math functions.
matlab\specfun       -  Specialized math functions.
matlab\matfun        -  Matrix functions - numerical linear
                        algebra.
matlab\datafun       -  Data analysis and Fourier transforms.
matlab\audio         -  Audio support.
matlab\polyfun       -  Interpolation and polynomials.
matlab\funfun        -  Function functions and ODE solvers.
matlab\sparfun       -  Sparse matrices.
matlab\graph2d       -  Two dimensional graphs.
matlab\graph3d       -  Three dimensional graphs.
matlab\specgraph     -  Specialized graphs.
matlab\graphics      -  Handle Graphics.
matlab\uitools       -  Graphical user interface tools.
```

```
matlab\strfun        -  Character strings.
matlab\iofun         -  File input/output.
matlab\timefun       -  Time and dates.
matlab\datatypes     -  Data types and structures.
matlab\verctrl       -  Version control.
matlab\winfun     -  Windows Operating System Interface Files
                     (DDE/ActiveX)
matlab\demos         -  Examples and demonstrations.
toolbox\local        -  Preferences.
simulink\simulink    -  Simulink
simulink\blocks      -  Simulink block library.
simulink\simdemos    -  Simulink 4 demonstrations and samples.
simdemos\aerospace   -  Simulink: Aerospace model demonstrations
                                  and samples.
simdemos\automotive  -  Simulink: Automotive model demonstrations
                                  and samples.
simdemos\simfeatures -  Simulink: Feature demonstrations
                                  and samples.
simdemos\simgeneral  -  Simulink: General model demonstrations
                                  and samples.
simdemos\simnew      -  Simulink: New features model
                                  demonstrations and samples.
simulink\dee         -  Differential Equation Editor
stateflow\stateflow  -  Stateflow
stateflow\sfdemos    -  Stateflow demonstrations and samples.
stateflow\coder      -  Stateflow Coder
rtw\rtw              -  Real-Time Workshop
rtw\rtwdemos         -  Real-Time Workshop Demonstrations:
asap2\asap2          -  (No table of contents file)
asap2\user           -  (No table of contents file)
cdma\cdma            -  CDMA Reference Blockset.
cdma\cdmamasks       -  CDMA Reference Blockset
                          mask helper functions.
cdma\cdmamex         -  CDMA Reference Blockset
                          S-Functions.
cdma\cdmademos       -  CDMA Reference Blockset
                          demonstrations and examples.
commblks\commblks    -  Communications Blockset.
commblks\commmasks   -  Communications Blockset
                          mask helper functions.
commblks\commmex     -  Communications Blockset S-functions.
commblks\commblksdemos -  Communications Blockset Demos.
commblks\commblksobsolete -  Archived Simulink Files from
                             Communications Toolbox Version 1.5.
comm\comm            -  Communications Toolbox.
```

```
comm\commdemos       -  Communications Toolbox Demonstrations.
comm\commobsolete    -  Archived MATLAB Files from
                          Communications Toolbox Version 1.5.
control\control      -  Control System Toolbox.
control\ctrlguis     -  Control System Toolbox --
                         GUI support functions.
control\ctrlobsolete -  Control System Toolbox --
                         obsolete commands.
control\ctrlutil     -  (No table of contents file)
control\ctrldemos    -  Control System Toolbox -- Demos.
dspblks\dspblks      -  DSP Blockset.
dspblks\dspmasks     -  DSP Blockset mask helper functions.
dspblks\dspmex       -  DSP Blockset S-Function MEX-files.
dspblks\dspdemos     -  DSP Blockset demonstrations
                         and examples.
dsprtw\util_c        -  (No table of contents file)
daq\daq              -  Data Acquisition Toolbox.
daq\daqdemos         -  Data Acquisition Toolbox -
                          Data Acquisition Demos.
database\database    -  Database Toolbox.
database\dbdemos     -   Database Toolbox
                           Demonstration Functions.
database\vqb         -  Visual Query Builder functions.
datafeed\datafeed    -  Datafeed Toolbox.
datafeed\dfgui       -   Datafeed Toolbox Graphical
                           User Interface
toolbox\dials        -  Dials & Gauges Blockset
toolbox\exlink       -  (No table of contents file)
filterdesign\filterdesign -  Filter Design Toolbox.
filterdesign\quantization -  (No table of contents file)
filterdesign\filtdesdemos -  Filter Design
                              Toolbox Demonstrations.
finderiv\finderiv    -  Financial Derivatives Toolbox.
ftseries\ftseries    -  Financial Time Series Toolbox
ftseries\ftsdemos    -  (No table of contents file)
ftseries\ftsdata     -  (No table of contents file)
ftseries\ftstutorials -  (No table of contents file)
finance\finance      -  Financial Toolbox.
finance\calendar     -  Financial Toolbox calendar functions.
finance\findemos     -  Financial Toolbox demonstration functions.
finance\finsupport   -  (No table of contents file)
toolbox\fixpoint     -  Fixed-Point Blockset
fixpoint\fxpdemos    -  Fixed-Point Blockset Demos
fixpoint\obsolete    -  Obsolete Fixed-Point Blockset
fuzzy\fuzzy          -  Fuzzy Logic Toolbox.
```

```
fuzzy\fuzdemos       -  Fuzzy Logic Toolbox Demos.
garch\garch          -  GARCH Toolbox.
garch\garchdemos     -  (No table of contents file)
images\images        -  Image Processing Toolbox.
images\imdemos       -  Image Processing Toolbox ---
                          demos and sample images
instrument\instrument -  Instrument Control Toolbox.
instrument\instrumentdemos -  (No table of contents file)
lmi\lmictrl          -  LMI Control Toolbox: Control Applications
lmi\lmilab           -  LMI Control Toolbox
toolbox\compiler     -  MATLAB Compiler
toolbox\rptgen       -  MATLAB Report Generator
map\map              -  Mapping Toolbox
map\mapdisp          -  Mapping Toolbox Map Definition and Display.
map\mapproj          -  Mapping Toolbox Projections.
mpc\mpccmds          -  Model Predictive Control Toolbox.
mpc\mpcdemos         -  Model Predictive Control Toolbox
motdsp\motdsp        -  Motorola DSP Developer's Kit
motdsp\motdspmex     -  Motorola DSP Developers Kit
motdspasm\bin        -  (No table of contents file)
motdsp\motdspblks    -  Motorola DSP Developers Kit
                          block libraries
motdsp\motdspmasks   -  Motorola DSP blockset mask
                          helper functions.
motdsp\motdspdemos   -  MOTDSP Toolbox demonstrations
                          and examples:
mutools\commands     -  Mu-Analysis and Synthesis Toolbox.
mutools\subs         -  Mu-Analysis and Synthesis Toolbox.
nnet\nnet            -  Neural Network Toolbox.
nnet\nnutils         -  (No table of contents file)
nnet\nncontrol       -  Neural Network Toolbox
                          Control System Functions.
nnet\nndemos         -  Neural Network Demonstrations.
nnet\nnobsolete      -  (No table of contents file)
toolbox\ncd          -  Nonlinear Control Design Blockset
toolbox\optim        -  Optimization Toolbox.
toolbox\pde          -  Partial Differential Equation Toolbox.
powersys\powersys    -  Power System Blockset.
powersys\powerdemo   -  Power System Blockset Demos.
rtw\ada              -  Real-Time Workshop Ada Coder
targets\ecoder       -  Real-Time Workshop Embedded Coder
toolbox\reqmgt       -  Requirements Management Interface.
toolbox\robust       -  Robust Control Toolbox.
toolbox\sb2sl        -  SB2SL (converts SystemBuild to Simulink)
signal\signal        -  Signal Processing Toolbox.
```

```
signal\fdatoolgui    -  Signal Processing Toolbox GUI.
signal\sptoolgui     -  Signal Processing Toolbox GUI
signal\sigdemos      -  Signal Processing Toolbox
                          Demonstrations.
rtw\accel            -  Simulink Accelerator
simulink\mdldiff     -  Model Differencing for
                          Simulink and Stateflow
simulink\simcoverage -  Simulink Model Coverage Tool
toolbox\rptgenext    -  Simulink Report Generator
toolbox\splines      -  Spline Toolbox.
toolbox\stats        -  Statistics Toolbox.
toolbox\symbolic     -  Symbolic Math Toolbox.
ident\ident          -  System Identification Toolbox.
ident\idobsolete     -  (No table of contents file)
ident\idguis         -  (No table of contents file)
ident\idutils        -  (No table of contents file)
ident\iddemos        -  (No table of contents file)
ident\idhelp         -  (No table of contents file)
wavelet\wavelet      -  Wavelet Toolbox.
wavelet\wavedemo     -  Wavelet Toolbox Demonstrations.
xpc\xpc              -  xPC Target
build\xpcblocks      -  (No table of contents file)
xpc\xpcdemos         -  xPC Target -- demos and
                          sample script files.
kernel\embedded      -  xPC Target Embedded Option
MatlabR12\work       -  (No table of contents file)

For more help on directory/topic, type "help topic".

>> help

HELP topics:

matlab\general       -  General purpose commands.
matlab\ops           -  Operators and special characters.
matlab\lang          -  Programming language constructs.
matlab\elmat         -  Elementary matrices and
                          matrix manipulation.
matlab\elfun         -  Elementary math functions.
matlab\specfun       -  Specialized math functions.
matlab\matfun        -  Matrix functions -
                          numerical linear algebra.
matlab\datafun       -  Data analysis and Fourier transforms.
matlab\audio         -  Audio support.
matlab\polyfun       -  Interpolation and polynomials.
```

```
matlab\funfun        -  Function functions and ODE solvers.
matlab\sparfun       -  Sparse matrices.
matlab\graph2d       -  Two dimensional graphs.
matlab\graph3d       -  Three dimensional graphs.
matlab\specgraph     -  Specialized graphs.
matlab\graphics      -  Handle Graphics.
matlab\uitools       -  Graphical user interface tools.
matlab\strfun        -  Character strings.
matlab\iofun         -  File input/output.
matlab\timefun       -  Time and dates.
matlab\datatypes     -  Data types and structures.
matlab\verctrl       -  Version control.
matlab\winfun    -  Windows Operating System
                        Interface Files (DDE/ActiveX)
matlab\demos         -  Examples and demonstrations.
toolbox\local        -  Preferences.
simulink\simulink    -  Simulink
simulink\blocks      -  Simulink block library.
simulink\simdemos    -  Simulink 4 demonstrations
                                 and samples.
simdemos\aerospace - Simulink: Aerospace model demonstrations
                               and samples.
simdemos\automotive -  Simulink: Automotive model demonstrations
                                 and samples.
simdemos\simfeatures -  Simulink: Feature demonstrations
                                   and samples.
simdemos\simgeneral  -  Simulink: General model demonstrations
                                   and samples.
simdemos\simnew  -  Simulink: New features model demonstrations
                              and samples.
simulink\dee         -  Differential Equation Editor
stateflow\stateflow  -  Stateflow
stateflow\sfdemos    -  Stateflow demonstrations and samples.
stateflow\coder      -  Stateflow Coder
rtw\rtw              -  Real-Time Workshop
rtw\rtwdemos         -  Real-Time Workshop Demonstrations:
asap2\asap2          -  (No table of contents file)
asap2\user           -  (No table of contents file)
cdma\cdma            -  CDMA Reference Blockset.
cdma\cdmamasks       -  CDMA Reference Blockset mask helper functions.
cdma\cdmamex         -  CDMA Reference Blockset S-Functions.
cdma\cdmademos   -  CDMA Reference Blockset demonstrations
                        and examples.
commblks\commblks    -  Communications Blockset.
commblks\commmasks   -  Communications Blockset mask helper functions.
```

```
commblks\commmex     -  Communications Blockset S-functions.
commblks\commblksdemos -  Communications Blockset Demos.
commblks\commblksobsolete -  Archived Simulink Files from
                                Communications Toolbox Version 1.5.
comm\comm  -  Communications Toolbox.
comm\commdemos  -  Communications Toolbox Demonstrations.
comm\commobsolete -  Archived MATLAB Files from
                        Communications Toolbox Version 1.5.
control\control      -  Control System Toolbox.
control\ctrlguis     -  Control System Toolbox --
                          GUI support functions.
control\ctrlobsolete -  Control System Toolbox --
                          obsolete commands.
control\ctrlutil     -  (No table of contents file)
control\ctrldemos    -  Control System Toolbox -- Demos.
dspblks\dspblks      -  DSP Blockset.
dspblks\dspmasks     -  DSP Blockset mask helper functions.
dspblks\dspmex       -  DSP Blockset S-Function MEX-files.
dspblks\dspdemos     -  DSP Blockset demonstrations and examples.
dsprtw\util_c        -  (No table of contents file)
daq\daq              -  Data Acquisition Toolbox.
daq\daqdemos         -  Data Acquisition Toolbox -
                          Data Acquisition Demos.
database\database    -  Database Toolbox.
database\dbdemos     -   Database Toolbox Demonstration
                           Functions.
database\vqb         -  Visual Query Builder functions.
datafeed\datafeed    -  Datafeed Toolbox.
datafeed\dfgui       -  Datafeed Toolbox Graphical
                          User Interface
toolbox\dials        -  Dials & Gauges Blockset
toolbox\exlink       -  (No table of contents file)
filterdesign\filterdesign -  Filter Design Toolbox.
filterdesign\quantization -  (No table of contents file)
filterdesign\filtdesdemos -  Filter Design Toolbox
                                Demonstrations.
finderiv\finderiv    -  Financial Derivatives Toolbox.
ftseries\ftseries    -  Financial Time Series Toolbox
ftseries\ftsdemos    -  (No table of contents file)
ftseries\ftsdata     -  (No table of contents file)
ftseries\ftstutorials -  (No table of contents file)
finance\finance      -  Financial Toolbox.
finance\calendar     -  Financial Toolbox calendar
                          functions.
finance\findemos     -  Financial Toolbox demonstration
```

```
                          functions.
finance\finsupport   -  (No table of contents file)
toolbox\fixpoint     -  Fixed-Point Blockset
fixpoint\fxpdemos    -  Fixed-Point Blockset Demos
fixpoint\obsolete    -  Obsolete Fixed-Point Blockset
fuzzy\fuzzy          -  Fuzzy Logic Toolbox.
fuzzy\fuzdemos       -  Fuzzy Logic Toolbox Demos.
garch\garch          -  GARCH Toolbox.
garch\garchdemos     -  (No table of contents file)
images\images        -  Image Processing Toolbox.
images\imdemos       -  Image Processing Toolbox ---
                          demos and sample images
instrument\instrument -  Instrument Control Toolbox.
instrument\instrumentdemos -  (No table of contents file)
lmi\lmictrl          -  LMI Control Toolbox: Control
                          Applications
lmi\lmilab           -  LMI Control Toolbox
toolbox\compiler     -  MATLAB Compiler
toolbox\rptgen       -  MATLAB Report Generator
map\map              -  Mapping Toolbox
map\mapdisp          -  Mapping Toolbox Map Definition
                          and Display.
map\mapproj          -  Mapping Toolbox Projections.
mpc\mpccmds          -  Model Predictive Control Toolbox.
mpc\mpcdemos         -  Model Predictive Control Toolbox
motdsp\motdsp        -  Motorola DSP Developer's Kit
motdsp\motdspmex     -  Motorola DSP Developers Kit
motdspasm\bin        -  (No table of contents file)
motdsp\motdspblks    -  Motorola DSP Developers Kit
                          block libraries
motdsp\motdspmasks   -  Motorola DSP blockset mask helper
                          functions.
motdsp\motdspdemos   -  MOTDSP Toolbox demonstrations
                          and examples:
mutools\commands     -  Mu-Analysis and Synthesis Toolbox.
mutools\subs         -  Mu-Analysis and Synthesis Toolbox.
nnet\nnet            -  Neural Network Toolbox.
nnet\nnutils         -  (No table of contents file)
nnet\nncontrol       -  Neural Network Toolbox Control
                          System Functions.
nnet\nndemos         -  Neural Network Demonstrations.
nnet\nnobsolete      -  (No table of contents file)
toolbox\ncd          -  Nonlinear Control Design Blockset
toolbox\optim        -  Optimization Toolbox.
toolbox\pde          -  Partial Differential Equation Toolbox.
```

```
powersys\powersys    -  Power System Blockset.
powersys\powerdemo   -  Power System Blockset Demos.
rtw\ada              -  Real-Time Workshop Ada Coder
targets\ecoder       -  Real-Time Workshop Embedded Coder
toolbox\reqmgt       -  Requirements Management Interface.
toolbox\robust       -  Robust Control Toolbox.
toolbox\sb2sl        -  SB2SL (converts SystemBuild to Simulink)
signal\signal        -  Signal Processing Toolbox.
signal\fdatoolgui    -  Signal Processing Toolbox GUI.
signal\sptoolgui     -  Signal Processing Toolbox GUI
signal\sigdemos      -  Signal Processing Toolbox
                          Demonstrations.
rtw\accel            -  Simulink Accelerator
simulink\mdldiff     -  Model Differencing for Simulink and
                          Stateflow
simulink\simcoverage -  Simulink Model Coverage Tool
toolbox\rptgenext    -  Simulink Report Generator
toolbox\splines      -  Spline Toolbox.
toolbox\stats        -  Statistics Toolbox.
toolbox\symbolic     -  Symbolic Math Toolbox.
ident\ident          -  System Identification Toolbox.
ident\idobsolete     -  (No table of contents file)
ident\idguis         -  (No table of contents file)
ident\idutils        -  (No table of contents file)
ident\iddemos        -  (No table of contents file)
ident\idhelp         -  (No table of contents file)
wavelet\wavelet      -  Wavelet Toolbox.
wavelet\wavedemo     -  Wavelet Toolbox Demonstrations.
xpc\xpc              -  xPC Target
build\xpcblocks      -  (No table of contents file)
xpc\xpcdemos         -  xPC Target -- demos and
                           sample script files.
kernel\embedded      -  xPC Target Embedded Option
MatlabR12\work       -  (No table of contents file)
For more help on directory/topic, type "help topic".
>>
```

Команда `help <имя_группы>` выведет на экран список функций, размещенных в этой группе с краткими пояснениями. Например, команда `help wavelet` выдаст список функций, реализующих методы вэйвлет-анализа:

```
>> help wavelet
  Wavelet Toolbox.
  Version 2.0   (R12) 16-June-2000
  What's new.
    Readme     - New features, bug fixes, and changes in this version.
```

```
                To display the Readme file for Version 2.0, type
                "whatsnew wavelet" in the MATLAB command window.
Wavelet Toolbox GUI (Graphical User Interface).
  wavemenu    - Start graphical user interface tools.
Wavelets: General.
  biorfilt    - Biorthogonal wavelet filter set.
  centfrq     - Wavelet center frequency.
  dyaddown    - Dyadic downsampling.
  dyadup      - Dyadic upsampling.
  intwave     - Integrate wavelet function psi.
  orthfilt    - Orthogonal wavelet filter set.
  qmf         - Quadrature mirror filter.
  scal2frq    - Scale to frequency.
  wavefun     - Wavelet and scaling functions.
  wavemngr    - Wavelet manager.
  wfilters    - Wavelet filters.
  wmaxlev     - Maximum wavelet decomposition level.

Wavelet Families.
  biorwavf    - Biorthogonal spline wavelet filters.
  cgauwavf    - Complex Gaussian wavelet.
  cmorwavf    - Complex Morlet wavelet.
  coifwavf    - Coiflet wavelet filter.
  dbaux       - Daubechies wavelet filter computation.
  dbwavf      - Daubechies wavelet filters.
  fbspwavf    - Complex Frequency B-Spline wavelet.
  gauswavf    - Gaussian wavelet.
  mexihat     - Mexican hat wavelet.
  meyer       - Meyer wavelet.
  meyeraux    - Meyer wavelet auxiliary function.
  morlet      - Morlet wavelet.
  rbiowavf    - Reverse Biorthogonal spline wavelet filters.
  shanwavf    - Complex Shannon wavelet.
  symaux      - Symlet wavelet filter computation.
  symwavf     - Symlet wavelet filter.

Continuous Wavelet: One-Dimensional.
  cwt         - Real or Complex Continuous
                  wavelet coefficients 1-D.

Discrete Wavelets: One-Dimensional.
  appcoef     - Extract 1-D approximation coefficients.
  detcoef     - Extract 1-D detail coefficients.
  dwt         - Single-level discrete
                  1-D wavelet transform.
```

```
  dwtmode     - Discrete wavelet transform extension mode.
  idwt        - Single-level inverse discrete
                  1-D wavelet transform.
  upcoef      - Direct reconstruction from
                  1-D wavelet coefficients.
  upwlev      - Single-level reconstruction of
                  1-D wavelet decomposition.
  wavedec     - Multi-level 1-D wavelet decomposition.
  waverec     - Multi-level 1-D wavelet reconstruction.
  wrcoef      - Reconstruct single branch from
                  1-D wavelet coefficients.

Discrete Wavelets: Two-Dimensional.
  appcoef2    - Extract 2-D approximation coefficients.
  detcoef2    - Extract 2-D detail coefficients.
  dwt2        - Single-level discrete 2-D wavelet transform.
  dwtmode     - Discrete wavelet transform extension mode.
  idwt2       - Single-level inverse discrete
                  2-D wavelet transform.
  upcoef2     - Direct reconstruction from
                  2-D wavelet coefficients.
  upwlev2     - Single-level reconstruction of
                  2-D wavelet decomposition.
  wavedec2    - Multi-level 2-D wavelet decomposition.
  waverec2    - Multi-level 2-D wavelet reconstruction.
  wrcoef2     - Reconstruct single branch from
                  2-D wavelet coefficients.

Wavelets Packets Algorithms.
  bestlevt    - Best level tree (wavelet packet).
  besttree    - Best tree (wavelet packet).
  entrupd     - Entropy update (wavelet packet).
  wentropy    - Entropy(wavelet packet).
  wp2wtree    - Extract wavelet tree from
                  wavelet packet tree.
  wpcoef      - Wavelet packet coefficients.
  wpcutree    - Cut wavelet packet tree.
  wpdec       - Wavelet packet decomposition 1-D.
  wpdec2      - Wavelet packet decomposition 2-D.
  wpfun       - Wavelet packet functions.
  wpjoin      - Recompose wavelet packet.
  wprcoef     - Reconstruct wavelet packet coefficients.
  wprec       - Wavelet packet reconstruction 1-D.
  wprec2      - Wavelet packet reconstruction 2-D.
  wpsplt      - Split (decompose) wavelet packet.
```

```
Discrete Stationary Wavelet Transform Algorithms.
  iswt        - Inverse discrete stationary
                  wavelet transform 1-D.
  iswt2       - Inverse discrete stationary
                  wavelet transform 2-D.
  swt         - Discrete stationary wavelet transform 1-D.
  swt2        - Discrete stationary wavelet transform 2-D.

De-noising and Compression for Signals and Images.
  ddencmp     - Default values for de-noising or
                 compression.
  thselect    - Threshold selection for de-noising.
  wbmpen      - Penalized threshold for wavelet
                  1-D or 2-D denoising.
  wdcbm       - Thresholds for wavelet
                  1-D using Birge-Massart strategy.
  wdcbm2      - Thresholds for wavelet
                   2-D using Birge-Massart strategy.
  wden        - Automatic 1-D de-noising using wavelets.
  wdencmp     - De-noising or compression using wavelets.
  wnoise      - Generate noisy wavelet test data.
  wnoisest    - Estimate noise of 1-D wavelet coefficients.
  wpbmpen     - Penalized threshold for
                  wavelet packet denoising.
  wpdencmp    - De-noising or compression using
                  wavelet packets.
  wpthcoef    - Wavelet packet coefficients thresholding.
  wthcoef     - Wavelet coefficient thresholding 1-D.
  wthcoef2    - Wavelet coefficient thresholding 2-D.
  wthresh     - Perform soft or hard thresholding.
  wthrmngr    - Threshold settings manager.

Tree Management Utilities.
  allnodes    - Tree nodes.
  depo2ind    - Node depth-position to node index.
  drawtree    - Draw wavelet packet decomposition
                 tree (GUI).
  dtree       - Constructor for the class DTREE.
  get         - Get tree object field contents.
  ind2depo    - Node index to node depth-position.
  isnode      - True for existing node.
  istnode     - Determine indices of terminal nodes.
  leaves      - Determine terminal nodes.
  nodeasc     - Node ascendants.
```

```
    nodedesc    - Node descendants.
    nodejoin    - Recompose node.
    nodepar     - Node parent.
    nodesplt    - Split (decompose) node.
    noleaves    - Determine nonterminal nodes.
    ntnode      - Number of terminal nodes.
    ntree       - Constructor for the class NTREE.
    plot        - Plot tree object.
    read        - Read values in tree object fields.
    readtree    - Read wavelet packet decomposition tree
                  from a figure.
    set         - Set tree object field contents.
    tnodes      - Determine terminal nodes.
    treedpth    - Tree depth.
    treeord     - Tree order.
    wptree      - Constructor for the class WPTREE.
    wpviewcf - Plot wavelet packets colored coefficients.
    write       - Write values in tree object fields.
    wtbo        - Constructor for the class WTBO.
    wtreemgr    - NTREE object manager.

  General Utilities.
    wcodemat    - Extended pseudocolor matrix scaling.
    wextend     - Extend a Vector or a Matrix.
    wkeep       - Keep part of a vector or a matrix.
    wrev        - Flip vector.

  Other.
    instdfft    - Inverse non-standard 1-D
                    fast Fourier transform.
    nstdfft     - Non-standard 1-D fast Fourier transform.
    wvarchg     - Find variance change points.
   Wavelets Information.
    waveinfo    - Information on wavelets.
   Demonstrations.
    wavedemo    - Wavelet Toolbox demos.
   See also WAVEDEMO.
>>
```

Команда `help <имя_функции>` выдаст подробную информацию о функции. Например, команда `help waveinfo` выдаст информацию о функции, возвращающей справочную информацию о вэйвлете конкретного вида:

```
>> help waveinfo

 WAVEINFO Information on wavelets.
```

```
    WAVEINFO provides information for all the wavelets
    within the toolbox.

    WAVEINFO('wname') provides information for the wavelet
    family whose short name is specified by the string
    'wname'.

    Available family short names are:
    'haar'   : Haar wavelet.
    'db'     : Daubechies wavelets.
    'sym'    : Symlets.
    'coif'   : Coiflets.
    'bior'   : Biorthogonal wavelets.
    'rbio'   : Reverse biorthogonal wavelets.
    'meyr'   : Meyer wavelet.
    'dmey'   : Discrete Meyer wavelet.
    'gaus'   : Gaussian wavelets.
    'mexh'   : Mexican hat wavelet.
    'morl'   : Morlet wavelet.
    'cgau'   : Complex Gaussian wavelets.
    'cmor'   : Complex Morlet wavelets.
    'shan'   : Complex Shannon wavelets.
    'fbsp'   : Complex Frequency B-spline wavelets.

    or user-defined short names for their own wavelet
    families (see WAVEMNGR).

    WAVEINFO('wsys') provides information on wavelet packets.

    See also WAVEMNGR.
>>
```

Вы можете также познакомиться с некоторыми возможностями системы MATLAB с помощью команд `intro` и `demo`. Для более основательного знакомства с системой желательно ознакомиться с книгой «MATLAB User's Guide» (Руководство пользователя).

А.2. РАБОТА В КОМАНДНОМ ОКНЕ

А.2.1. ВХОД В СИСТЕМУ MATLAB

Войти в пакет MATLAB в большинстве операционных систем (например, в системе MS DOS) после входа в саму операционную среду можно, набрав в ответ на системный запрос команду matlab. При работе в ОС Windows-9x, Windows-2000,

Windows XP необходимо найти соответствующую иконку и кликнуть по ней. Выход пакета осуществляется с помощью команды `quit`. В версии пакета 5.0 и выше, работающих в ОС семейства Windows, используется собственный встроенный редактор текста. В обоих случаях командное окно пакета MATLAB находится в одном окне, а редактор текста — в другом окне.

В командном окне MATLAB помимо собственно команд MATLAB можно использовать системные команды ОС MS DOS. Например, команда `dir` выводит на экран содержимое текущего каталога, команда `what` выводит только список m-файлов[1] текущего каталога. Команда `cd` позволяет сменить текущий каталог, а команды `delete имя_файла` и `type имя_файла` стирают и печатают на экране содержимое файла соответственно.

А.2.2. ИНТЕРАКТИВНЫЙ ДОСТУП К СПРАВОЧНОЙ ИНФОРМАЦИИ

Получить информацию о функциях пакета MATLAB в процессе работы можно следующими способами :

- команда `help`;
- команда `lookfor`;
- меню Help;
- просмотр и вывод на печать страниц документации;
- обращение к web-серверу фирмы The MathWorks.

Команда HELP

Основной и наиболее быстрый способ выяснить синтаксис и особенности применения m-функции — использовать команду

```
help <имя m-функции>
```

Соответствующая информация появляется непосредственно в командном окне. Например, команда `help zeros` выведет в командное окно следующую информацию на английском языке:

```
>> help zeros
 ZEROS  Zeros array.
    ZEROS(N) is an N-by-N matrix of zeros.
    ZEROS(M,N) or ZEROS([M,N]) is an M-by-N matrix of zeros.
    ZEROS(M,N,P,...) or ZEROS([M N P ...]) is an ...
    M-by-N-by-P-by-array of zeros.
    ZEROS(SIZE(A)) is the same size as A and all zeros.
    See also ONES.
```

[1] Здесь и далее m-файлом мы будем называть любой текстовый файл, содержащий набор команд пакета MATLAB, имеющий расширение `.m`. Подробнее об этих файлах и их роли в системе MATLAB см. в Приложении В.

Следует обратить внимание, что текст интерактивной справки использует верхний регистр для написания имен функций и переменных, чтобы выделить их из основной части текста. Однако при использовании функций их имена необходимо вводить с помощью символов нижнего регистра.

Команда `help` без аргументов выводит на экран список каталогов, которые имеются в системе с кратким описанием их содержимого. Повторный набор этой команды с именем каталога, например `help elmat`, выведет список функций, предназначенных для создания и работы с матрицами специального вида. Ввод команды с именем определенной функции выдаст на экран описание этой функции. Следует особо обратить внимание, что в качестве ответа на запрос о помощи выводятся все строки комментариев, которые написаны в начале каждой функции — как созданной разработчиками системы, так и собственными функциями пользователя.

Команда LOOKFOR

Эта команда позволяет выполнить поиск m-функции по ключевому слову; при этом анализируется первая строка комментария, и она же выводится на экран, если в ней встретилось ключевое слово. Например, в системе MATLAB нет m-функции с именем `inverse`, и поэтому на команду

```
>> help inverse
```

ответом будет

`inverse.m not found` (`inverse.m` не найден).

Однако команда `lookfor inverse` покажет все совпадения, найденные в справочных файлах пакетов прикладных программ, подключеных к пакету MATLAB. Добавление к команде `lookfor` опции `-all` в виде `lookfor <слово> -all` расширяет область поиска — <слово> ищется в первом блоке комментариев, то есть в блоке комментариев между заголовком функции и первым оператором.

Меню HELP

Это меню командного окна системы MATLAB позволяет активизировать следующие окна:

```
HelpWindow

Help Tips

Help Desk(HTML)

Examples and Demos

AboutMATLAB

Subscribe (HTML)
```

Окно справки `HelpWindow` позволяет получить в отдельном окне то же самое, что и команда `help`. Отличие состоит в том, что в этом окне можно погружаться внутрь

раздела с помощью двойного щелчка мыши, а не повторно набирая команду `help` с новыми аргументами.

В пункте меню `Help Tips` приведена краткая справка по использованию помощи, то есть описаны все пункты этого меню.

Вызов пункта меню `Help Desk` позволяет получить доступ к большому объему справочной информации и к документации по системе, размещаемой на жестких дисках, либо на диске CD-ROM в формате HTML. При использовании этого пункта меню желательно, чтобы в системе был установлен какой-либо Internet-браузер (например, Internet Explorer или Netscape Navigator). Все операторы и функции пакета MATLAB описаны подробно и с большим числом примеров. Реализована поисковая система, позволяющая выполнить необходимые запросы.

В пункте меню `Examples and Demos` приведено большое число примеров использования пакета MATLAB для решения задач из разных областей. Пункт меню `AboutMatlab` выводит на экран стандартную заставку MATLAB, а в пункте меню Subscribe HTML предлагается подписаться (при наличии лицензии на продукт) на доступ через Интернет к услугам фирмы MathWorks. Версии справочной документации доступны для просмотра и распечатки в формате PDF с помощью программы Adobe Acrobat. Он позволяет просматривать текст в формате печатной страницы с необходимым набором шрифтов, графики и изображений, с полным ощущением чтения книги. Одновременно это и наилучший способ получения копий нужных страниц.

А.3. РЕДАКТИРОВАНИЕ И ПОВТОРНЫЙ ВЫЗОВ КОМАНДНОЙ СТРОКИ

Командная строка MATLAB легко редактируется. Курсор можно перемещать с помощью стрелок $\Leftarrow$, $\Rightarrow$ и удалять неправильно набранные символы с помощью клавиш Backspace или Delete. При работе на IBM PC совместимом компьютере доступны также и другие редактирующие клавиши: Home, End и Delete. При работе на других системах необходимо использовать команду edit для ознакомления со списком доступных команд. Удобным свойством системы является возможность использовать клавиши-стрелки $\Downarrow$, $\Uparrow$ для доступа к буферу с ранее введенными командами. Это дает возможность осуществить повторный вызов ранее выполненной команды, отредактировать ее и выполнить вновь. Для небольших процедур это гораздо удобнее, чем писать и отлаживать m-файлы, что требует постоянного перехода из командного окна MATLAB в окно текстового редактора (см. Приложение В). Например, если вы хотите сравнить графики функций `sin(mx)` и `sin(nx)` на интервале [0, 2.] для различных значений `n` и `m`, то можете воспользоваться командной строкой

```
>> m=2;n=3;x=0:.01:2*pi;y=sin(m*x);z=sin(n*x); plot(x,y,x,z)
```

А.3.1. ФОРМАТ ВВОДА

Поскольку все вычисления в MATLAB выполняются с двойной точностью, формат вывода может управляться с помощью следующих команд:

`format short`	с фиксированной точкой и 4 знаками после точки
`format long`	с фиксированной точкой и 14 знаками после точки
`format short e`	научная нотация с 4 десятичными знаками
`format long e`	научная нотация с 15 десятичными знаками
`format short g`	с фиксированной точкой и 3 знаками после точки
`format long g`	научная нотация с 13 десятичными знаками

По умолчанию MATLAB использует формат `short`. После вызова одного из приведенных выше форматов он сохраняется до вызова другого формата. Команда `format compact` подавляет большинство пустых строк, позволяя большее количество информации вывести на экран или страницу. Она не зависит от других команд формата.

А.3.2. КОПИЯ ПРОТОКОЛА СЕССИИ

Легче всего протокол сессии получить с помощью команды `diary`. Вызов команды `diary <имя_файла>` приведет к тому, что все появившееся далее на экране (кроме графики) будет записано в файл `<имя_файла>`. Если имя файла в команде будет опущено, то протокол сессии будет записан в файл с именем `diary`. Команда `diary off` приведет к выключению режима записи команд в протокол сессии, а команда `diary on` к восстановлению режима записи. После завершения записи сессии вы можете отредактировать этот файл как обычный текстовой файл и при необходимости распечатать его или использовать для последующего написания m-файла.

А.4. СОЗДАНИЕ МАТРИЦ

MATLAB работает практически с одним видом объектов — с числовыми прямоугольными матрицами, элементами которых могут быть в общем случае комплексные числа. Все переменные представляют собой матрицы. В некоторых случаях матрицы 1×1 интерпретируются как скаляры, а матрицы с одной строкой или одним столбцом интерпретируются как векторы. В MATLAB матрицы могут быть созданы разными способами.

1. Введены явно с помощью списка элементов.
2. Сгенерированы встроенными операторами или функциями.
3. Созданы в m-файлах (см. Приложение В).
4. Загружены из внешнего файла данных.

А.4.1. ЯВНОЕ ЗАДАНИЕ МАТРИЦ

Например, любой из приведенных далее операторов

```
>> A=[1 2 3; 4 5 6; 7 8 9] <Enter>
```

или

```
>> A=[1 2 3 <Shift>+<Enter>
4 5 6 <Shift>+<Enter>
7 8 9] <Enter>
```

создает матрицу 3×3 и присваивает ее значение переменной

```
A =
     1     2     3
     4     5     6
     7     8     9.
```

Элементы внутри строки матрицы могут отделяться друг от друга не только пробелами, но и запятыми. При вводе чисел в экспоненциальной форме (например, $2,34e-94$) нельзя использовать пробелы. Ввод больших матриц лучше выполнять с помощью m-файлов, в которых легко находить и исправлять ошибки (см. Приложение В). В MATLAB встроен ряд функций, позволяющих создавать функции специального вида, например, `rand`, `magic`, `zeros` и `ones` позволяют легко сгенерировать матрицы.

Функция `rand(n)` создает матрицу размером $n \times n$, каждый элемент которой — случайное число с равномерным законом распределения в диапазоне $[0, 1]$.

Функция `rand(m,n)` создает матрицу размера $m \times n$, каждый элемент которой — случайное число с равномерным законом распределения в диапазоне $[0, 1]$.

Функция `magic(n)` создает матрицу размером $n \times n$, которая является магическим квадратом (суммы элементов по строкам и столбцам равны).

Функция `zeros(m,n)` создает нулевую матрицу размера $m \times n$.

Функция `ones(m,n)` создает матрицу размера $m \times n$, каждый элемент которой равен единице.

Матрицы могут быть сгенерированы также с помощью цикла `for`.

Ссылки на отдельные элементы матриц и векторов осуществляются с помощью индексов в круглых скобках обычным образом. Например, `A(1,3)` означает элемент матрицы, стоящий на 1-й строке и 3-м столбце матрицы `A`, а `x(3)` означает 3-й элемент вектора `x`. В качестве индексов векторов и матриц могут использоваться только положительные числа. Ссылаться на элементы матрицы `A` можно, используя единственный индекс `A(k)`. В этом случае данная матрица рассматривается как один длинный вектор-столбец, сформированный из столбцов исходной матрицы. Например, обратиться ко второму элементу второй строки матрицы `A` можно, указав `A(2,2)` или `A(5)`.

А.4.2. ПОДМАТРИЦЫ И ИСПОЛЬЗОВАНИЕ ДВОЕТОЧИЯ

Для записи алгоритмов сложной обработки данных в компактной форме в системе MATLAB используются векторы и подматрицы. Использование нотации с двоеточием (которая используется и для генерации векторов и подматриц) и векторов вместо индексов является ключом к эффективной манипуляции этими объектами. Эффективное использование этих возможностей позволяет минимизировать число явных циклов, использование которых существенно замедляет работу MATLAB, и делает на-

писанную программу простой и легко читаемой. (Правда, овладение данной технологией требует от пользователя пакета определенных усилий.) Например, выражение 1 : 5 фактически является вектором-строкой [12345]. Отметим, элементы вектора могут быть не только целыми, но и действительными.

Например, команда

```
>> x=0.2:0.2:1.2
```

создает вектор

```
x = [0.2, 0.4, 0.6, 0.8, 1.0, 1.2],
```

а команда

```
>> 5:-1:1
```

создает вектор

```
[5 4 3 2 1 ].
```

Для создания и вывода на экран таблицы синусов необходимо выполнить следующую последовательность команд:

```
>> x = [0.0:0.1:2.0]'; % транспонирование вектора
>> y = sin(x); % вычисление вектора, содержащего значения sin
>> [x y] % вывод таблицы значений.
```

Отметим, что так как `sin` является скалярной функцией (то есть функцией, действующей поэлементно), поэтому результатом ее применения к вектору `x` будет вектор `y`.

Для доступа к подматрицам может быть использовано двоеточие. Например, `A(1:2,3)` является вектором-столбцом, состоящим из двух первых элементов третьего столбца матрицы `A`:

```
>> A(2:3,3)
ans =
     6
     9.
```

Двоеточие само по себе означает всю строку или весь столбец, например:

```
>> A(:,3)
ans =

     3
     6
     9

>> A(3,:);
ans =
     7     8     9.
```

В качестве индекса подматрицы может использоваться произвольный целый вектор:

```
>> A(1,[2 3])
ans =
     2     3.
```

Описанные способы индексирования могут использоваться с обеих сторон знака присваивания. Например, после выполнения команды

```
>>  A(:,[2 4 5]) = B(:,1:3)
```

2, 4 и 5-й столбцы матрицы `A` будут заменены на первые три столбца матрицы `B`.

А.4.3. ФУНКЦИИ ПОСТРОЕНИЯ МАТРИЦ

В MATLAB имеются следующие стандартные функции для построения матриц:

`eye(m,n)` — создание единичной матрицы размером $m \times n$;

`zeros(m,n)` — создание нулевой матрицы размером $m \times n$;

`ones(m,n)` — создание матрицы размером $m \times n$, каждый элемент которой равен единице;

`diag(x)` — создание матрицы, у которой на главной диагонали стоят элементы вектора x;

`diag(A)` — создание матрицы, у которой на главной диагонали стоят диагональные элементы матрицы A;

`triu(A)` — создание верхней треугольной матрицы, элементы которой, расположенные на главной диагонали и выше главной диагонали, равны соответствующим элементам матрицы A;

`tril(A)` — создание нижней треугольной матрицы, элементы которой, расположенные на главной диагонали и ниже главной диагонали, равны соответствующим элементам матрицы A.

А.5. ОПЕРАЦИИ, ВЫРАЖЕНИЯ И ПЕРЕМЕННЫЕ

А.5.1. ПРАВИЛА ЗАПИСИ ОПЕРАТОРОВ

Пакет MATLAB является интерпретирующим языком непосредственных вычислений. Это означает, что выражения, которые вы вводите, интерпретируются и вычисляются. Операторы пакета MATLAB обычно имеют форму:

```
>> имя_переменной = выражение
```

или просто

```
>> выражение
```

Выражение, как правило, формируется из операторов, функций и имен переменных. После выполнения выражения генерируется матрица, которая выводится на

экран и присваивается соответствующей переменной для последующего использования. Если имя переменной в левой части и знак = отсутствуют, автоматически генерируется переменная `ans` (answer — ответ), которой присваивается результат вычислений.

Обычно оператор завершается клавишей <Enter>. Однако при необходимости оператор может быть продолжен на следующей строке. Для этого его необходимо завершить тремя или более точками, после которых следует <Enter>. С другой стороны, в одной строке может быть несколько операторов, разделенных запятой или точкой с запятой.

Если последним символом в строке является точка с запятой, то вывод значений результата не производится, но присвоение значения выражения переменной выполняется. Это помогает подавить вывод ненужных промежуточных результатов. Важно помнить, что MATLAB различает строчные и прописные буквы в именах команд, функций и переменных.

Для получения списка всех переменных, расположенных в рабочем пространстве, используется команда `who`. Переменная может быть удалена из рабочего пространства командой `clear <имя_переменной>`. Команда `clear` без аргументов очищает все непостоянные переменные, расположенные в рабочем пространстве. Примером постоянной переменной является переменная `eps` (epsilon), значение которой по умолчанию равно 10^{-6}. Данная переменная используется для оценки точности вычислений итеративных процессов.

Вывод на дисплей или вычисления могут быть прерваны на большинстве компьютеров, не покидая MATLAB, с помощью комбинации клавиш <Ctrl>+<C> (<Ctrl>+<Break> на PC).

А.5.2. МАТРИЧНЫЕ ОПЕРАЦИИ

В MATLAB доступны следующие матричные операции: $< + >$ — сложение;

$< - >$ — вычитание;

$< * >$ — умножение;

$< \hat{} >$ — возведение в степень;

$<'>$ — транспонирование;

$< \backslash >$ — левое деление;

$< / >$ — правое деление.

Данные матричные операции применимы, конечно, и к скалярам — матрицам 1×1. Если размерность матриц не соответствует используемой операции, то система генерирует сообщение об ошибке, за исключением случаев, когда одним из операндов является скаляр, потому что в этом случае операция выполняется между скаляром и каждым элементом матрицы второго операнда.

Отметим, что операция возведения в степень применима только для квадратных матриц.

Если `A` является невырожденной квадратной матрицей, а `b` — вектор-столбец или вектор-строка соответственно, тогда вектор `x = A\ b` является решением уравнения `Ax=b`, а `x=b/A` является решением уравнения `xA=b`. Если `A` — квадратная матрица, то

при левом делении для факторизации используется метод исключения Гаусса. Если матрица не квадратная, то для ее факторизации используется метод ортогонализации Хаусхольдера с ведущим столбцом, а приведенная матрица используется для решения переопределенной системы уравнений в смысле наименьших квадратов. Правое деление определяется в терминах левого деления по формуле `b/A = (A'\b')'`.

А.6. ОПЕРАЦИИ С МАССИВАМИ

Матричные операции сложения и вычитания действуют поэлементно, а остальные приведенные выше операции — нет, они являются матричными операциями. Следует отметить, что приведенные выше операции *, ., \, / могут стать поэлементными, если перед ними поставить точку. Например, команды

```
>> [1,2,3,4].*[1,2,3,4]
```

или

```
>> [1,2,3,4].^2
```

дадут один и тот же результат

```
ans=
[1,4,9,16].
```

А.7. СОХРАНЕНИЕ ДАННЫХ ИЗ РАБОЧЕЙ ОБЛАСТИ

При выходе из системы MATLAB значения всех переменных, находящихся в рабочей области теряются. Для сохранения их значений на диске в нетекстовом формате в файле с именем `matlab.mat` перед выходом из пакета необходимо выполнить команду `save`. Для восстановления ранее сохраненных значений переменных используется команда `load`, восстанавливающая все переменные рабочего пространства.

А.8. ОПЕРАТОРЫ FOR, WHILE, IF, CASE И ОПЕРАТОРЫ ОТНОШЕНИЯ

Операторы управления MATLAB `for`, `while`, `if`, `case` и операторы отношения (`and`, `or`, `xor`, `not`) при использовании в своей основной форме работают так же, как и в большинстве языков программирования.

А.8.1. ЦИКЛ FOR

Например, для данного n оператор

```
>> x = [ ]; for i = 1:5,x=[x,i^2], end
```

```
x =
     1
x =
     1     4
x =
     1     4     9
x =
    1     4     9    16
x =
     1     4     9    16    25
>>
```

создает определенный вектор размерности n, оператор

```
>> x = [ ]; for i = 5:-1:1,x=[x,i^2], end
x =
    25
x =
    25    16
x =
    25    16     9
x =
    25    16     9     4
x =
    25    16     9     4     1
```

создает вектор с теми же элементами, но размещенными в обратном порядке.

Заметим, что матрица может быть пустой (например, в случае оператора `x = [ ]`).

Последовательность операторов

```
>> m=3;
      n=3
      for i = 1:m
         for j = 1:n
            H(i, j) = 1/(i+j-1);
         end
      end
     H
H =
    1.0000    0.5000    0.3333
    0.5000    0.3333    0.2500
    0.3333    0.2500    0.2000
```

создает и выводит на экран матрицу Гильберта размерности $m \times n$.

Точка с запятой, которая завершает внутренний оператор, предотвращает вывод на экран ненужных промежуточных результатов.

А.8.2. ЦИКЛ WHILE

В общем виде цикл `while` записывается в виде

```
while <условие>
   <операторы>
end
```

`<операторы>` будут повторяться до тех пор, пока `<условие>` будет оставаться истинным. Например, для заданного числа `a` приведенная далее последовательность операторов вычислит и выведет на дисплей наименьшее неотрицательное число `n`, такое, что `n^2 < a`:

```
>> n = 0;a=100;
>> while n^2 < a
         n = n + 1;
      end
      n
n=
    10.
```

А.8.3. УСЛОВНЫЙ ОПЕРАТОР IF

В общем виде простой оператор `if` имеет следующий вид:

```
if <условие>
  <операторы1>
else
  <операторы2>
end
```

`<операторы1>` выполняются, если `<условие>` истинно, `<операторы2>` — если <условие> ложно.

В пакете MATLAB возможно также множественное ветвление, например,

```
if n < 0
   parity = 0;
elseif rem(n,2) == 0 % если условие n < 0 не выполняется
   parity = 2; % если число n четное
else
   parity = 1; % если число n нечетное
end
```

При использовании двухвариантного условного оператора часть, связанная с `elseif`, не используется.

А.8.4. ОПЕРАТОР ПЕРЕКЛЮЧЕНИЯ CASE

При необходимости построить конструкцию ветвления с более чем двумя логическими условиями удобнее использовать не вложенные операторы `if`, а оператор переключения `switch ... case`. Этот оператор имеет следующую структуру:

```
switch <выражение> % <выражение> - обязательно скаляр
                                   или строка
case <значение1>
   <операторы1> % <операторы1> выполняются, если
                % <выражение> = <значение1>
case <значение2>
   <операторы2> % <операторы2> выполняются, если
                % <выражение> = <значение2>
...
otherwise
   <операторы>  % <операторы> выполняются,
                % если <выражение> не совпало
                % ни с одним из значений
```

А.9. УСЛОВИЯ (ОПЕРАТОРЫ ОТНОШЕНИЯ)

В MATLAB используются следующие операторы отношения:

< — меньше чем

> — больше чем

<= — меньше или равно

>= — больше или равно

== — равно

. = — не равно

Отметим, что знак = используется в операторах присваивания, в то время как в операторах отношения используется знак ==. Операторы отношения (или, другими словами, логические переменные, которые они создают) могут объединяться с помощью следующих логических операторов:

& — И

| — ИЛИ

. — НЕ

Операторы отношения, примененные к скалярам, также возвращают скаляр, значение которого равняется 1 или 0 в зависимости от того, является ли результат истиной или ложью.

Операторы отношения, примененные к матрицам одного размера, возвращают матрицу того же размера, у которой в качестве элементов стоят 0 или 1, в зависимости от соотношения между соответствующими элементами исходных матриц.

Операторы `while` и `if` интерпретируют отношение между матрицами как истинное в том случае, если результирующая матрица не имеет нулевых элементов.

Например, если необходимо выполнить оператор `<операторы>`, когда матрицы `A` и `B` совпадают, следует использовать оператор соотношения

```
if A == B <операторы> end
```

Если необходимо выполнить оператор в том случае, когда матрицы `A` и `B` не равны, можно использовать следующий оператор

```
if A == B else <операторы> end
```

А.10. ФУНКЦИИ MATLAB

В MATLAB существует большое количество функций, созданных разработчиками системы, большинство из которых предоставлено в виде m-файлов, содержащих исходные тексты. Можно подойти к классификации данных функций различными способами, например по областям их использования (тригонометрические, спецфункции, функции линейной алгебры и т. д.). Далее мы используем подход, основанный на виде аргумента функции: скаляр, вектор, матрица.

А.10.1. СКАЛЯРНЫЕ ФУНКЦИИ

Скалярные функции MATLAB действуют только на скаляры. Если аргументом данных функций является матрица, то они действуют поэлементно. К таким функциям, например, относятся

`sin asin exp abs round`

`cos acos log` (натуральный логарифм) `sqrt floor`

`tan atan rem` (остаток от деления двух чисел) `sign ceil`.

А.10.2. ВЕКТОРНЫЕ ФУНКЦИИ

Аргументами векторных функций являются векторы (строки или столбцы). Если в качестве аргумента функции указана матрица размером `m×n` ($m >= 2$), то данная функция действуют постолбцово, то есть результатом действия является вектор-столбец, каждый элемент которой является результатом действия этой функции на соответствующий столбец. Построчное действие такой функции (если необходимо) может быть достигнуто использованием операции транспонирования.

Названия некоторых из этих функций приведены ниже:

```
max sum median any

min prod mean all

sort std.
```

Например, максимальный элемент прямоугольной матрицы находится с помощью команды `max(max(A))`, а не с помощью `max(A)`, так как результат, возвращенный функцией `max(A)`, — вектор, каждая компонета которого есть максимальный элемент соответствующего столбца.

А.10.3. МАТРИЧНЫЕ ФУНКЦИИ

Наибольшую мощь системе MATLAB дают матричные функции, наиболее употребительные из которых приведены ниже:

`eig` — собственные значения и собственные векторы;

`chol` — факторизация Холецкого;

`svd` — сингулярная декомпозиция;

`inv` — обратная матрица;

`lu` — LU-факторизация;
`qr` — QR-факторизация;
`hess` — вычисление формы Хессенберга;
`schur` — декомпозиция Шура;
`rref` — приведение к треугольной форме методом Гаусса;
`expm` — матричная экспонента;
`sqrtm` — матричный корень квадратный;
`poly` — характеристический полином;
`det` — определитель;
`size` — размерность;
`norm` — норма вектора или матрицы;
`cond` — число обусловленности;
`rank` — ранг матрицы;

Функции MATLAB могут возвращать одновременно несколько переменных. Например, функция

```
y = eig(A),
```

или просто

```
eig(A)
```

генерирует вектор-столбец, содержащий собственные значения матрицы `A`, в то время как оператор

```
[U,D] = eig(A)
```

генерирует матрицу `U`, чьи столбцы являются собственными векторами `A`, а диагональная матрица `D` содержит на главной диагонали собственные значения этой матрицы.

Приложение В

М-ФАЙЛЫ

MATLAB может выполнять последовательность операторов, записанных в файл на диске. Такие файлы называются m-файлами, потому что имена этих файлов имеют вид `<имя>.m`. Большая часть работы в MATLAB будет состоять в создании, редактировании и выполнении таких m-файлов. Существует два типа m-файлов: файлы-программы (или сценарии) и файлы-функции.

В.1. ФАЙЛЫ-ПРОГРАММЫ

Файлы-программы состоят из последовательности обычных операторов пакета MATLAB. Если m-файл с таким сценарием имеет имя, например, `inverse.m`, то команда `inverse`, введенная в командной строке, вызовет выполнение последовательности операторов, находящейся в данном файле. Переменные в программе являются глобальными и изменят значения таких же переменных (если таковые есть) в рабочей области текущей сессии. Программы или сценарии часто используются для ввода данных в большие матрицы; в таких файлах легко исправить ошибки ввода. Если, например, файл на диске с именем `data.m` содержит строки

```
A=[
1 2 3 4
5 6 7 8
];
```

то выполнение команды data приведет к появлению в рабочей области матрицы `A`. Внутри m-файлов можно ссылаться на другие m-файлы, в том числе и рекурсивно на самого себя.

В.1.1. ФАЙЛЫ-ФУНКЦИИ

Файлы-функции фактически дают возможность расширять MATLAB, поскольку определенные пользователем новые функции, специфические для решения конкретных задач, имеют тот же статус, что и другие функции MATLAB. Переменные в функциях являются по умолчанию локальными, но начиная с версии 4.0 и выше разрешено объявлять глобальные переменные (`global`).

Рассмотрим пример файла-функции, находящейся в файле `randint.m`:

```
function z = randint(m,n,a,b)
% randint(m,n,a,b) - функция, возвращающая
% случайную матрицу с
% целыми элементами из диапазона [a,b]
```

```
% randint(m,n) - функция, возвращающая случайную матрицу с целыми
% элементами из диапазона [0,9]

if nargin < 3
   a = 0;
   b = 9;
end

z = floor((b-a+1)*rand(m,n)) + a;
```

Отметим, что совпадение имени функции и имени файла является в пакете MATLAB обязательным условием. Первая строка функции содержит объявление выходных аргументов, имени функции и входные аргументы (списка формальных параметров). Без такой строки весь следующий файл является программой (или сценарием), но не функцией. Так, например, оператор

```
z = randint(4,5)
```

приведет к передаче чисел 4 и 5 переменным `m` и `n`, а выходной результат будет передан переменной `z`. Поскольку переменные в файле-функции локальные, их имена никак не влияют на имена и значения переменных в текущей рабочей области MATLAB.

Символ % указывает на то, что вся строка символов после него является комментарием и игнорируется при исполнении. Тем не менее несколько первых строк-комментариев, которые являются кратким описанием данной функции, доступны при вводе оператора, то есть являются той помощью, которая вызывается командой `help randint`. Как показал опыт работы с версией пакета MATLAB 6.0—6.2, при использовании в комментариях прописной буквы «Я» возникает ошибка периода исполнения. Выход из данной ситуации состоит в использовании строчной (заглавной) буквы «я», что позволяет избежать указанной ошибки. В более поздних версиях указанная проблема была устранена.

Отметим, что здесь мы использовали функцию `nargin`, возвращающую длину списка формальных параметров функции. Использование функции `nargin` в данном примере позволяет установить значение отсутствующих входных аргументов по умолчанию переменных `a` и `b`. В общем случае наличие такой функции позволяет использовать функции с переменным числом входных аргументов и в зависимости от их числа направлять вычисления по разным логическим веткам функции.

Функция может иметь множественные выходные аргументы, например:

```
function [mean, stdev] = STAT(x)
% STAT - функция, возвращающая для вектора x среднее значение и
% стандартное отклонение
% функция STAT возвращает для матрицы x две строки,
% первая из которых содержит средние значения,
% вторая - стандартные
% отклонения соответствующих столбцов матрицы x

[m n] = size(x);
if m == 1
```

```
   m = n; % если входные данные вектор
end

mean = sum(x)/m;
stdev = sqrt(sum(x.^2)/m - mean.^2);
```

Данный файл следует записать на диск под именем `stat.m`. Если ввести команду

```
>> [xm,xd]=stat(x),
```

то переменным `xm`, `xd` будут присвоены среднее значение и стандартное отклонение элементов вектора `x`.

Несмотря на наличие нескольких выходных аргументов, можно присвоить значение функции одной переменной. Например, `xm = stat(x)` (никаких скобок вокруг `xm` не требуется) присвоит `xm` среднее значение `x`.

Отметим оператор `x.^2`, который является возведением в степень каждого элемента вектора `x`, оператор с функцией `sum`, которая является векторной функцией (см. Приложение А), функцию `sqrt`, которая является скалярной функцией (см. Приложение А) и деление в выражении `sum(x)/m`, которое является матрично-скалярной операцией.

Другим примером является функция, позволяющая вычислить наибольший общий делитель двух чисел:

```
function [b, steps] = bisect(fun, x, tol)
% функция, возвращающая решение
% уравнения fun(x)=0 методом
% половинного деления
% входные переменные:
% fun - строка, содержащая имя
% вещественной функции, зависящей от
% вещественной переменной
% x - начальное приближение, используемое
% для поиска корня
% tol - переменная, определяющая точность
% нахождения корня
% выходные переменные:
% b - значение корня
% step - матрица, содержащая значения корня
% и соответствующие значения
% функции на концах отрезка
% на каждом шаге вычислительного процесса
if nargin < 3
   tol = eps;
end; % проверка длины списка формальных
     % параметров, если значение переменной tol
     % не указано, то точность корня
     % определяется встроенной переменной eps
```

```
trace = (nargout == 2); % проверка длины списка
                        % выходных параметров,
                        % если список состоит из
                        % двух переменных, то trace = 1
% выбор шага изменения переменной x
if x ~= 0
   dx = x/20;
else
   dx = 1/20;
end;
a = x - dx; % левый конец отрезка
fa = feval(fun,a); % значение функции
                   % в левом конце отрезка
b = x + dx; % правый конец отрезка
fb = feval(fun,b); % значение функции
                   % в правом конце отрезка
% поиск отрезка, на котором
% функция меняет знак
while (fa > 0) == (fb > 0)
   dx = 2.0*dx;
   a = x - dx;
   fa = feval(fun,a);
   if (fa > 0) ~= (fb > 0)
      break
   end;
   b = x + dx;
   fb = feval(fun,b);
end;
if trace
   steps = [a fa; b fb];
end;

% основной цикл
while abs(b - a) > 2.0*tol*max(abs(b),1.0)
    c = a + 0.5*(b - a);
    fc = feval(fun,c);
    if trace
       steps = [steps; [c fc]];
    end;
    if (fb > 0) == (fc > 0)
       b = c; fb = fc;
    else
       a = c; fa = fc;
    end;
end;
```

Отметим, что некоторые функции MATLAB являются встроенными, в то время как другие поставляются в виде m-файлов. Текст реально имеющихся m-файлов (пакета MATLAB или ваших собственных) можно просмотреть с помощью команды `type <имя_функции>`.

В.1.2. ТЕКСТОВЫЕ СТРОКИ, СООБЩЕНИЯ ОБ ОШИБКАХ

Текстовые строки вводятся в MATLAB в виде текста в одинарных кавычках. Например, оператор

```
s = 'This is a test'
```

присваивает данный текст переменной `s`.

Вывод текстовой строки осуществляется с помощью оператора `disp`. Например, оператор `disp(s)` выведет на экран сообщение:

```
This is a test
```

Сообщения об ошибках лучше выводить с помощью функции `error`, так как после обращения к данной функции выполнение m-файла будет прекращено. Например, если в процессе выполнения m-файла будет выполнен оператор

```
error('Sorry, the matrix must be symmetric'),
```

то после вывода сообщения на экран выполнение m-файла будет прекращено.

В m-файле может быть запрос на интерактивный ввод данных, организованный с помощью оператора `input`. После ввода оператора

```
iter = input('Введите число итераций: ')
```

на экран выводится запрос на ввод, и выполнение программы приостанавливается до того момента, пока пользователь не введет с клавиатуры требуемые входные данные. После нажатия клавиши <Enter> данные присваиваются переменной `iter`, и выполнение программы будет продолжено.

В.2. РАБОТА С М-ФАЙЛАМИ

Во время работы в MATLAB часто возникает необходимость создавать или редактировать m-файлы, после чего возвращаться в командное окно пакета MATLAB для отладки и/или вычислений. Начиная с версии 5.0 (в ОС Windows-95) в пакете имеется специальный редактор/отладчик, в котором можно исправлять текст m-файлов и выполнять пошаговую отладку программы. Для работы с редактором/отладчиком необходимые m-файлы должны быть доступны. Для этого либо текущая директория должна быть директорией с вашими файлами, либо необходимо проложить туда путь (в смысле MS DOS). Это можно сделать либо с помощью команд MS DOS непосредственно из командного окна (команда `cd`), либо с помощью пункта меню File/Set Path, который позволяет установить путь к соответствующим папкам.

В.2.1. СПИСОК ПУТЕЙ ДОСТУПА К ФАЙЛАМ

Для поиска m-файлов система MATLAB использует механизм путей доступа, поскольку m-файлы записываются в каталоги или папки файловой системы. Например, при поиске файла с именем test пакет MATLAB выполняет следующие действия:

1) просматривает, не является ли test именем переменной;

2) просматривает, не является ли test встроенной функцией;

3) ищет в текущем каталоге m-файл с именем test.m;

4) ищет m-файл с именем test.m во всех каталогах списка путей доступа.

Реально применяемые правила поиска являются более сложными из-за ограничений, которые связаны с использованием подфункций[1], личных (private) функций и объектно-ориентированных механизмов. Однако приведенный выше упрощенный порядок поиска точно отражает механизм поиска m-файлов, с которыми обычно работает пользователь.

Работа со списком путей доступа. В процессе сеанса работы можно вывести на терминал или внести изменения в список путей доступа, используя следующие функции:

- `path` — выводит на экран список путей доступа;
- `path(s)` — заменяет существующий список списком s;
- `addpath/home/lib` и `path(path,'/home/lib')` — добавляют новый каталог текущего подкаталога в список путей доступа;
- `rmpath/home/lib` — удаляет путь `/home/lib` из списка.

Список путей доступа, используемый по умолчанию, определен в файле `pathdef.m`, который размещен в каталоге `local`; этот файл выполняется при каждом запуске пакета MATLAB.

Кроме работы из командной строки существует средство просмотра путей доступа `Path Browser` (см. далее), которое поддерживает удобный графический интерфейс для просмотра и изменения списка путей.

Текущий каталог. Пакет MATLAB использует понятие текущего каталога при работе с m- и mat-файлами во время сеанса работы.

Начальный текущий каталог определен в файле запуска, который ассоциирован с ярлыком запуска пакета MATLAB, расположенном на рабочем столе.

Для вывода текущего каталога на экран терминала предназначена команда `cd`. Для изменения текущего каталога следует использовать команду

```
cd <новый путь доступа>
```

[1] Внутри функций допускается определение других функций, при этом следует иметь в виду, что доступ к таким функциям возможен только из функций, внутри которых они определены.

Средство просмотра и редактирования путей доступа Path Browser. Как было указано выше, при работе в системе Windows-95 имеется специальное средство для просмотра и изменения путей доступа

```
Path Browser
```

Рис. В.1. Окно `Path Browse`

Интерфейс данного средства, представленный на рис. В.1, открывается либо из меню

```
File/Set Path
```

командного окна, либо с помощью кнопки на инструментальной панели.

После дополнения списка путей доступа необходимо сохранить новый путь с помощью пункта меню `File/Save Path`, в противном случае установленный путь будет известен системе только на время одного сеанса работы.

В.2.2. ИСПОЛЬЗОВАНИЕ РЕДАКТОРА/ОТЛАДЧИКА

Редактор/отладчик предоставляет как средства редактирования текста m-файла, так и средства пошаговой его отладки. Один из способов вызова редактора — вызов из командной строки MATLAB с помощью команды edit. Например, команда edit test откроет встроенный редактор для редактирования файла `test.m`, если в меню `File` в диалоговом окне `Preferences` не установлен вами другой редактор.

Можно открыть редактор и другим способом — с помощью меню

```
File/New
```

или кнопки `New File` на панели инструментов (см. далее). Для открытия существующего m-файла выберите пункт `File/Open` или щелкните на кнопке `Open File`.

После вызова редактор/отладчик будет иметь вид, показанный на рис. В.2.

Рис. В.2. Общий вид редактора/отладчика

Редактор, используемый в системе, имеет синтаксическую раскраску, то есть слово или символ по мере ввода приобретают тот цвет, который соответствует их типу.

Редактор различает такие типы вводимых слов:

- комментарии;
- ключевые слова;
- незаконченные строки;
- законченные строки;
- другой текст.

С помощью пункта меню `Tools/Fonts` можно настроить такие важные параметры, как используемый шрифт. Это особенно важно для работы с русским текстом, поскольку не все шрифты правильно воспроизводят русский текст. В остальном данный редактор не отличается от обычного многооконного текстового редактора — в нем работают все редактирующие клавиши (`Del`, `Bspace`, `Home` и т. д.).

При редактировании файлов вы можете непосредственно перейти к требуемой строке при помощи пункта меню `Edit/GoTo Line` и указать номер требуемой строки в

появившемся окне. После редактирования файла и повторного его запуска из командного окна желательно предварительно сохранить новый вариант файла. Но можно запускать редактируемый файл на счет, не выходя из редактора (то есть не переходя в командное окно), с помощью пункта меню Debug ⇒ `Save` and `Run`. При этом предварительное сохранение текста исправлений не требуется.

Одна из важных особенностей данного редактора состоит в том, что после проведения вычислений можно в редакторе просмотреть значения переменных, которые они имеют в текущий момент в рабочей области (рис. В.3). Для этого достаточно установить курсор мыши на этой переменной, и появится прямоугольник с желтым фоном, на котором выводится текущее значение переменной. Если переменная представляет из себя большую матрицу, то таким образом увидеть целиком ее не удастся. Для просмотра (и возможного исправления при отладке) всех значений матрицы необходимо перейти в окно `Workspace` и выполнить двойной клин по имени строки, содержащей имя выбранной переменной.

Рис. В.3. Окно просмотра и редактирования используемых переменных

В.2.3. ОТЛАДКА М-ФАЙЛОВ

Отладка программного кода — это процесс, в ходе которого могут быть выявлены ошибки двух видов:

1) синтаксические, связанные с неточностью записи имен m-функций и/или арифметических выражений. Пакет MATLAB обнаруживает большинство синтак-

сических ошибок, сопровождая их сообщением об ошибке с указанием номера строки соответствующего m-файла;

2) ошибки периода выполнения, связанные в первую очередь с ошибками алгоритма и приводящие к непредвиденным результатам.

Как показывает опыт работы, достаточно легко можно исправить синтаксические ошибки, которые сопровождаются сообщениями о причинах их возникновения. Ошибки времени выполнения выявить более сложно, потому что локальная рабочая область m-функции оказывается потерянной, если ошибка приводит к возврату в рабочую область системы пакета MATLAB. Для выявления причин данной ошибки можно использовать один из следующих приемов:

1. Реализовать вывод результатов промежуточных вычислений на дисплей, удалив в соответствующих операторах точки с запятой, которые подавляют вывод на экран промежуточных результатов.

2. Добавить в m-файл команды `keyboard`, которые останавливают выполнение m-файла и разрешают проверить и изменить переменные рабочей области вызываемой m-функции. (В этом командном режиме появляется специальное приглашение «K». Для просмотра значений доступных переменных можно использовать окно `Workspace`, или ввести имя соответствующей переменной в командной строке и нажать клавишу <Enter>. Возврат к выполнению функции реализуется командой `return`.)

3. Закомментировать заголовок функции и выполнить m-файл как сценарий. (Это позволяет проследить результаты промежуточных вычислений в рабочей области системы.)

4. Использовать отладчик системы MATLAB.

Отладчик полезен для исправления ошибок во время выполнения программы, так как он дает возможность отслеживать рабочие области функции и проверять или изменять значения соответствующих переменных. Отладчик позволяет устанавливать и удалять контрольные точки, то есть специальным образом помеченные строки m-файла, в которых выполнение останавливается. Это дает возможность изменять содержимое рабочей области, просматривать стек вызова m-функций и выполнять m-файл построчно. Отладчик может функционировать как в режиме командной строки, так и в режиме графического интерфейса пользователя (рис. В.4). Далее мы рассмотрим отладку только в режиме графического интерфейса пользователя, так как он наиболее прост и нагляден.

Рассмотрим возможности отладки, которые нам предоставляет

`Editor/Debuger`

Для его запуска используется команда `edit <имя_файла>` или пункт меню `File/Open`. Можно открыть окно редактора/отладчика и с помощью пункта меню `File/New/M-file`. При таком варианте имя отлаживаемого файла открывается уже из меню самого редактора/отладчика. Общий вид окна редактора показан на рис. В.2,

Рис. В.4. Кнопки основной панели редактора/отладчика

а кнопки панели инструментов, которые удобнее всего использовать при отладке, показаны на рис. В.4.

Способ использования этих кнопок понятен из их названия. Часть кнопок представляют собой обычные редактирующие кнопки, используемые при сохранении, копировании, печати и поиске файлов в программных продуктах, работающих под управлением ОС Windows. Другая часть связана непосредственно с отладкой. Это кнопки установки и очистки точек остановки, кнопки пошагового перемещения по программе (`Step`) с заходом в подпрограммы (`Step in`) и без захода в подпрограммы (`Step out`), кнопка начала вычислений (`Run`) и кнопка остановки отладки (`Exit Debugg Mode`).

Приложение С

СОЗДАНИЕ ГРАФИЧЕСКОГО ИНТЕРФЕЙСА ПОЛЬЗОВАТЕЛЯ

В настоящее время одним из непреложных атрибутов любой прикладной программы является интерактивный интерфейс пользователя, который разрабатывается для неоднократно решаемых задач с несколькими входными параметрами. Наличие интерфейса, во-первых, освобождает пользователя от необходимости вникать в детали программирования данного программного продукта, во-вторых, позволяет менять исходные данные, не прекращая выполнения текущей программы, и в реальном времени наблюдать за изменениями решения задачи.

В настоящее время выработаны достаточно общие требования, которым должны соответствовать программные интерфейсы, определены основные элементы управления графическим интерфейсом. Например, если речь идет о решении учебной и/или научной задачи, обязательными элементами интерфейса должны быть:

1. Одно или несколько окон для визуализации результатов расчетов.

2. Редактируемые окна, позволяющие задавать значения входных параметров задачи.

3. Управляющие кнопки, предназначенные для запуска и остановки процесса расчета, вывода результатов, завершения работы с программой.

4. Поясняющие надписи (статический текст).

Необходимо отметить, что разработка графического интерфейса является достаточно трудоемкой задачей, которая может быть решена с использованием функций графического ввода, а также специальной функцией `uicontrol`, разработанной для интерактивного взаимодействия пользователя с рисунком. Однако для упрощения процедуры создания интерфейса пользователя в пакете MATLAB, как и во всех современных программных средствах, основанных на объектно-ориентированной парадигме программирования, существует возможность визуального проектирования интерфейса, называемая технологией визуального программирования. В данном приложении мы познакомимся с основными этапами технологии визуального программирования пакета MATLAB на примере создания интерфейса задачи моделирования относительных движений материальной точки, участвующей в двух круговых движениях (глава 1). Напомним, что в данной задаче приходится задавать значения переменных `R1` — радиус первой окружности; `T1` — период обращения по первой окружности; `R2` — радиус второй окружности; `T2` — период обращения по второй окружности; `Tfin` — правая граница временного интервала, на котором вычисляются кинематические характеристики; `N` — число точек, в которых проводится вычисление значений кинематических характеристик материальных точек.

Перед созданием графического интерфейса желательно разработать его проект, то есть изобразить на листе бумаги примерный вид вашего интерфейса. Для рассматриваемой задачи интерфейс должен содержать пять окон для ввода значений переменных `R1`, `T1`, `R2`, `T2`, `Tfin`, `N` и, соответственно, пять поясняющих надписей к каждому полю ввода значения переменной, три кнопки, позволяющие завершить работу с программой, построить траекторию движения материальной точки и мгновенные значения разности относительной скорости материальной точки, движущейся по окружности меньшего радиуса.

Для создания графического интерфейса (GUI — Graphics User Interface) с помощью средств визуального программирования необходимо запустить соответствующую программу командой `guide` или через меню: `File` ⇒ New ⇒ GUI. По истечении некоторого времени, определяемого быстродействием компьютера, на экране появится окно, представленное на рис. С.1.

Рис. С.1. Интерфейс средства создания GUI

Для ввода названия формы открыть окно `Property Inspector`, нажав на кнопку № 1 (рис. С.1) или сделав двойной клик в любом месте поля формы, затем установить новое значение поля Name (рис. С.2).

В левой части окна расположена панель управления, содержащая элементы интерфейса. Для размещения на панели формы окна, в котором будут выводиться графики, используется кнопка № 2 (рис. С.1). Щелкнув по данному элементу на панели

Рис. С.2. Инспектор свойств объекта GUI

управления и переведя мышь на панель формы, необходимо поместить указатель мыши, имеющий форму креста, в ту точку, где будет находиться левый верхний угол окна. Нажав и удерживая левую кнопку мыши, необходимо вытянуть получающийся прямоугольник до нужных размеров. При необходимости размещения нескольких окон на панели формы повторяется описанная выше последовательность действий.

Надписи на панели формы создаются с помощью кнопки № 3 (рис. С.1), которая переносится повторением описанной выше последовательности действий. После этого на панели формы появится надпись `Static Text`. Для изменения надписи необходимо выбрать надпись, щелкнув по ней левой кнопкой мыши, открыть окно `Property Inspector` и ввести новый текст надписи в поле `String`. Для изменения цвета фона, на котором выводится надпись, следует установить новые значения поля `BackgroundColor`.

Для создания редактируемых окон ввода используется кнопка № 4 (рис. С.1), которая переносится повторением описанной выше последовательности действий. Для изменения значения окна установить в окне `Property Inspector` новое значение в поле `String`.

Для создания и размещения кнопок на панели формы используется кнопка № 5 (рис. С.1), которая переносится повторением описанной выше последовательности действий. Для изменения надписи на кнопке установить в окне `Property Inspector` новое значение в поле `String`.

Построенные таким образом кнопки, окна вывода и редактирования, окна статического текста, кнопки, а также другие объекты, размещенные на панели формы, можно выровнять и установить определенные промежутки между ними с помощью панели `Alignment Tools`. Для включения данной панели используется кнопка № 6 (рис. С.1). Для задания ряда объектов, с которыми будут выполняться какие-либо действия, необходимо их выделить, щелкая по каждому из них при нажатой клавише `Shift`. Выделенные объекты отмечаются черными точками вокруг соответствующих объектов. При необходимости изменить размер какого-либо объекта, размещенного

на панели формы необходимо щелкнуть по данному объекту левой кнопкой мыши и изменить размер, так же как меняется размер любого окна Windows.

Так как по умолчанию размер окна устанавливается в пикселях, это может привести к тому, что при изменении размеров окна и кнопки могут наехать друг на друга. Для устранения этого нежелательного явления следует задать единицы измерения размера всех элементов в безразмерных переменных — долях размера окна. Данный размер называется `normalized`. Для этого необходимо выделить с помощью удерживаемой клавиши `Ctrl` и левой кнопки мыши все объекты, размещенные на форме. Затем открыть окно `Property Inspector` и установить в поле `Units` значение `normalized`. Аналогично следует задать размер шрифтов — значения поля `Font Units`.

После разработки внешнего вида графического интерфейса необходимо сохранить этот файл на диске, вызвав стандартное окно Windows для сохранения файла: `File` ⇒ `Save`. После сохранения на диске появляются два файла: `NAME.m` и `NAME.fig` (где `NAME` — выбранное вами имя файла). Первый файл — это текст программы, реализующий разработанный интерфейс, второй — набор данных, необходимых для работы интерфейса. Для запуска программы на выполнение достаточно в командной строке MATLAB ввести имя вашего файла.

Проект графического интерфейса решения рассматриваемой в главе 1 задачи представлен на рис. С.3. Ниже приведен соответствующий программный код, сохраненный на диске под именем `Moon.m`.

Рис. С.3. Проект GUI для задачи об описании относительных круговых движений

```
% листинг файла Moom.m
function varargout = moon(varargin)
% MOON Application M-file for moon.fig
% FIG = MOON launch moon GUI.
% MOON('callback_name', ...) invoke the named callback.

% Last Modified by GUIDE v2.0 16-May-2002 23:30:06

if nargin == 0  % LAUNCH GUI

fig = openfig(mfilename,'reuse');

% Generate a structure of handles to pass to callbacks,
    % and store it.
handles = guihandles(fig);
guidata(fig, handles);

if nargout > 0
varargout{1} = fig;
end

elseif ischar(varargin{1}) % INVOKE NAMED SUBFUNCTION
                           % OR CALLBACK

try
[varargout{1:nargout}] = ...
        feval(varargin{:}); % FEVAL switchyard
catch
disp(lasterr);
end
end

%| ABOUT CALLBACKS:
%| GUIDE automatically appends subfunction prototypes
%| to this file, and
%| sets objects' callback properties to call
%| them through the FEVAL
%| switchyard above. This comment describes that mechanism.
%|
%| Each callback subfunction declaration has
%| the following form:
%| <SUBFUNCTION_NAME>(H, EVENTDATA, HANDLES, VARARGIN)
%|
%| The subfunction name is composed using the object's
%| Tag and the
```

```
%| callback type separated by '_', e.g. 'slider2_Callback',
%| 'figure1_CloseRequestFcn', 'axis1_ButtondownFcn'.
%|
%| H is the callback object's handle (obtained using GCBO).
%|
%| EVENTDATA is empty, but reserved for future use.
%|
%| HANDLES is a structure containing handles of components
%| in GUI using
%| tags as fieldnames, e.g. handles.figure1,
%| handles.slider2. This
%| structure is created at GUI startup using GUIHANDLES
%| and stored in
%| the figure's application data using GUIDATA. A copy
%| of the structure
%| is passed to each callback.  You can store additional
%| information in
%| this structure at GUI startup, and you can
%| change the structure
%| during callbacks.  Call guidata(h, handles) after
%| changing your
%| copy to replace the stored original so that
%| subsequent callbacks see
%| the updates. Type "help guihandles" and "help guidata"
%| for more
%| information.
%|
%| VARARGIN contains any extra arguments you have passed to the
%| callback. Specify the extra arguments by editing the callback
%| property in the inspector. By default, GUIDE sets
%| the property to:
%| <MFILENAME>('<SUBFUNCTION_NAME>', gcbo, [], guidata(gcbo))
%| Add any extra arguments after the last argument,
%| before the final
%| closing parenthesis.

% ---------------------------------------------------
function varargout = edit1_Callback(h, eventdata,...
                                    handles, varargin)
% Stub for Callback of the uicontrol handles.edit1.
disp('edit1 callback not implemented yet .')

% ---------------------------------------------------
function varargout = edit2_Callback(h, eventdata,...
                                   handles, varargin)
```

```
% Stub for Callback of the uicontrol handles.edit2.
disp('edit2 callback not implemented yet .')

% ---------------------------------------------------
function varargout = edit3_Callback(h, eventdata,...
                                    handles, varargin)
% Stub for Callback of the uicontrol handles.edit3.
disp('edit3 callback not implemented yet .')

% ---------------------------------------------------
function varargout = edit4_Callback(h, eventdata,...
                                    handles, varargin)
% Stub for Callback of the uicontrol handles.edit4.
disp('edit4 callback not implemented yet .')

% ---------------------------------------------------
function varargout = edit5_Callback(h, eventdata,...
                                    handles, varargin)
% Stub for Callback of the uicontrol handles.edit5.
disp('edit5 callback not implemented yet .')

% ---------------------------------------------------
function varargout = edit6_Callback(h, eventdata,...
                                    handles, varargin)
% Stub for Callback of the uicontrol handles.edit6.
disp('edit6 callback not implemented yet .')

% -------------------------------------------------------
function varargout = pushbutton1_Callback(h, eventdata,...
                                          handles, varargin)
% Stub for Callback of the uicontrol handles.pushbutton1.
disp('pushbutton1 callback not implemented yet .')

% -------------------------------------------------------
function varargout = pushbutton2_Callback(h, eventdata,...
                                          handles, varargin)
% Stub for Callback of the uicontrol handles.pushbutton2.
disp('pushbutton2 callback not implemented yet .')

% -------------------------------------------------------
function varargout = pushbutton3_Callback(h, eventdata,...
                                          handles, varargin)
% Stub for Callback of the uicontrol handles.pushbutton2.
disp('pushbutton3 callback not implemented yet .')
```

Созданный GUI, текст которого приведен выше, не обладает никакой функциональностью — он обладает только стандартными функциями Windows — менять размеры окна и закрывать окно и выводить сообщения о некоторых событиях (команды `disp`). Событие с точки зрения интерфейса происходит при нажатии на левую кнопку мыши. Сообщения, выводимые командой `disp` в командное окно пакета MATLAB, призваны помочь разработчику интерфейса установить взаимно однозначное соответствие между элементом управления, размещенного на панели формы, и именем соответствующей ему функции.

Основным средством взаимодействия GUI с функциями, выполняющими требуемые действия, является задание значений полей, определяющих свойства кнопок и редактируемых окон, называемое `Callback`. Значение (тип строка), присваемое данному полю данному полю, есть имя функции, вызываемой при активации соответствующего объекта. Обратите внимание на тот факт, что в программном коде, сгенерированном редактором `GUI`, название каждой функции состоит из двух слов: названия того элемента, который вы разместили на панели формы, и названия средства взаимодействия `Callback`.

Дальнейшая доработка интерфейса состоит в редактировании соответствующих функций в тексте исходного файла, осуществляемых в текстовом редакторе пакета MATLAB. Так как перед выполнением каждого вычисления программа должна получить значения переменных `R1`, `T1`, `R2`, `T2`, `Tfin`, `N`, представляется целесообразным создать специальную функцию, считывающую значения соответствующих полей (строковые переменные), и преобразующую их в числовые значения, текст которой приводится ниже:

```
function read_data(handles)
global R1 T1 R2 T2 Tfin N % задание глобальных переменных
R1 = str2double(get(handles.edit1,'String')); % преобразование
                                              % значения поля
                                              % edit1 в число
T1= str2double(get(handles.edit2,'String')); % преобразование
                                             % значения поля
                                             % edit2 в число
R2 = str2double(get(handles.edit3,'String')); % преобразование
                                              % значения поля
                                              % edit3 в число
T2 = str2double(get(handles.edit4,'String')); % преобразование
                                              % значения поля
                                              % edit4 в число
Tfin = str2double(get(handles.edit5,'String')); % преобразование
                                                % значения поля
                                                % edit5 в число
N = str2double(get(handles.edit6,'String')); % преобразование
                                             % значения поля
                                             % edit6 в число
```

Кнопка `Выход` становится функционирующей после добавления в функцию `pushbutton1_Callback` команды `close`:

```
function varargout = pushbutton1_Callback(h, eventdata,...
                                          handles, varargin)
% Stub for Callback of the uicontrol handles.pushbutton1.
close;
```

При решении рассматриваемой задачи мы проводили два вида расчетов:

1) вычисление и визуализации траектории;

2) вычисление мгновенных значений скорости движения Луны относительно Солнца. Выполнение первого вида расчетов при проектировании интерфейса мы связали с кнопкой Траектория (функция `pushbutton1_Callback`).

Следовательно, необходимо дополнить функцию `pushbutton1_Callback` соответствующей последовательностью команд, описанных в разделе 2 главы 1:

```
function varargout = pushbutton2_Callback(h, eventdata,...
                                           handles, varargin)
% Stub for Callback of the uicontrol handles.pushbutton2.
global R1 T1 R2 T2 Tfin N % список глобальных переменных
                          % задается в теле
                          % каждой функции
read_data(handles); % получение значений переменных
                    % R1, T1, R2, T2, Tfin, N
% блок вычислений и построения траектории
t=0:Tfin/N:Tfin;
Xz=R1*cos(2*pi*t/T1);
Yz=R1*sin(2*pi*t/T1);
Xm=R2*cos(2*pi*t/T2);
Ym=R2*sin(2*pi*t/T2);
Xotn=Xz+Xm;
Yotn=Yz+Ym;
plot(Xz,Yz,Xotn,Yotn);
```

Выполнение второго вида расчетов при проектировании интерфейса мы связали с кнопкой `Относительная скорость` (соответственно, с функцией `pushbutton2_Callback`), поэтому необходимо дополнить функцию `pushbutton2_Callback` соответствующей последовательностью команд, описанных в разделе 2 главы 1:

```
function varargout = pushbutton3_Callback(h, eventdata,...
                                          handles, varargin)
% Stub for Callback of the uicontrol handles.pushbutton3.
global R1 T1 R2 T2 Tfin N % список глобальных
                          % переменных задается в теле
                          % каждой функции
read_data(handles); % получение значений переменных
                    % R1, T1, R2, T2, Tfin, N
% блок вычислений и построения мгновенных значений
```

```
% относительной скорости
dt=Tfin/N;
t=0:dt:Tfin;
Xz=R1*cos(2*pi*t/T1);
Yz=R1*sin(2*pi*t/T1);
Xm=R2*cos(2*pi*t/T2);
Ym=R2*sin(2*pi*t/T2);
Vx=diff(Xz)/dt;
Vy=diff(Yz)/dt;
vx=diff(Xm)/dt;
vy=diff(Ym)/dt;
V=(Vx.^2+Vy.^2).^0.5-...
      (Vx.*vx+Vy.*vy)./(Vx.^2+Vy.^2).^0.5;

t1=0:dt:Tfin-dt;
plot(t1,V);
```

Состояние интерфейса после нажатия, соответственно, на кнопки `Траектория` и `Относительная скорость` представлено на рис. С.4, С.5.

Рис. С.4. Визуализация траектории движения

Рис. C.5. Визуализация модуля относительной скорости

Приложение D

ОЦЕНКИ ПОГРЕШНОСТЕЙ ЧИСЛЕННОГО ИНТЕГРИРОВАНИЯ

Для нахождения зависимостей погрешности вычисления определенного интеграла на отрезке $[a, b]$ от числа отрезков разбиения интервала интегрирования разложим подынтегральную функцию в ряд Тейлора:

$$f(x) = f(x_i) + f'(x_i)(x - x_i) + \frac{1}{2} f''(x_i)(x - x_i)^2 + \ldots \tag{D.1}$$

Тогда интеграл от данной функции на отрезке $[x_i, x_{i+1}]$ будет равен

$$\int\limits_{x_i}^{x_{i+1}} f(x)dx = f(x_i)\Delta x + \frac{1}{2} f'(x_i)(\Delta x)^2 + \frac{1}{6} f''(x_i)(\Delta x)^3 + \ldots \tag{D.2}$$

Оценим погрешность метода левых прямоугольников. Погрешность интегрирования Δ_i на отрезке $[x_i, x_{i+1}]$ равняется разности между точным значением интеграла и его оценкой $f(x_i)\Delta x$:

$$\Delta_i = \left[\int\limits_{x_i}^{x_{i+1}} f(x)\, dx \right] - f(x_i)\Delta x \approx \frac{1}{2} f'(x_i)(\Delta x)^2. \tag{D.3}$$

Из (D.3) видно, что основной член погрешности на каждом отрезке имеет порядок $(\Delta x)^2$ или в символической записи $O((\Delta x)^2)$. Поскольку полное число отрезков равно N, а $\Delta x = (b - a)/N$, то полная погрешность метода левых прямоугольников по порядку величины равна $N\Delta \approx N(\Delta x)^2 \approx O(N^{-1})$. Аналогично можно показать, что погрешность метода правых прямоугольников также пропорциональна N^{-1}.

Погрешность формулы трапеций оценивается аналогичным образом. Так как значение интеграла на отрезке $[x_i, x_{i+1}]$ вычисляется по формуле $[f(x_i) + f(x_{i+1})]\Delta x/2$, то погрешность равна

$$\Delta_i = \left[\int\limits_{x_i}^{x_{i+1}} f(x)\, dx \right] - \frac{1}{2}[f(x_i) + f(x_{i+1})]\Delta x. \tag{D.4}$$

Заменив в (D.4) первый член выражением (D.2), а значение функции в точке x_{i+1} разложением в ряд Тейлора:

$$f(x_{i+1}) = f(x_i) + f'(x_i)\Delta x + \frac{1}{2} f''(x_i)(\Delta x)^2 + \ldots,$$

раскрыв скобки и приведя подобные, обнаруживаем, что член, пропорциональный первой производной функции, сокращается, и погрешность на одном отрезке равна

$$\frac{1}{4}f''(x)(\Delta x)^3 \approx O\left((\Delta x)^3\right) \approx O\left(N^{-3}\right).$$

Следовательно, полная погрешность формулы трапеций на отрезке $[a, b]$ по порядку величины равна $O(N^{-2})$.

Так как формула Симпсона основывается на приближении функции $f(x)$ параболой, можно ожидать, что в данном случае погрешность по порядку величины будет определяться членами, пропорциональными третьей производной функции. Однако последовательное повторение действий, выполненных при оценке погрешности метода трапеций, показывает, что эти члены сокращаются в силу их симметричности, поэтому в разложении в ряд Тейлора следует удержать член, пропорциональный $f''''(x)(\Delta x)^4$. Следовательно, погрешность формулы Симпсона на отрезке пропорциональна $[x_i, x_{i+1}]$, а полная погрешность на отрезке по порядку величины составляет $O(N^{-4})$.

Получим оценку погрешности вычисления интеграла от функции, зависящей от двух переменных, который с геометрической точки зрения представляет собой объем фигуры под поверхностью, заданной функцией $f(x, y)$. В прямоугольном приближении данный интеграл равен сумме объемов параллелепипедов с площадью основания и высотой, равной значению функции $f(x, y)$ в одном из углов. Для определения погрешности разложим функцию в ряд Тейлора:

$$f(x, y) = f(x_i, y_i) + f'_x(x_i, y_i)(x - x_i) + f'_y(x_i, y_i)(y - y_i) + \ldots, \tag{D.5}$$

где f'_x, f'_y — частные производные по соответствующим переменным.

Погрешность вычисления интеграла Δ_i равна

$$\Delta_i = \iint f(x, y)\,dxdy - f(x_i, y_i)\,\Delta x \Delta y. \tag{D.6}$$

Подставив (D.5) в (D.6), выполнив интегрирование и приведя подобные, получаем, что член пропорциональный $f(x_i, y_j)$ сокращается, а интеграл от $(x - x_i)dx$ дает $(\Delta x)^2/2$. Интеграл от данного выражения по dy дает еще один множитель Δy. Аналогичный вклад дает интеграл от члена, пропорционального $(y - y_i)$. Так как порядок погрешности Δy также составляет $O((\Delta x)^3)$, то погрешность интегрирования по прямоугольнику $x_i \leqslant x \leqslant x_{i+1}$, $y_i \leqslant y \leqslant y_{i+1}$ равна

$$\Delta_i \approx \frac{1}{2}\left[f'_x(x_i, y_i) + f'_y(x_i, y_i)\right](\Delta x)^3. \tag{D.7}$$

Из (D.7) видно, погрешность интегрирования по одному параллелепипеду составляет $O((\Delta x)^3)$. Так, имеется N параллелепипедов, полная погрешность по порядку величины равна $N(\Delta x)^3$. Однако в двумерном случае $N \sim 1/(\Delta x)^2$, поэтому полная погрешность $\Delta_i \sim (\Delta x) \sim O(N^{-1/2})$. Напомним, что в одномерном случае полная погрешность метода прямоугольников $\Delta_i \sim O(N^{-1})$.

Соответствующие оценки для двумерных обобщений формулы трапеций и Симпсона показывают, что они соответственно равны $O(N^{-1})$ и $O(N^{-2})$. Вообще можно показать, что если для одномерного случая погрешность составляет $O(n^{-\alpha})$, то в d-мерном случае она равна $O(n^{-\alpha/d})$. Отметим, что в отличие от данных методов погрешность метода Монте-Карло не зависит от размерности и меняется как $O(n^{-1/2})$. Следовательно, для достаточно больших d интегрирование по методу Монте-Карло будет приводить к меньшим погрешностям при тех же значениях N.

Сергей Владимирович ПОРШНЕВ

КОМПЬЮТЕРНОЕ МОДЕЛИРОВАНИЕ ФИЗИЧЕСКИХ ПРОЦЕССОВ В ПАКЕТЕ MATLAB

Учебное пособие

Издание второе, исправленное

Зав. редакцией
физико-математической литературы *А. П. Погода*
Художественный редактор *С. Ю. Малахов*
Верстка LaTeX *О. В. Вильчевская*
Подготовка иллюстраций *М. О. Мотыгина*
Выпускающие *Г. М. Матвеева, Т. В. Ананченко*

ЛР № 065466 от 21.10.97
Гигиенический сертификат 78.01.07.953.П.007216.04.10
от 21.04.2010 г., выдан ЦГСЭН в СПб

Издательство «ЛАНЬ». lan@lanbook.ru; www.lanbook.com
192029, Санкт-Петербург, Общественный пер., 5. Тел./факс: (812)412-29-35, 412-05-97,
412-92-72. Бесплатный звонок по России: 8-800-700-40-71

Подписано в печать 20.06.10. Бумага офсетная.
Гарнитура Литературная. Формат 70×100 $^{1}/_{16}$.
Печать офсетная. Усл. п. л. 59,80. Тираж 1000 экз.

Заказ № .

Отпечатано в полном соответствии с качеством
предоставленных материалов в ОАО «Дом печати — ВЯТКА»
610033, г. Киров, ул. Московская, 122