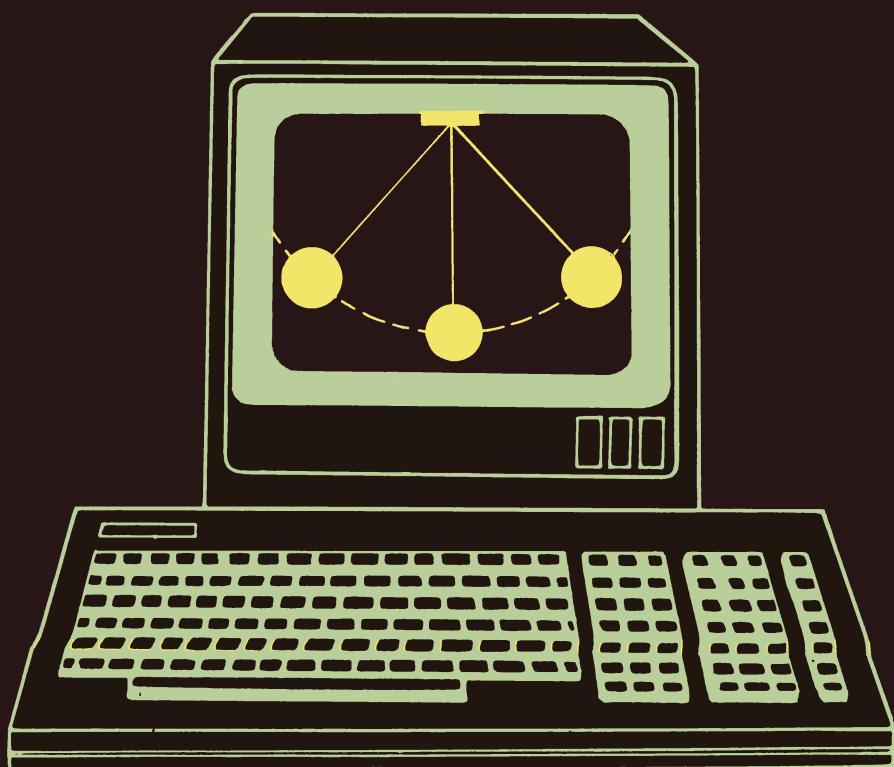


Х. Гулд, Я. Тобочник

Компьютерное моделирование в физике

2



Издательство «Мир»

КОМПЬЮТЕРНОЕ МОДЕЛИРОВАНИЕ В ФИЗИКЕ

Часть 2

HARVEY GOULD
Department of Physics, Clark University

JAN TOBOCHNIK
Department of Physics, Kalamazoo College

An Introduction to
Computer Simulation Methods
Applications to Physical Systems

Part 2

Addison-Wesley Publishing Company
Reading, Massachusetts · Menlo Park, California · New York
Don Mills, Ontario · Wokingham, England · Amsterdam · Bonn · Sydney
Singapore · Tokyo · Madrid · San Juan

Х. Гулд, Я. Тобочник

Компьютерное моделирование в физике

Часть вторая

Перевод с английского
канд. физ.-мат. наук А. Н. Полюдова
и В. А. Панченко



Москва "Мир" 1990

ББК 32.97:53

Г94

УДК53.072

Гулд Х., Тобочник Я.

Г85 Компьютерное моделирование в физике: В 2-х частях. Часть 2: Пер. с англ. — М.: Мир, 1990. — 400 с., ил.

ISBN 5-03-001594-9

Книга авторов из США предназначена для обучения читателя моделированию физических экспериментов на компьютере (и тем самым обучению физике). Все главы второй части связаны со статистической физикой и квантовой механикой. Каждая глава содержит теоретический материал, методы решения соответствующих задач, тексты программ, задачи и контрольные вопросы. В основном изложении используется True Basic, в приложении программы приведены на Паскале и Фортране-77; здесь же дан справочный материал, облегчающий перенос программ на различные модели компьютеров. Может служить учебным пособием.

Для студентов физических и технических вузов, аспирантов, преподавателей физики, молодых специалистов.

Г 1604010000—445 36—90
041(01)—90

ББК 32.97:53

Редакция литературы по физике и астрономии

ISBN 5-03-001592-2 (русск.)

ISBN 5-03-001594-9

ISBN 1-201-16504-2 (англ.)

© Addison-Wesley Publishing
Company 1988

© перевод на русский язык,
Панченко В.А., Полюдов А.Н.,
1990

ЧИСЛЕННОЕ ИНТЕГРИРОВАНИЕ

10

В этой главе иллюстрируется применение методов Монте-Карло и простых классических методов для численной оценки определенных интегралов.

10.1. ПРОСТЫЕ ОДНОМЕРНЫЕ МЕТОДЫ ЧИСЛЕННОГО ИНТЕГРИРОВАНИЯ

Основной целью этой главы является ознакомление с методами Монте-Карло на примере задачи вычисления определенных интегралов. Однако, чтобы в будущем получить представление о применении метода Монте-Карло для интегрирования, полезно сначала обсудить широко распространенные «классические» методы *численного интегрирования*. Мы увидим, что, хотя эти методы обычно предпочтительны в случае малых размерностей, они практически не годятся для многомерных интегралов и для вычисления последних наиболее пригодны методы Монте-Карло.

Рассмотрим одномерный определенный интеграл вида

$$F = \int_a^b dx f(x). \quad (10.1)$$

Для некоторых подынтегральных функций $f(x)$ интеграл в (10.1) можно вычислить аналитически, найти в справочниках или оценить с помощью асимптотических рядов. Однако, большинство общеизвестных функций проинтегрировать таким способом не удастся и интегралы от них нужно вычислять численно.

Классические методы численного интегрирования основаны на геометрической интерпретации интеграла (10.1) как площади под графиком функции $f(x)$ в пределах от $x = a$ до $x = b$ (рис. 10.1).

Ось x делится на n равных отрезков длиной Δx , где Δx равно

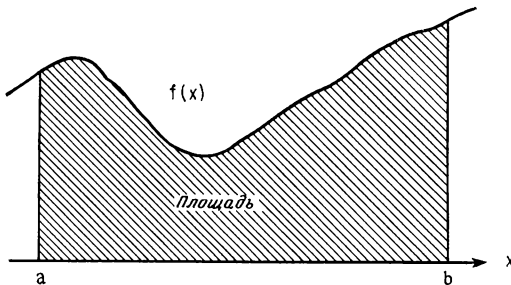


Рис. 10.1. Интеграл F равен площади под кривой $f(x)$.

$$\Delta x = \frac{b - a}{n} \quad (10.2a)$$

и

$$x_n = x_0 + n\Delta x. \quad (10.2b)$$

Для приведенного выше случая $x_0 = a$, а $x_n = b$.

Простейшей оценкой площади под кривой $f(x)$ служит сумма площадей прямоугольников, как показано на рис. 10.2. В обычном методе *прямоугольников* значение $f(x)$ вычисляется в *начале* каждого отрезка и оценка F_n интеграла дается выражением

$$F_n = \sum_{i=0}^{n-1} f(x_i)\Delta x. \quad (10.3)$$

Другим приближением является формула *трапеций*, в которой интеграл оценивается вычислением площади трапеции со сторонами, равными значениям $f(x)$ в начале и конце отрезка. Это приближение эквивалентно замене функции отрезком прямой, соединяющей значения $f(x)$ в начальной и конечной точках отрезка. Поскольку площадь под кривой от точки x_i до x_{i+1} равна $\frac{1}{2}[f(x_{i+1}) + f(x_i)]\Delta x$, то полная площадь F_n определяется выражением

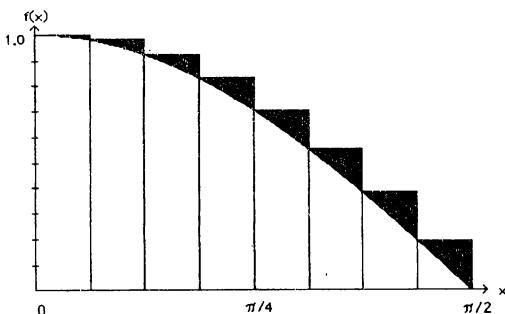


Рис. 10.2. Метод прямоугольников для функции $f(x) = \cos x$ на отрезке $0 \leq x \leq \pi/2$. Погрешность приближения показана закрашенными фигурами. Численное значение оценки интеграла в случае $n = 8$ приведено в табл. 10.1.

$$F_n = \left[\frac{1}{2} f(x_0) + \sum_{i=1}^{n-1} f(x_i) + \frac{1}{2} f(x_n) \right] \Delta x. \quad (10.4)$$

Обычно более высокую точность обеспечивает использование квадратичной, или параболической, интерполяции по трем соседним точкам. Например, уравнение полинома второй степени, проходящего через точки (x_0, y_0) , (x_1, y_1) и (x_2, y_2) , можно записать в виде

$$y(x) = y_0 \frac{(x - x_1)(x - x_2)}{(x_0 - x_1)(x_0 - x_2)} + y_1 \frac{(x - x_0)(x - x_2)}{(x_1 - x_0)(x_1 - x_2)} + y_2 \frac{(x - x_0)(x - x_1)}{(x_2 - x_0)(x_2 - x_1)}. \quad (10.5)$$

Чему равно значение $y(x)$ при $x = x_1$? Площадь под параболой $y(x)$ между точками x_0 и x_2 может быть найдена посредством простого интегрирования и выражается формулой

$$F_0 = \frac{1}{3} (y_0 + 4y_1 + y_2) \Delta x, \quad (10.6)$$

где $\Delta x = x_1 - x_0 = x_2 - x_1$. Полная площадь всех параболических сегментов выражается *формулой Симпсона*:

$$F_n = \frac{1}{3} [f(x_0) + 4f(x_1) + 2f(x_2) + 4f(x_3) + \dots + 2f(x_{n-2}) + 4f(x_{n-1}) + f(x_n)] \Delta x. \quad (10.7)$$

Обратите внимание на то, что в формуле Симпсона n должно быть четным числом.

10.2. ЧИСЛОВОЙ ПРИМЕР

Практически формула Симпсона пригодна для достаточно регулярных функций $f(x)$, т.е. для функций, которые можно аппроксимировать полиномом. Если $f(x)$ является такой «гладкой» функцией, можно вычислить площадь для заданного числа отрезков n , а затем удвоить число отрезков и вновь вычислить эту площадь. Если обе оценки достаточно близки, то вычисления прекращаются. В противном случае можно продолжать удваивать n , до тех пор пока не будет достигнута требуемая

точность. Понятно, что данная процедура не будет работать, если функция $f(x)$ не является гладкой.

Можно ли понять заранее, какой из методов, трапеций или Симпсона, окажется лучше? Один способ основан на предположении, что функцию $f(x)$ можно представить полиномом; в этом случае погрешность можно оценить (см. приложение 10А). Однако подчеркнем, что величина погрешности сильно зависит от характера самой функции $f(x)$ и ее поведения на концах отрезка интегрирования и, следовательно, никакой численный метод не может быть универсальным.

Приведем пример программы вычисления интеграла от функции $f(x)$ методом прямоугольников.

```
PROGRAM integ
! вычисляется интеграл от функции f(x) в пределах от x = a до x = b
CALL initial(a,b,h,n)
CALL rectangle(a,b,h,n,area)
CALL output(area)
END

SUB initial(a,b,h,n)
  LET a = 0                                ! нижний предел
  LET b = 0.5*pi                          ! верхний предел
  INPUT prompt "число интервалов = ": n
  LET h = (b - a)/n                        ! шаг интегрирования
END SUB

SUB rectangle(a,b,h,n,area)
  DECLARE DEF f
  LET x = a
  FOR i = 0 to n - 1
    LET sum = sum + f(x)
    LET x = x + h
  NEXT i
  LET area = sum*h
END SUB

SUB output(area)
  PRINT using "****.*****": area
END SUB

DEF f(x) = cos(x)
```

Реализация формулы Симпсона, иллюстрирующая ее связь с формулой трапеций, приводится в нижеследующей подпрограмме, которую нужно подставить в программу `integ` вместо подпрограммы `rectangle`.

```

SUB Simpson (a,b,h,n,area)
! приближение Симпсона
DECLARE DEF f
LET sumall = f(a) + f(b)
LET sumall = 0.5*sumall           ! вклад концевых точек
LET sumeven = sumall             ! вклад только четных членов
LET x = a
FOR i = 1 to n - 1
  LET x = x + h
  LET integrand = f(x)
  LET sumall = sumall + integrand ! четные и нечетные члены
  LET parity = mod(i,2)          ! 0 четное число, 1 нечетное число
  IF parity = 0 then LET sumeven = sumeven + integrand
NEXT i
LET sum = 4*sumall - 2*sumeven
LET area = h*sum/3               ! численная оценка
END SUB

```

Вместо того чтобы на оценивать погрешности различных классических формул, о которых шла речь в разд. 10.1, рассмотрим точность метода прямоугольников для интеграла от функции $f(x) = \cos x$ в пределах от $x = 0$ до $x = \pi/2$. В табл. 10.1 приведены результаты расчетов по программе `integ` в порядке возрастания n . Как видно из анализа зависимости от n разности между численной оценкой F_n и точным результатом, равным единице, погрешность убывает как $1/n$. Эта наблюдаемая зависимость погрешности от n согласуется с общим результатом, полученным в приложении 10А.

ЗАДАЧА 10.1. Прямоугольное приближение

а. Модифицируйте программу `integ` так, чтобы можно было наглядно увидеть площадь под кривой $y = f(x)$ и площади прямоугольников (см. рис. 10.2).

б. Используя метод прямоугольников, оцените численно определенные интегралы от функций $f(x) = 2x + 3x^2 + 4x^3$ и $f(x) = e^{-x}$ на

ТАБЛИЦА 10.1. Оценки методом прямоугольников интеграла от функции $\cos x$ на отрезке от $x = 0$ до $x = \pi/2$ в зависимости от числа отрезков n . Погрешность Δ_n равна разности между приближенным значением F_n и точным результатом, равным единице. Анализ зависимости Δ_n от n показывает, что Δ_n убывает примерно как $1/n$

n	F_n	Δ_n
2	1.34076	0.34076
4	1.18346	0.18346
8	1.09496	0.09496
16	1.04828	0.04828
32	1.02434	0.02434
64	1.01222	0.01222
128	1.00612	0.00612
256	1.00306	0.00306
512	1.00153	0.00153
1024	1.00077	0.00077

отрезке $0 \leq x \leq 1$. Как приблизительно зависит погрешность от n в каждом случае?

ЗАДАЧА 10.2. Метод средней точки

а. Одна из распространенных модификаций метода прямоугольников заключается в вычислении $f(x)$ в *средней* точке каждого отрезка. Внесите необходимые изменения в программу `integ` и вычислите интеграл от $\cos x$. Как соотносится величина погрешности с результатами, приведенными в табл. 10.1? Как приблизительно зависит погрешность от n ?

б. Используя метод средней точки, оцените определенные интегралы от функций $f(x) = 2x + 3x^2 + 4x^3$ и $f(x) = e^{-x}$ на отрезке $0 \leq x \leq 1$. Как приблизительно зависит погрешность от n в каждом случае?

ЗАДАЧА 10.3. Формулы трапеций и Симпсона

а. Каким образом можно модифицировать подпрограмму `Simpson` для одновременного получения оценок интеграла от функции $f(x)$ по формулам трапеций и Симпсона?

б. Используя оба метода, оцените численно интегралы от функций $f(x) = 2x + 3x^2 + 4x^3$ и $f(x) = e^{-x}$ на отрезке $0 \leq x \leq 1$. Как приблизительно зависит погрешность от n в каждом случае? Какой метод приводит к лучшим результатам при одинаковом времени вычислений?

в. Используя формулу Симпсона, оцените значение интеграла от функции $f(x) = (2\pi)^{-1/2} e^{-x^2}$ на отрезках $-1 \leq x \leq 1$, $-2 \leq x \leq 2$ и $-3 \leq x \leq 3$.

г. Используя оба метода, оцените определенный интеграл от функции $f(x) = (1 + x^2)^{-1}$ на отрезках $0 \leq x \leq 1$ и $0 \leq x \leq 2$. Какой метод приводит к лучшим результатам? Помните о том, что метод более высокого порядка не всегда дает более высокую точность.

д. Нами уже описана процедура оценки одномерных определенных интегралов, т.е. для выбранной классической формулы интегрирования вычисляются F_n и F_{2n} для приемлемого значения n . Если разность $|F_{2n} - F_n|$ слишком велика, то удваиваем n до тех пор, пока не будет достигнута требуемая точность. Сходится ли последовательность F_n, F_{2n}, \dots к истинному значению интеграла F , и если да, то существует ли какой-нибудь способ экстраполяции к пределу? Рассмотрим эту идею на примере формулы трапеций. Поскольку мы нашли, что погрешность этого приближения убывает приблизительно как n^{-2} , то мы вправе записать $F = F_n + Cn^{-2}$. Постройте график оценки F_n в зависимости от n^{-2} и получите экстраполированное значение F . Какие предположения необходимо сделать для того, чтобы процедура экстраполяции была успешной? Примените описанный метод к интегралам, рассмотренным в предыдущих задачах, и сравните полученные результаты с тем, что дают один метод трапеций и метод Симпсона. Более изящно описанная идея воплощена в методе Ромберга (см. книгу Пресс и др.).

10.3. ЧИСЛЕННОЕ ИНТЕГРИРОВАНИЕ МНОГОМЕРНЫХ ИНТЕГРАЛОВ

Многие физические задачи содержат усреднение по многим переменным. Например, предположим, что нам известна зависимость от скорости и координаты полной энергии системы десяти взаимодействующих между собой частиц. Поскольку в трехмерном пространстве каждая частица

имеет по три компоненты скорости и координаты, то полная энергия является функцией 60 переменных. Следовательно, для расчета средней энергии, приходящейся на частицу, требуется вычислять $N = 60$ -мерный интеграл. Если разделить область изменения каждой координаты на p отрезков, то в данном случае потребуется вычислять сумму по p^{60} точкам. Совершенно очевидно, что для больших значений N применять обычные численные методы нельзя; тем не менее стандартные методы все еще используются для $N = 2-5$.

Дополнительное усложнение в вычисление N -мерных интегралов вносит трудность определения $N-1$ пределов интегрирования. По сравнению с этим границы одномерного интеграла представляются двумя числами: верхним и нижним пределами.

Простейший метод оценки многомерных интегралов заключается в сведении этих интегралов к произведению одномерных интегралов. Данный метод эффективен в случае простых пределов интегрирования и гладких подынтегральных функций. Проиллюстрируем метод на примере двумерного интеграла вида

$$F = \int_{x_1}^{x_2} \int_{y_1(x)}^{y_2(x)} dx dy f(x, y). \quad (10.8)$$

Область интегрирования определяется нижним и верхним пределами y при данном значении x , обозначенными соответственно $y_1(x)$ и $y_2(x)$, и нижним и верхним пределами x , обозначенными соответственно x_1 и x_2 . Определим функцию $g(x)$ как внутренний интеграл по переменной y :

$$g(x) = \int_{y_1(x)}^{y_2(x)} dy f(x, y) \quad (10.9)$$

и запишем

$$F = \int_{x_1}^{x_2} dx g(x). \quad (10.10)$$

Ниже приводится структура программы двумерного интегрирования. Заметим, что для простоты использован метод прямоугольников.

```

PROGRAM integ2          ! вычисление двумерного интеграла от f(x,y)
CALL initial(x1,x2,h,n)
CALL integx(x1,x2,h,n,area)
CALL output(area)
END

SUB initial (x1,x2,h,n)
  LET x1 = -1              ! нижний предел x
  LET x2 = 1              ! верхний предел x
  INPUT prompt "число интервалов = ": n
  LET h = (x2 - x1)/n
END SUB

SUB integx (x1,x2,h,n,area)
  LET x = x1
  FOR i = 0 to n - 1
    CALL integy(x,n,g)
    LET sum = sum + g
    LET x = x + h
  NEXT i
  LET area = sum*h
END SUB

SUB integy (x,n,g)
  DECLARE DEF y1
  DECLARE DEF y2
  DECLARE DEF f
  LET h = (y2(x) - y1(x))/n
  LET y = y1(x)
  LET g = 0
  FOR i = 0 to n - 1
    LET g = g + f(x,y)
    LET y = y + h
  NEXT i
  LET g = g*h              ! оценка интеграла по y для данного x
END SUB

SUB output (area)
  PRINT using "####.#####": area
END SUB

```

```
DEF γ1(x) = -sqrt(1 - x*x)
```

```
DEF γ2(x) = sqrt(1 - x*x)
```

```
DEF f(x, γ) = x*x + 6*x*γ + γ*γ
```

ЗАДАЧА 10.4. Двумерное численное интегрирование

а. Проинтегрируйте функцию $f(x, y) = x^2 + 6xy + y^2$ по области, определяемой условием $x^2 + y^2 \leq 1$. Примените метод прямоугольников, как и в программе `integ2`, и выберите число интервалов по каждой переменной равным $n = 2^p$. Используйте значения p в диапазоне $p = 2-7$.

б. Модифицируйте программу `integ2` так, чтобы интегрирование по каждой переменной выполнялось с помощью формулы Симпсона, и повторите расчет, проведенный в п. «а».

в. Оцените зависимость погрешности от n в п.п. «а» и «б», используя различные значения p .

Вопрос о зависимости от n погрешности численной аппроксимации d -мерного интеграла рассмотрен в приложении 10А. Там показано, что если для $d = 1$ погрешность аппроксимации убывает как n^{-a} , то в d -мерном случае эта погрешность убывает как $n^{-a/d}$.

10.4. ВЫЧИСЛЕНИЕ ИНТЕГРАЛОВ ПРОСТЕЙШИМ МЕТОДОМ МОНТЕ-КАРЛО

Каким образом можно с помощью кучи камней измерить площадь пруда? Предположим, что пруд расположен в центре поля известной площади A . Бросайте камни произвольным образом так, чтобы они падали в случайных точках в пределах поля, и считайте количество всплесков при попадании камней в пруд. Площадь пруда приблизительно равна площади поля, умноженной на долю камней, попавших в пруд. Эта простая процедура является примером метода *Монте-Карло*.

Поясним подробнее суть этого метода. Представим себе прямоугольник высотой H и длиной $b - a$ такой, что функция $f(x)$ лежит внутри

него (рис. 10.3). Генерируем n пар случайных чисел x_i и y_i , удовлетворяющих условиям $a \leq x_i \leq b$ и $0 \leq y_i \leq H$. Доля точек (x_i, y_i) , которые удовлетворяют условию $y_i \leq f(x_i)$, представляет собой оценку отношения интеграла от функции $f(x)$ к площади прямоугольника. Отсюда оценка F_n в методе «проб и ошибок» определяется выражением

$$F_n = A \frac{n_s}{n}, \quad (10.11)$$

где n_s — число «всплесков» или точек, лежащих под кривой, n — общее количество точек, а A — площадь прямоугольника. Заметим, что не следует путать величину n в формуле (10.11) с числом интервалов n , которое фигурировало в рассмотренных выше численных методах.

Другая разновидность метода Монте-Карло основывается на теореме анализа, гласящей, что интеграл (10.1) определяется средним значением подынтегральной функции $f(x)$ на отрезке $a \leq x \leq b$. Для вычисления этого среднего возьмем x_i не с постоянным шагом, а случайным образом и произведем *выборку* значений $f(x)$. Оценка F_n одномерного интеграла (10.1) методом «выборочного среднего» выражается формулой

$$F_n = (b - a) \langle f \rangle = (b - a) \frac{1}{n} \sum_{i=1}^n f(x_i), \quad (10.12)$$

где x_i — случайные числа, равномерно распределенные на отрезке $a \leq x_i \leq b$, а n — количество *испытаний*. Обратите внимание на то, что формулы (10.3) и (10.12) различаются только тем, что в первой формуле n точек выбираются с постоянным шагом, а во второй — случайным образом. Мы обнаружим, что для интегралов невысокой размерности

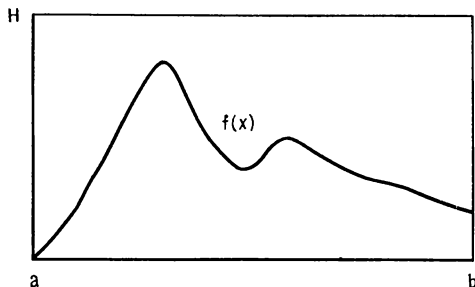


Рис. 10.3. Функция $f(x)$ находится в прямоугольной области высотой H и длиной $(b - a)$.

формула (10.3) оказывается точнее, но для многомерных интегралов все же предпочтительнее формула (10.12).

Может показаться странным, что с помощью детерминированного компьютера удастся генерировать последовательности «случайных» чисел. Пока нас вполне устроит использование «псевдослучайных» последовательностей, генераторы которых предусмотрены в различных языках программирования. В приведенной ниже программе иллюстрируется употребление функций `rnd` и `randomize`, имеющихся в языке True BASIC, для получения случайных чисел, равномерно распределенных на отрезке $[0, 1]$. Свойства этих функций будут рассмотрены в гл. 11.

```
PROGRAM random          ! при каждом обращении генерируется новая
RANDOMIZE                ! случайная последовательность
LET nrandom = 10        ! количество случайных чисел
CALL random1(nrandom)
END

SUB random1(nrandom)
  FOR i = 1 to nrandom
    LET x = rnd
    PRINT x
  NEXT i
END SUB
```

Если мы хотим получить случайные числа, равномерно распределенные на отрезке $[a, b]$, то используем соотношение $x = a + (b - a)r$, где r — равномерно распределенные числа на отрезке $[0, 1]$. Это соотношение реализовано в следующей подпрограмме.

```
SUB random2(a,b,nrandom)
  FOR i = 1 to nrandom
    LET x = a + (b - a)*rnd
    PRINT x
  NEXT i
END SUB
```

Следующая подпрограмма генерирует случайные целые числа на отрезке $[1, N]$.

```

SUB random_integers(nrandom)
  LET N = 10
  FOR i = 1 to nrandom
    LET x = int(N*rnd + 1)
    PRINT x
  NEXT i
END SUB

```

ЗАДАЧА 10.5. Вычисление одномерных интегралов методом Монте-Карло

а. Составьте программу, которая реализует метод проб и ошибок Монте-Карло, выражаемый итоговой формулой (10.11). Найдите оценку F_n интеграла от функции $f(x) = 4\sqrt{1-x^2}$ в зависимости от числа испытаний n . Положите $a = 0$, $b = 1$, $H = 1$ и произведите выборку функции $\sqrt{1-x^2}$. Умножьте оценку на 4. Вычислите разность между F_n и точным результатом, равным π . Эта разность является мерой погрешности метода Монте-Карло. Постройте график зависимости погрешности от n в дваждылогарифмическом масштабе. Какой будет приближительная функциональная зависимость погрешности от n при больших значениях n ?

б. Оцените интеграл от $f(x)$, используя метод Монте-Карло выборочного среднего (10.12), и вычислите погрешность как функцию от числа испытаний n для значений n до 10 000. Определите приближительную функциональную зависимость погрешности от n для больших значений n . Сколько необходимо провести испытаний, чтобы получить F_n с точностью до двух десятичных знаков?

в. Определите машинное время, затрачиваемое на одно испытание, для этих двух разновидностей метода Монте-Карло. Какой из методов предпочтительней в этом случае?

10.5. ВЫЧИСЛЕНИЕ МНОГОМЕРНЫХ ИНТЕГРАЛОВ МЕТОДОМ МОНТЕ-КАРЛО

Численное интегрирование методом Монте-Карло наиболее эффективно для оценки многомерных интегралов, которые, как правило, не вычисляются обычными численными методами. В качестве примера расчета многомерных интегралов этим методом рассмотрим нахождение центра

масс и момента инерции твердых тел. Предположим, что масса распределена непрерывно с известной плотностью $\rho(x, y)$. Для простоты будем рассматривать двумерные тела. Масса малого элемента площади $dxdy$ равна

$$dm = \rho(x, y) dxdy, \quad (10.13)$$

а полная масса тела равна

$$M = \iint \rho(x, y) dxdy. \quad (10.14)$$

Пределы интегрирования определяются геометрией тела. Аналогично можно выразить координаты центра масс X, Y :

$$X = \frac{1}{M} \iint x \rho(x, y) dx dy \quad (10.15a)$$

и

$$Y = \frac{1}{M} \iint y \rho(x, y) dx dy. \quad (10.15b)$$

Если тело вращается вокруг оси z , то его момент инерции равен

$$I_z = \iint (x^2 + y^2) \rho(x, y) dxdy. \quad (10.16)$$

Вычисление M, R и I методом выборочного среднего осуществляется просто. Например, обобщение метода выборочного среднего для вычисления момента инерции имеет вид

$$I_{z,n} = (x_2 - x_1)(y_2 - y_1) \frac{1}{n} \sum_{i=1}^n (x_i^2 + y_i^2) \rho(x_i, y_i), \quad (10.17)$$

где $I_{z,n}$ — оценка I_z для n испытаний, а x_i и y_i — независимые случайные числа на отрезках $x_1 \leq x \leq x_2$ и $y_1 \leq y \leq y_2$.

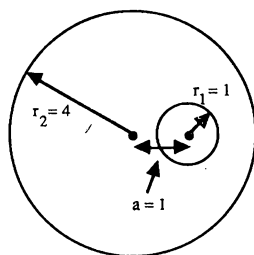


Рис. 10.4. Форма твердого диска, рассматриваемого в задаче 10.6.

ЗАДАЧА 10.6. Свойства твердого диска

а. Вычислите методами Монте-Карло полную массу, центр масс и момент инерции диска, изображенного на рис. 10.4. Внешний радиус диска $r_2 = 4$, а радиус круговой полости $r_1 = 1$. Центр полости находится на расстоянии $a = 1$ от центра большего диска. Плотность остальной части этого тела однородна и равна $\rho = 1$.

б. Повторите расчеты п. «а» с плотностью $\rho(x, y) = 1 + \frac{1}{2}(x^2 + y^2)$, где x и y измеряются относительно центра большего диска.

*в. Вычислите рассмотренные выше величины, используя формулу Симпсона. Какой метод более удобен в этом случае? Какой метод эффективнее с точки зрения использования машинного времени?

10.6. АНАЛИЗ ПОГРЕШНОСТИ МЕТОДА МОНТЕ-КАРЛО

Как метод Монте-Карло, так и классические методы численного интегрирования дают приближенные результаты. Точность определяется числом испытаний в методе Монте-Карло и количеством отрезков разбиения в классических методах. В задаче 10.5 мы воспользовались знанием точных значений интегралов и установили, что погрешность метода Монте-Карло стремится к нулю при больших n приблизительно как $n^{-1/2}$. В дальнейшем мы оценим погрешность для случая, когда точный ответ неизвестен. Наш основной результат состоит в том, что зависимость погрешности от n не чувствительна к виду подынтегральной функции и, что наиболее ценно, не зависит от кратности интеграла. Поскольку в

задаче 10.3 мы нашли, что погрешность метода Симпсона для одномерного интеграла пропорциональна n^{-4} , то делаем вывод, что для интегралов небольшой размерности стандартные численные методы, такие, как формула Симпсона, обычно предпочтительнее методов Монте-Карло, если область интегрирования не очень сложна. Однако поскольку с ростом размерности погрешность стандартных численных методов увеличивается (см. приложение 10А), то для интегралов большой размерности главными являются методы Монте-Карло.

Так как нахождение надлежащей меры погрешности в расчетах методом Монте-Карло является тонким делом, получим эту погрешность на конкретном примере. Вычислим методом Монте-Карло интеграл от функции $f(x) = 4\sqrt{1-x^2}$ на отрезке $[0, 1]$. Результат наших вычислений, полученный методом выборочного среднего для конкретной последовательности $n = 10\,000$ случайных чисел, равен $F_n = 3.1489$. Сравнивая эту оценку с точным ответом $F = \pi \approx 3.1416$, находим, что соответствующая погрешность для случая $n = 10\,000$ испытаний приблизительно составляет 0.0073. Как соотносится наше значение F_n с результатом, полученным вами в задаче 10.5 с тем же значением n ? Мы знаем, что если подынтегральная функция не равна константе, то F_n не будет, вообще говоря, равно F . Как можно узнать, достаточно ли $n = 10\,000$ испытаний для достижения требуемой точности? Сколь малым можно ожидать отличие F_n от F для данного значения n ? Конечно, мы не можем ответить на эти вопросы со всей определенностью, поскольку если бы фактическая погрешность F_n была известна, то можно было бы внести в F_n соответствующую поправку и получить F . Самое лучшее, что мы можем сделать, — это вычислить *вероятность* того, что истинное значение F лежит в некотором интервале с центром в F_n .

Одной возможной мерой погрешности, которая, по-видимому, хорошо вам знакома, является *дисперсия* σ^2 , определяемая выражением

$$\sigma^2 = \langle f^2 \rangle - \langle f \rangle^2, \quad (10.18)$$

где

$$\langle f \rangle = \frac{1}{n} \sum_{i=1}^n f(x_i) \quad (10.19a)$$

и

$$\langle f^2 \rangle = \frac{1}{n} \sum_{i=1}^n f(x_i)^2. \quad (10.19b)$$

Величина σ называется *стандартным отклонением*. Нам известно, что если бы f не зависела от x , то σ равнялось бы нулю. Для нашего примера и той же последовательности случайных чисел, которая использовалась для вычисления F_n , получаем $\sigma = 0.8850$. Поскольку фактическая погрешность, как мы знаем, гораздо меньше σ , то напрашивается вывод, что σ не может быть непосредственной мерой погрешности.

Одним из условий, которому должна удовлетворять мера погрешности, является ее убывание с ростом n . Как вы считаете, будет ли σ , определенная выражением (10.18), убывать с ростом n ? Один из способов получения оценки погрешности заключается в проведении серии дополнительных расчетов по n испытаний в каждом. Каждый такой расчет дает среднее значение, или *измерение*, которое мы обозначим M_α . Эти измерения, вообще говоря, не будут одинаковыми, поскольку каждое измерение производится со своей последовательностью случайных чисел. Качественно величина разброса измерений служит мерой погрешности одного измерения. Предположим, имеется набор из m измерений $\{M_\alpha\}$, состоящих из одинакового числа испытаний. Удобной мерой разброса этих измерений является *стандартное отклонение средних* σ_m , которое мы определим как

$$\sigma^2 = \langle M^2 \rangle - \langle M \rangle^2, \quad (10.20)$$

где

$$\langle M \rangle = \frac{1}{m} \sum_{\alpha=1}^m M_\alpha \quad (10.21a)$$

и

$$\langle M^2 \rangle = \frac{1}{m} \sum_{\alpha=1}^m M_\alpha^2. \quad (10.21b)$$

Чтобы лучше уяснить смысл σ_m , найдем *оценку* σ_m , проведя $m = 10$ измерений. В результате каждого измерения, состоящего из $n = 10\,000$ испытаний, получают свои значения среднего M и σ . Найденные значения M и σ для каждого измерения приведены в табл. 10.2.

Из табл. 10.2 видно, что в разных сериях измерения M отличаются. Исходя из значений M в табл. 10.2 и соотношения (10.20), получаем оценку для σ_m :

ТАБЛИЦА 10.2. Вычисления среднего значения функции $f(x) = 4\sqrt{1-x^2}$ на отрезке $[0, 1]$ методом Монте-Карло. Всего проведено 10 измерений (серий) по 10 000 испытаний в каждом. Для каждого измерения приведены среднее значение M и стандартное отклонение σ

Серия α	M	σ
1	3.14892	0.88501
2	3.13255	0.89865
3	3.14042	0.88924
4	3.14600	0.88525
5	3.15257	0.88757
6	3.13972	0.89698
7	3.13107	0.89700
8	3.13585	0.89406
9	3.13442	0.89746
10	3.14047	0.89213

$$\sigma_m = 0.0068. \quad (10.22)$$

Это значение σ_m согласуется с полученным выше результатом для погрешности, равной 0.0073. Отсюда можно заключить, что стандартное отклонение средних σ_m представляет собой меру погрешности одного измерения. Более точная интерпретация σ_m состоит в том, что с вероятностью 0.68 значение F_n (т.е. M_α) отличается от «истинного» среднего не более чем на σ_m . Следовательно, *вероятная ошибка* нашего первого измерения F_n с $n = 10\,000$ составляет 3.149 ± 0.007 .

Хотя σ_m дает нам оценку вероятной ошибки, наш метод получения σ_m , требующий проведения дополнительных измерений, не используется. В приложении 10Б мы приводим аналитический вывод соотношения

$$\sigma_m \approx \sigma/\sqrt{n}. \quad (10.23)$$

Данное соотношение между σ_m и σ в пределе очень большого числа измерений превращается в точное равенство. Заметим, что из (10.23) следует, что вероятная ошибка убывает как корень квадратный из числа испытаний. В случае нашего примера находим, что вероятная ошибка первоначального измерения приблизительно равна $0.8850/100 \approx 0.009$, что согласуется с известной ошибкой 0.007 и с нашей оценкой $\sigma_m \approx 0.007$.

Один из способов убедиться в правильности соотношения (10.23) основан не на проведении дополнительных измерений, а на разбиении

ТАБЛИЦА 10.3. Средние значения f_s функции

$f(x) = 4\sqrt{1-x^2}$ на отрезке $0 \leq x \leq 1$ для 10 групп по 1000 испытаний в каждой. Среднее значение $f(x)$ по всем 10 группам равно 3.14892, что совпадает со значением F_n первого измерения в табл. 10.2

Группа s	f_s
1	3.14326
2	3.15633
3	3.10940
4	3.15337
5	3.15352
6	3.11506
7	3.17989
8	3.12398
9	3.17565
10	3.17878

начального измерения из n испытаний на s групп. Обозначим среднее значение функции $f(x_s)$ в каждой из s групп через f_s и проанализируем отклонения F_s по каждой группе. Например, разобьем 10 000 испытаний первого измерения на $s = 10$ групп по $n/s = 1000$ испытаний в каждой. Значения f_s приведены в табл. 10.3.

Заметим, что, как и ожидалось, средние значения $f(x)$ для каждой группы различны. В качестве меры погрешности разумно взять стандартное отклонение средних каждой группы. Обозначим эту величину σ :

$$\sigma_s = \langle f_s^2 \rangle - \langle f_s \rangle^2, \quad (10.24)$$

где усреднение выполняется по всем группам. Из табл. 10.3 получаем $\sigma_s = 0.025$. Хотя σ_s можно рассматривать в качестве вероятной ошибки, мы видим, что эта величина почти в три раза превышает оценку 0.009, приведенную в (10.22). Кроме того, необходимо получить оценку ошибки, которая не зависит от разбиения данных на группы. Такой величиной является не σ_s , а отношение σ_s/\sqrt{s} , которое в нашем примере приблизительно равно $0.025/3.16 \approx 0.008$. Это значение согласуется с σ_m и отношением σ/\sqrt{n} . Значит, n испытаний можно интерпретировать либо как одно измерение, либо как набор s измерений. В первой интерпретации вероятная ошибка равна стандартному отклонению для n испытаний, поделенному на квадратный корень из числа испытаний. Аналогично вторая интерпретация означает, что вероятная ошибка равна стандартному отклонению s измерений f_s , поделенному на квадратный корень из числа измерений.

Заметим, что ошибку можно сделать сколь угодно малой, либо увеличивая число испытаний, либо повышая «эффективность» испытаний и посредством этого уменьшая стандартное отклонение σ . Некоторые методы уменьшения дисперсии рассматриваются в разд. 10.8–10.9.

ЗАДАЧА 10.7. Оценка ошибки метода Монте-Карло

а. Оцените интеграл от $f(x) = e^{-x}$ на отрезке $0 \leq x \leq 1$. Воспользуйтесь методом Монте-Карло выборочного среднего с $n = 3\,600$ испытаний. Вычислите стандартное отклонение σ , определяемое формулой (10.18). Значительно ли меняется ваша оценка с увеличением числа испытаний? Поскольку точный ответ можно получить аналитически, вычислите ошибку. Как соотносится эта ошибка с полученной по формуле (10.23)?

б. Произведите девятнадцать дополнительных измерений интеграла по $n = 3\,600$ испытаний каждое. Чему равно стандартное отклонение σ_m для двадцати измерений? Совпадает ли σ_m с оценкой ошибки, полученной в п. «а»? Сильно ли изменится ваша оценка σ_m , если увеличить количество измерений?

в. Разбейте свое первое измерение на $s = 20$ групп по 180 испытаний в каждой. Вычислите стандартное отклонение σ_s групп. Совпадает ли величина $\sigma_s/s^{1/2}$ с полученными ранее оценками ошибки?

г. Разбейте свое первое измерение на $s = 10$ групп по 360 измерений в каждой и вычислите стандартное отклонение групп. Чему равно значение $\sigma_s/s^{1/2}$ в каждом случае? Как соотносятся стандартные отклонения групп для этих двух различных разбиений данных?

д. Вычислите с точностью до двух десятичных знаков интеграл

$$\int_0^1 dx e^{-x^2}, \quad (10.25)$$

используя в качестве оценки вероятной ошибки величину σ_n/\sqrt{n} .

*ЗАДАЧА 10.8. Важность случайности

В гл. 11 мы узнаём, что генератор случайных чисел, включенный во

многие языки программирования, основан на методе «вычетов». В этом методе каждый последующий член последовательности может быть найден из предыдущего по формуле

$$x_{n+1} = ax_n + c \bmod m, \quad (10.26)$$

где x_0 — «начальное число», а a , c и m — неотрицательные целые числа. Случайные числа на единичном отрезке $[0, 1]$ получаются по формуле $r_n = x_n/m$. Запись $y = x \bmod m$ означает, что если x больше m , то *модуль* m вычитается из x до тех пор, пока не выполнится условие $0 \leq y \leq m$. В конце концов эта последовательность чисел начнет повторяться, определяя, таким образом, *период* генератора случайных чисел. Для изучения работы плохого генератора случайных чисел выберем параметры x_0 , a , c и m так, чтобы последовательность (10.26) обладала плохими статистическими свойствами, например коротким периодом. Чему равен этот период в случае $x_0 = 1$, $a = 5$, $c = 0$ и $m = 32$? Оцените интеграл из задачи 10.7а, проведя одно измерение из 3 600 испытаний с помощью генератора «случайных чисел» (10.26) с приведенными выше значениями x_0 , a , c и m . Проанализируйте результат измерения тем же способом, что и раньше, т.е. вычислите среднее, среднее для каждой из двадцати групп и стандартное отклонение этих средних. Затем разбейте данные на десять групп и вычислите те же величины. Связаны ли стандартные отклонения групп такой же зависимостью, как и ранее? Если нет, то почему?

10.7. НЕРАВНОМЕРНЫЕ РАСПРЕДЕЛЕНИЯ ВЕРОЯТНОСТЕЙ

В предыдущих двух разделах мы познакомились с тем, как можно использовать равномерно распределенные случайные числа для оценки определенных интегралов. Однако желательно, вообще говоря, выборку подынтегральной функции $f(x)$ производить чаще в тех областях x , где $|f(x)|$ велика или быстро меняется. Поскольку в таких методах «выборки по значимости» требуются неравномерные распределения вероятностей, рассмотрим здесь метод «обратного преобразования» (обратной функции) для генерирования случайных чисел, которые распределены неравномерно.

Необходимо ввести понятие *плотности вероятности* $p(x)$, при этом $p(x)dx$ — вероятность того, что случайное число принадлежит отрезку

$[x, x + dx]$. Плотность вероятности нормируется так, чтобы

$$\int_{-\infty}^{+\infty} dx p(x) = 1. \quad (10.27)$$

Примем, что r — случайное число, равномерно распределенное на единичном интервале $[0, 1]$ с плотностью вероятности

$$p_u(r) = \begin{cases} 1, & \text{если } 0 \leq r \leq 1, \\ 0 & \text{в противном случае.} \end{cases} \quad (10.28)$$

Переменная, удовлетворяющая распределению вероятности (10.28), называется *равномерно уклоняющейся*. Наша цель состоит в том, чтобы найти соотношение между x и r такое, что если r распределена по закону (10.28), то x будет распределена с плотностью вероятности $p(x)$. Для получения этого соотношения сначала вычислим интеграл

$$P(x) = \int_{-\infty}^x dx' p(x'). \quad (10.29)$$

Функция $P(x)$ интерпретируется как *интегральная функция распределения*, которая равна вероятности получения случайного числа, меньшего или равного x . Геометрическая интерпретация: $P(x)$ представляет собой часть площади под кривой плотности вероятности $p(x)$ слева от x . Требуемое соотношение между величинами x и r легче сформулировать, чем вывести. Ниже мы покажем, что обращение выражения

$$P(x) = r \quad (10.30)$$

приводит к искомому соотношению

$$x = P^{-1}(r). \quad (10.31)$$

Чтобы разобраться в формулах (10.30) и (10.31), заметим, что поскольку переменная r равномерно распределена на единичном отрезке, то функция $P(x)$, связанная с r соотношением (10.30), также распределена равномерно. Вероятность того, что $P(x)$ принадлежит отрезку

$[P(x), P(x) + dP(x)]$, равна $dP(x)$, и она же, согласно (10.30), равна dr . Соотношение между $dP(x)$ и dx можно найти, дифференцируя равенство (10.29):

$$\frac{dP(x)}{dx} = p(x). \quad (10.32)$$

Отсюда, используя (10.28), имеем для $0 \leq r \leq 1$

$$dP(x) = p(x)dx = p_u(r)dr. \quad (10.33)$$

Из выражения (10.33) видим, что переменная x распределена с требуемой плотностью вероятности $p(x)$.

Последовательность шагов, соответствующая методу обратного преобразования, следующая: генерируется случайное число r и решается (10.31) с получением соответствующего значения x . В качестве примера воспользуемся (10.31) для получения случайных чисел, равномерно распределенных на отрезке $[a, b]$. Требуемая плотность вероятности $p(x)$ равна

$$p(x) = \begin{cases} 1/(b - a), & \text{если } a \leq x \leq b, \\ 0 & \text{в противном случае.} \end{cases} \quad (10.34)$$

Функцию интегрального распределения вероятности $P(x)$ на отрезке $a \leq x \leq b$ можно найти, подставив (10.34) в (10.29) и вычислив интеграл. В результате получим

$$P(x) = \frac{x - a}{b - a}. \quad (10.35)$$

Если подставить выражение (10.35) для $P(x)$ в (10.30), то найдем искомое соотношение

$$x = P^{-1}(r) = a + (b - a)r. \quad (10.36)$$

Переменная x , определяемая формулой (10.36), распределена с плотностью вероятности (10.34). Соотношение (10.36) было использовано в подпрограмме **random2** из разд. 10.4.

Приведем пример использования метода обратного преобразования для функции распределения

$$p(x) = \begin{cases} ae^{-ax}, & \text{если } 0 \leq x \leq \infty, \\ 0, & x < 0. \end{cases} \quad (10.37)$$

Если подставить (10.37) в (10.29) и выполнить интегрирование, то получим $P(x) = 1 - e^{-ax}$. В этом случае решение уравнения (10.30) дает $x = -a^{-1} \ln(1 - r)$. Поскольку $1 - r$ распределена так же, как и r , мы можем написать

$$x = P^{-1}(r) = -a^{-1} \ln r. \quad (10.38)$$

Переменная x , полученная из (10.38), распределена с плотностью вероятности $p(x)$ по закону (10.37). Заметим, что на большинстве компьютеров вычисление натурального логарифма, входящего в (10.38), будет протекать с использованием разложения в степенной ряд для каждого значения r . Поэтому метод обратного преобразования может оказаться не самым эффективным.

Для применения метода обратного преобразования должны выполняться два условия. Вид функции $P(x)$ должен быть таким, чтобы интеграл (10.29) вычислялся аналитически или численно и чтобы было возможно разрешить соотношение $P(x) = r$ относительно x . К сожалению, гауссово, или нормальное, распределение

$$p(x) = \sigma^{-1}(2\pi)^{-1/2} e^{-x^2/2\sigma^2} \quad (10.39)$$

является примером распределения, для которого невозможно получить $P(x)$ аналитически. Однако мы можем сгенерировать двумерное гауссово распределение $p(x, y) dx dy$:

$$p(x, y) dx dy = \sigma^{-2} (2\pi)^{-1} e^{-(x^2 + y^2)/2\sigma^2} dx dy, \quad (10.40)$$

воспользовавшись методом Бокса—Мюллера. Перейдем к полярным координатам

$$r = (x^2 + y^2)^{1/2}, \quad \theta = \arctg y/x, \quad (10.41)$$

положим $\rho = r^2/2$ и запишем вероятность в виде

$$p(\rho, \theta) d\rho d\theta = (2\pi)^{-1} e^{-\rho} d\rho d\theta. \quad (10.42)$$

Анализ выражения (10.42) показывает, что если генерировать ρ рас-

предельным по экспоненциальному закону (10.37), а θ генерировать равномерно распределенным на отрезке $[0, 2\pi]$, то переменные

$$x = (2\rho)^{1/2} \cos \theta, \quad y = (2\rho)^{1/2} \sin \theta \quad (10.43)$$

будут распределены по закону (10.39) с нулевым средним и $\sigma = 1$. Другие процедуры получения гауссова распределения мы рассмотрим в задаче 11.12 и приложении 10В.

ЗАДАЧА 10.9. Неравномерные плотности вероятности

а. Напишите программу, которая позволит убедиться в том, что последовательность случайных чисел $\{x_n\}$, формируемая по формуле (10.38), распределена по экспоненциальному закону (10.37).

б. Постройте генератор случайных чисел, распределенных с плотностью вероятности

$$p(x) = \begin{cases} 2(1-x), & \text{если } 0 \leq x \leq 1, \\ 0 & \text{в противном случае.} \end{cases} \quad (10.44)$$

в. Убедитесь в том, что переменные x и y в (10.43) распределены в соответствии с гауссовым законом. Чему равны среднее значение и стандартное отклонение x и y ?

*г. Как можно с помощью выражений (10.43) получить распределение Гаусса с произвольными средним и дисперсией?

10.8. ВЫБОРКА ПО ЗНАЧИМОСТИ

Поскольку анализ, проведенный в разд. 10.6, показал, что оценка погрешности метода Монте-Карло пропорциональна σ , желательно познакомиться с методами «выборки по значимости»¹⁾, которые уменьшают σ и повышают эффективность каждого испытания. В качестве примера этого метода введем положительную функцию $p(x)$ такую, что

¹⁾В советской литературе используются также термины *выборка по важности* или *существенная выборка*. — Прим. перев.

$$\int_a^b dx p(x) = 1. \quad (10.45)$$

Тогда интеграл (10.1) можно переписать в виде

$$F = \int_a^b dx \left[\frac{f(x)}{p(x)} \right] p(x). \quad (10.46)$$

Мы можем вычислить интеграл (10.46), производя выборку в соответствии с «распределением вероятности» $p(x)$ и конструируя сумму

$$F_n = \frac{1}{n} \sum_{i=1}^n \frac{f(x)}{p(x)}. \quad (10.47)$$

Заметим, что для однородного случая $p(x) = 1/(b-a)$ и (10.47) переходит в (10.12).

Мы хотим выбрать функцию $p(x)$ так, чтобы дисперсия подынтегрального выражения $f(x)/p(x)$ была минимальна. Поскольку в общем случае вычислить σ аналитически не удастся, определим ее *a posteriori* и выберем функцию $p(x)$, которая ведет себя как $f(x)$ там, где $f(x)$ велика. Если мы сумеем подобрать подходящую $p(x)$, то подынтегральное выражение будет медленно меняющейся функцией и, следовательно, дисперсия уменьшится. В качестве примера опять рассмотрим интеграл (см. задачу 10.7)

$$F = \int_0^1 dx e^{-x^2}. \quad (10.48)$$

Оценка F с $p(x) = 1$ на отрезке $0 \leq x \leq 1$ приведена в первом столбце табл. 10.4. Подходящей весовой функцией может быть $p(x) = Ae^{-x}$, где A выбирается из условия нормировки $p(x)$ на единичном отрезке. Заметим, что $p(x)$ выбрана положительно-определенной и качественно подобна функции $f(x)$. Результаты представлены во втором столбце табл. 10.4. Видно, что, хотя машинное время, приходящееся на одно испытание, в неоднородном случае больше, использование неравномерного распределения вероятности эффективнее за счет меньшего значения σ .

ТАБЛИЦА 10.4. Оценки интеграла (10.48) методом Монте-Карло с однородной $p(x) = 1$ и неоднородной $p(x) = A \exp(-x)$ плотностями вероятности. Постоянная A выбрана из условия нормировки $p(x)$ на единичном отрезке. Точное значение интеграла приблизительно равно 0.7468. Приведены оценки F_n , стандартное отклонение σ и вероятная ошибка $\sigma/n^{1/2}$. Время ЦП (с) дано только для сравнения и относится к компьютеру Macintosh для программы, написанной на языке True BASIC

	$p(x) = 1$	$p(x) = Ae^{-x}$
n (испытаний)	20 000	1 000
F_n	0.7452	0.7482
σ	0.2009	0.0544
$\sigma/n^{1/2}$	0.0016	0.0017
Время ЦП на испытание (с)	0.0077	0.0280
Полное время ЦП	154	28

ЗАДАЧА 10.10. Выборка по значимости

а. Выберите в качестве весовой функции $p(x) = e^{-x}$ и вычислите интеграл

$$\int_0^{\infty} dx x^{3/2} e^{-x}. \quad (10.49)$$

б. Положите $p(x) = e^{-ax}$ и оцените интеграл

$$\int_0^{\pi} dx \frac{1}{x^2 + \cos^2 x}. \quad (10.50)$$

Определите такое значение a , которое минимизирует дисперсию этого интеграла.

10.9. МЕТОДЫ СЛУЧАЙНОГО БЛУЖДЕНИЯ

Один общий метод получения произвольного неравномерного распределения вероятности был предложен Метрополисом, Розенблатом и Теллером в 1953 г. Метод Метрополиса представляет собой частный случай процедуры выборки по значимости, в которой некоторые возможные выборки

отбрасываются (см. приложение 10В). В гл. 16 он будет применен в задачах статистической механики для генерирования больцмановского распределения вероятности.

Для простоты познакомимся с одномерным вариантом метода Метрополиса. Предположим, 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 стремится к распределению вероятности $p(x)$. Случайное блуждание определяется заданием *вероятности перехода* $w(x_i \rightarrow x_j)$ от одного значения x_i к другому — x_j , для того чтобы распределение точек x_0, x_1, x_2, \dots сходилось к $p(x)$. Можно показать, что достаточно (но не необходимо) удовлетворить условию «детального баланса»

$$p(x_i) w(x_i \rightarrow x_j) = p(x_j) w(x_j \rightarrow x_i). \quad (10.51)$$

Соотношение (10.51) не определяет $w(x_i \rightarrow x_j)$ единственным образом. Рассмотрим простейший вариант $w(x_i \rightarrow x_j)$, который удовлетворяет условию (10.51):

$$w(x_i \rightarrow x_j) = \min \left[1, \frac{p(x_j)}{p(x_i)} \right]. \quad (10.52)$$

Данный переход $w(x_i \rightarrow x_j)$ можно описать следующими шагами. Предположим, что «пешеход» находится в точке с координатой x_n . Для формирования x_{n+1} :

- 1) Выбираем пробную 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 этот переход и полагаем $x_{n+1} = x_t$.
- 4) Если $w < 1$, генерируем случайное число r .
- 5) Если $r \leq w$, принимаем этот переход и полагаем $x_{n+1} = x_t$.
- 6) Если пробный переход не принят, то полагаем $x_{n+1} = x_n$.

Прежде чем сформируется асимптотическое распределение вероятности с плотностью $p(x)$, необходимо произвести выборку ряда точек случайного блуждания. Каким выбрать максимальный «размер шага» δ ? Если δ слишком велико, то будет приниматься малая часть пробных шагов и выборка $p(x)$ будет неэффективна. С другой стороны, если δ слишком

мало, то приниматься будет большая часть пробных шагов, но опять выборка $p(x)$ будет неэффективна. Грубый критерий выбора величины δ заключается в том, что приниматься должно приблизительно от одной трети до половины шагов. Желательно также выбрать значение x_0 таким, чтобы распределение x_i достигало асимптотического распределения как можно скорей. Очевидно, что начинать случайное блуждание следует с такого значения x , при котором функция $p(x)$ максимальна.

Ниже приведена подпрограмма, реализующая алгоритм Метрополиса.

SUB Metropolis (x, delta, naccept)

DECLARE DEF p

LET xtrial = x + delta*(2*rnd - 1)

LET w = p(xtrial)/p(x)

IF rnd <= w **then**

LET x = xtrial

LET naccept = naccept + 1 ! число принятых переходов

END IF

END SUB

ЗАДАЧА 10.11. Распределение Гаусса

а. Используя подпрограмму **Metropolis**, разработайте программу генерирования гауссова распределения $p(x) = A \exp(-x^2/2\sigma^2)$. Является ли существенным численное значение нормировочной постоянной A ? Определите качественную зависимость коэффициента принятия и времени установления равновесия от максимального размера шага δ . Одним из критериев «равновесия» может быть выполнение условия $\langle x^2 \rangle \approx \sigma^2$. Каким целесообразно выбрать δ при $\sigma = 1$? Сколько необходимо провести испытаний до достижения равновесия при выбранном вами значении δ ? (Положите $x_0 = 0$.)

б. Модифицируйте свою программу так, чтобы она строила график асимптотического распределения вероятности, генерируемого алгоритмом Метрополиса.

*в. Вычислите автокорреляционную функцию $C(j)$, определяемую выражением

$$C(j) = \frac{\langle x_{i+j} x_i \rangle - \langle x_i \rangle^2}{\langle x_i x_i \rangle - \langle x_i \rangle^2}, \quad (10.53)$$

где $\langle \dots \rangle$ означает усреднение по случайному блужданию. Чему равно значение $C(j = 0)$? Каково было бы значение $C(j \neq 0)$, если бы x_i были совершенно случайными? Вычислите $C(j)$ для различных значений j и определите значение j , при котором функция $C(j)$ практически равна нулю?

ЗАДАЧА 10.12. Применение метода выборки по значимости

а. Хотя в данном случае метод Метрополиса не самый эффективный, используйте его для оценки интеграла

$$\int_0^4 dx x^2 e^{-x} \quad (10.54)$$

с $p(x) = x^2 e^{-x}$ на отрезке $0 \leq x \leq 4$. Постройте график числа посещений пешеходом точек x_0, x_1, x_2, \dots . Равномерно ли выбирается подынтегральная функция? Если нет, то в какой примерно области изменения переменной x отсчеты подынтегральной функции производятся чаще?

б. Вычислите интеграл

$$\int_0^1 dx x^2 e^{-x}, \quad (10.55)$$

используя эффективный метод выборки по значимости.

ЛИТЕРАТУРА

F. S. Acton, Numerical Methods That Work, Harper and Row, 1970. Прекрасная книга по численным методам.

S. E. Koonin, Computational Physics, Benjamin/Cummings, 1986. В гл. 8 рассматриваются многие вопросы по методу Монте-Карло, отраженные в данной главе. [Готовится перевод: С. Кунин. Вычислительная физика. — М.: Мир, 1991.]

M. H. Calos, P. W. Whitlock, Monte Carlo Methods, Vol. 1: Basics, John Wiley & Sons, 1986. Авторы — известные специалисты по методам Монте-Карло.

W. H. Press, B. P. Flannery, S. A. Teukolsky, W. T. Vetterling, Numerical Recipes, Cambridge University Press, 1986. [Имеется перевод: *В. П. Пресс, Б. П. Фланнери, С. А. Тьюкольски, В. Т. Веттерлинг*. Численные рецепты. — М.: Мир, 1990.] Проявление изобретательности в искусстве научных расчетов. Полезный справочник.

R. Y. Rubinstein, Simulation and the Monte Carlo Method, John Wiley & Sons, 1981. Современная, но ясно написанная книга по методам Монте-Карло.

G. L. Squires, Practical Physics, 3rd ed., Cambridge University Press, 1985. Введение в практические задачи анализа ошибок.

J. R. Taylor, An Introduction to Error Analysis, University Science Books, Oxford University Press, 1982. [Имеется перевод: *Дж. Тейлор*. Введение в теорию ошибок. — М.: Мир, 1985.]

H. D. Young, Statistical Treatment of Experimental Data, McGraw-Hill, 1962.

ДОПОЛНИТЕЛЬНАЯ ЛИТЕРАТУРА

Метод статистических испытаний/ Под ред. *Ю. А. Шнейдера*. — М.: ГИФМЛ, 1962. Книга написана на том же уровне, что и эта глава. В гл. 1, 2 рассматриваются вопросы оценок погрешности, приложения к вычислению интегралов, вопросам генерирования случайных чисел. В гл. 6 обсуждаются вопросы генерирования последовательностей равномерно распределенных случайных чисел и статистические критерии, предъявляемые к ним.

И. М. Соболев, Численные методы Монте-Карло. — М.: Наука, 1973. Гл. 1 посвящена получению случайных величин на компьютере. В § 1 гл. 2 описан метод обратного преобразования (обратных функций), а в § 5 рассмотрен метод отбора-отказа.

Методы Монте-Карло в статистической физике/ Под ред. *М. Калоса*. — М.: Мир, 1984. Книга рассчитана на подготовленного читателя и посвящена в основном современным приложениям метода (см. гл. 17).

ПРИЛОЖЕНИЕ 19А. ОЦЕНКИ ПОГРЕШНОСТЕЙ ЧИСЛЕННОГО ИНТЕГРИРОВАНИЯ

Выведем зависимость погрешности аппроксимации от количества отрезков разбиения для методов численного интегрирования, рассмотренных в разд. 10.1 и 10.3. Эти оценки основаны на предположении о существовании представления подынтегральной функции $f(x)$ рядом Тейлора

$$f(x) = f(x_i) + f'(x_i)(x - x_i) + \frac{1}{2}f''(x_i)(x - x_i)^2 + \dots \quad (10.56)$$

и интегрировании (10.1) на отрезке $x_i \leq x \leq x_{i+1}$:

$$\int_{x_i}^{x_{i+1}} dx f(x) = f(x_i)\Delta x + \frac{1}{2}f'(x_i)(\Delta x)^2 + \frac{1}{6}f''(x_i)(\Delta x)^3 + \dots \quad (10.57)$$

Сначала оценим погрешность метода прямоугольников, в котором функция $f(x)$ вычисляется на левом конце каждого отрезка. Погрешность Δ_i на отрезке $[x_i, x_{i+1}]$ равняется разности между (10.57) и оценкой $f(x_i)\Delta x$:

$$\Delta_i = \left[\int_{x_i}^{x_{i+1}} dx f(x) \right] - f(x_i)\Delta x \approx \frac{1}{2}f'(x_i)(\Delta x)^2 \quad (10.58)$$

Мы видим, 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 прямоугольников по порядку величины равна $n\Delta_i \approx n(\Delta x)^2$, или $O(n^{-1})$.

Оцененную погрешность формулы трапеций можно найти аналогичным образом. Погрешность на отрезке $[x_i, x_{i+1}]$ определяется разностью между точным значением интеграла и оценкой $[f(x_i) + f(x_{i+1})]\Delta x/2$:

$$\Delta_i = \left[\int_{x_i}^{x_{i+1}} dx f(x) \right] - \frac{1}{2}[f(x_i) + f(x_{i+1})]\Delta x. \quad (10.59)$$

Если для оценки интеграла использовать формулу (10.57), а для оцен-

ки функции $f(x_{i+1})$ в (10.59) — выражение (10.56), то увидим, что член, пропорциональный f' , сокращается и погрешность на одном отрезке равна $O((\Delta x)^3)$. Отсюда полная погрешность формулы трапеций на отрезке $[a, b]$ равна $O(n^{-2})$.

Поскольку формула Симпсона основывается на приближении функции $f(x)$ параболой на отрезке $[x_{i-1}, x_{i+1}]$, члены погрешности, пропорциональные f'' , сокращаются. Можно было бы ожидать, что вклад внесут члены погрешности порядка $f'''(x_i)(\Delta x)^4$, но они сокращаются в силу их симметричности. Следовательно, формула Симпсона правильно описывает член $(\Delta x)^4$ в разложении функции $f(x)$ в ряд Тейлора. Если в разложении функции $f(x)$ в ряд Тейлора сохранить член порядка $(\Delta x)^4$, то найдем, что погрешность на отрезке $[x_i, x_{i+1}]$ по порядку величины равна $f'''(x_i)(\Delta x)^5$, а полная погрешность формулы Симпсона на отрезке $[a, b]$ составляет $O(n^{-4})$.

Обобщим теперь оценки погрешности на двумерный случай. Двумерный интеграл от $f(x, y)$ представляет собой объем под поверхностью, определяемой функцией $f(x, y)$. В «прямоугольном» приближении этот интеграл запишется в виде суммы объемов параллелепипедов с площадью основания $\Delta x \Delta y$ и высотой, равной значению $f(x, y)$ в одном из углов. Для определения погрешности разложим $f(x, y)$ в ряд Тейлора:

$$f(x, y) = f(x_i, y_i) + f'_x(x_i, y_i)(x - x_i) + f'_y(x_i, y_i)(y - y_i) + \dots, \quad (10.60)$$

где f'_x и f'_y обозначают частные производные $f(x, y)$ соответственно по x и y . Запишем погрешность в виде

$$\Delta_i = \left[\iint f(x, y) dx dy \right] - f(x_i, y_i) \Delta x \Delta y. \quad (10.61)$$

Подставим (10.60) в (10.61) и почленно проинтегрируем. Получаем, что член, пропорциональный f , сокращается, а интеграл от $(x - x_i) dx$ дает $(\Delta x)^2/2$. Интеграл от этого выражения по dy дает еще один множитель Δy . Аналогичный вклад дает интеграл от члена, пропорционального $(y - y_i)$. Поскольку Δy составляет тоже $O(\Delta x)$, то соответствующая погрешность на отрезках $[x_i, x_{i+1}]$ и $[y_i, y_{i+1}]$ равна

$$\Delta_i \approx \frac{1}{2} [f'_x(x_i, y_i) + f'_y(x_i, y_i)] (\Delta x)^3. \quad (10.62)$$

Мы видим, что погрешность, связанная с одним параллелепипедом, составляет $O((\Delta x)^3)$. Поскольку всего имеется n параллелепипедов, пол-

ная погрешность по порядку величины равна $n(\Delta x)^3$. Однако в двумерном случае $n = A/(\Delta x)^2$ и, следовательно, полная погрешность составляет $O(n^{-1/2})$. По сравнению с этим в одномерном случае полная погрешность равна $O(n^{-1})$.

Соответствующие оценки погрешности для двумерных обобщений формул трапеций и Симпсона соответственно равны $O(n^{-1})$ и $O(n^{-2})$. Вообще если для одномерного случая погрешность составляет $O(n^{-a})$, то в d -мерном случае она равна $O(n^{-a/d})$. В противоположность этому погрешности метода Монте-Карло не зависят от размерности и меняются как $O(n^{-1/2})$. Таким образом, для достаточно больших d интегрирование по методу Монте-Карло будет приводить к меньшим погрешностям при тех же значениях n .

ПРИЛОЖЕНИЕ 10Б. АНАЛИТИЧЕСКИЙ ВЫВОД СТАНДАРТНОГО ОТКЛОНЕНИЯ ОТ СРЕДНЕГО

В разд. 10.6 мы привели эмпирические соображения в обоснование того, что ошибка одного измерения из n испытаний равняется σ/\sqrt{n} , где σ —стандартное отклонение для одного измерения. Сейчас мы приведем аналитический вывод этого соотношения. Обозначим измеряемую в эксперименте величину через x . Рассмотрим m наборов измерений, каждое состоит из n испытаний, итого общее число испытаний равно mn . Будем нумеровать индексом α отдельные измерения, а индексом i —испытания в каждом измерении. Обозначим через $x_{\alpha,i}$ i -е измерение в наборе с номером α . Значение измерения дается формулой

$$M_{\alpha} = \frac{1}{n} \sum_{i=1}^n x_{\alpha,i}. \quad (10.63)$$

Среднее \overline{M} полного числа mn испытаний равно

$$\overline{M} = \frac{1}{m} \sum_{\alpha=1}^m M_{\alpha} = \frac{1}{mn} \sum_{\alpha=1}^m \sum_{i=1}^n x_{\alpha,i}. \quad (10.64)$$

Разность между измерением α и средним по всем измерениям равна

$$e_{\alpha} = M_{\alpha} - \overline{M}. \quad (10.65)$$

Следовательно, мы можем записать дисперсию этих средних как

$$\sigma_m^2 = \frac{1}{m} \sum_{\alpha=1}^m e_{\alpha}^2. \quad (10.66)$$

Теперь мы хотим связать величину σ_m с дисперсией отдельных испытаний. Отклонение отдельного испытания от среднего равно

$$d_{\alpha,i} = x_{\alpha,i} - \bar{M}. \quad (10.67)$$

Следовательно, дисперсия σ^2 mn отдельных испытаний равна

$$\sigma^2 = \frac{1}{mn} \sum_{\alpha=1}^m \sum_{i=1}^n d_{\alpha,i}^2 \quad (10.68)$$

Запишем

$$\begin{aligned} e_{\alpha} &= M_{\alpha} - \bar{M} = \frac{1}{n} \sum_{i=1}^n x_{\alpha,i} - \bar{M} = \\ &= \frac{1}{n} \sum_{i=1}^n (x_{\alpha,i} - \bar{M}) = \frac{1}{n} \sum_{i=1}^n d_{\alpha,i}. \end{aligned} \quad (10.69)$$

Если подставить (10.69) в (10.66), то получим

$$\sigma^2 = \frac{1}{m} \sum_{\alpha=1}^n \left[\frac{1}{n} \sum_{i=1}^n d_{\alpha,i} \right] \left[\frac{1}{n} \sum_{j=1}^n d_{\alpha,j} \right]. \quad (10.70)$$

Суммирование в (10.70) по испытаниям i и j в наборе с номером α содержит два сорта членов с $i = j$ и $i \neq j$. Мы предполагаем, что $d_{\alpha,i}$ и $d_{\alpha,j}$ являются независимыми и в среднем половина из них положительна, а другая половина отрицательна. Следовательно, в пределе очень большого числа измерений можно ожидать, что сохраняться в формуле (10.70) будут только члены с $i = j$. В результате, имеем

$$\sigma^2 = \frac{1}{mn^2} \sum_{\alpha=1}^m \sum_{i=1}^n d_{\alpha,i}^2 \quad (10.71)$$

Если мы объединим (10.71) и (10.68), то получим требуемый результат:

$$\sigma_m^2 = \frac{\sigma^2}{n}. \quad (10.72)$$

ПРИЛОЖЕНИЕ 10В. МЕТОД ОТБОРА-ОТКАЗА

Хотя метод обратного преобразования, рассмотренный в разд. 10.7, в принципе можно использовать для генерирования любого требуемого распределения вероятностей, на практике это метод ограничивается функциями, для которых обратное преобразование $P^{-1}(r)$ можно найти аналитически или с помощью несложного численного приближения. Другим методом генерирования неравномерных распределений вероятностей является метод *отбора-отказа*, предложенный фон Нейманом. В основе этого метода лежит простое геометрическое соображение, применяемое при вычислении определенных интегралов методом проб и ошибок, который обсуждался в разд. 10.4.

Предположим, что $p(x)$ — нормированная функция распределения вероятности, которую мы хотим генерировать. Для простоты будем считать $p(x)$ не равной нулю на единичном отрезке. Рассмотрим положительно-определенную функцию сравнения $w(x)$ такую, что $w(x) > p(x)$ на всем интересующем нас интервале. Простым, хотя не всегда оптимальным выбором w является константа, превышающая максимальное значение $p(x)$. Поскольку площадь под кривой $p(x)$ в интервале от x до $x + \Delta x$ равна вероятности попадания x в этот интервал, можем следовать процедуре, аналогичной той, которая использовалась в методе проб и ошибок. Генерируем два случайных числа, определяющие расположение случайной точки в двумерном пространстве, которая распределена равномерно под функцией сравнения $w(x)$. Если эта точка не попадает под кривую $p(x)$, то мы ее отбрасываем, если она попадает под нее, то оставляем. В этом методе подразумевается, что принимаемые точки расположены равномерно под кривой $p(x)$ и ее координаты x распределены в соответствии с $p(x)$.

Одна из процедур генерирования случайных точек (x, y) , равномерно распределенных под заданной функцией сравнения $w(x)$, заключается в следующем:

1. Выбираем $w(x)$. В качестве одного из вариантов выбора можно подобрать такую $w(x)$, чтобы значения x , распределенные в соответствии с $w(x)$, можно было бы генерировать методом обратного преобразования. Пусть полная площадь под кривой $w(x)$ равна A .
2. Генерируем равномерно распределенное случайное число на отрезке $[0, A]$ и используем его для получения соответствующего значения x , распределенного согласно $w(x)$.
3. Для значения x , сгенерированного на шаге 2, получаем равномерно распределенное на отрезке $[0, w(x)]$ случайное число y . Точка (x, y) равномерно распределена по площади, расположенной под функцией сравнения $w(x)$. Если $y \leq p(x)$, то x принимается в качестве случайного числа, распределенного в соответствии с $p(x)$.

Конечно, шаги 2 и 3 нужно много раз повторить.

Заметим, что метод отбора-отказа эффективен, только если функция сравнения $w(x)$ близка к $p(x)$ на всем рассматриваемом интервале.

СЛУЧАЙНОЕ БЛУЖДЕНИЕ

11

В этой главе мы познакомимся с понятиями вероятности и случайных процессов на примере модельных физических систем.

11.1. ВВЕДЕНИЕ

Первоначальную формулировку задачи о «случайных блужданиях» предложил Пирсон в 1906 г. Если пьяный делает N шагов равной длины от фонарного столба в произвольных направлениях, то как далеко отойдет он от этого столба? Со времени такой формулировки случайного блуждания модели случайного блуждания получили широкое распространение в физике, биологии и общественных науках. Хорошо знакомыми по учебникам приложениями являются диффузия молекулы в газе и броуновское движение коллоидных взвесей в жидкости. Другим важным приложением методов случайного блуждания служит моделирование длинных полимерных цепочек, рассматриваемое в разд. 11.4. В разд. 10.9 мы узнали, что с помощью методов случайного блуждания можно вычислять определенные интегралы. В действительности многие задачи, как, например, решение уравнения Шредингера, можно переформулировать на языке случайного блуждания (см. гл. 17). Однако, чтобы хорошо разбираться в этих приложениях, сначала необходимо изучить простые модели.

11.2. ОДНОМЕРНОЕ СЛУЧАЙНОЕ БЛУЖДЕНИЕ

Хотя пьяницы не являются основным объектом изучения физики, рассмотрим сначала идеализированную одномерную задачу о движении пьяного (пешехода), который начинает двигаться от фонарного столба, расположенного в точке $x = 0$ (рис. 11.1). Все шаги имеют одинаковую длину l . Направление каждого шага пешехода не зависит от направления предыдущего. На каждом шаге по времени пешеход делает шаг вправо с вероятностью p и шаг влево с вероятностью $q = (1 - p)$. Пусть n_{\rightarrow} обозначает количество шагов вправо, а n_{\leftarrow} — влево. Полное число шагов равно $N = n_{\rightarrow} + n_{\leftarrow}$. Тогда полное смещение пешехода от начала координат после N шагов равно $x = (n_{\rightarrow} - n_{\leftarrow})l$, где $-Nl \leq x \leq Nl$. Основной интересующей нас величиной является вероятность $P_N(x)$ того, что после N шагов пешеход окажется на расстоянии x от столба. Мы

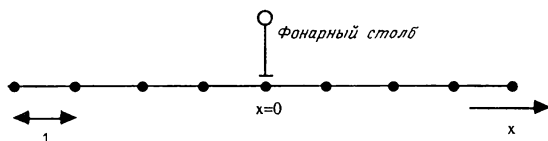


Рис. 11.1. Одномерное случайное блуждание пьяного.

можем вычислить среднее смещение $\langle x_N \rangle$ и дисперсию $\langle \Delta x_N^2 \rangle$ смещения пешехода по формулам

$$\langle x_N \rangle = \sum_{x=-Nl}^{Nl} x P_N(x) \quad (11.1)$$

и

$$\langle \Delta x_N^2 \rangle = \langle x_N^2 \rangle - \langle x_N \rangle^2, \quad (11.2)$$

где

$$\langle x_N^2 \rangle = \sum_{x=-Nl}^{Nl} x^2 P_N(x). \quad (11.3)$$

Усреднение производится по всевозможным блужданиям, состоящим из N шагов.

Исследование задачи случайного блуждания, сформулированной выше, можно провести аналитически, воспользовавшись теорией вероятностей (см. учебник Рейфа). В результате для величин $\langle x_N \rangle$ и $\langle \Delta x_N^2 \rangle$ получаются аналитические выражения

$$\langle x_N \rangle = (p - q)Nl \quad (11.4)$$

и

$$\langle \Delta x_N^2 \rangle = 4pqNl^2. \quad (11.5)$$

Заметим, что, согласно формуле (11.4), в симметричном случае $p = q = 1/2$ получаем среднее $\langle x_N \rangle = 0$.

Теперь рассмотрим наш пример случайного блуждания с другой точки зрения: на языке диффузии молекулы в разреженном газе. Предположим, что молекула пробегает расстояние ℓ между столкновениями с другими молекулами. Если допустить, что последовательные смещения молекулы между столкновениями статистически независимы, то движение молекулы будет тождественно блужданию пьяного. Поскольку движение такой молекулы можно к тому же описать некоторым диффузионным процессом, опишем кратко связь случайного блуждания с диффузией. В разд. 11.6

мы установим, что процесс диффузии частично характеризуется линейным соотношением

$$\langle \Delta R(t)^2 \rangle = 2dDt, \quad (11.6)$$

связывающим между собой время t и среднеквадратичное смещение молекулы $\langle \Delta R(t)^2 \rangle$ из начального положения при $t = 0$. Коэффициент пропорциональности D в формуле (11.6) называется *коэффициентом самодиффузии* молекулы, а d — размерность пространства. Чтобы сопоставить (11.5) и (11.6), положим время между шагами равным τ , так что $N = t/\tau$, и отождествим длину свободного пробега ℓ с шагом пешехода l . Тогда формулу (11.5) можно переписать в виде

$$\langle \Delta x^2 \rangle = 4pq\ell^2(t/\tau). \quad (11.7)$$

Сравнивая (11.6) и (11.7), мы делаем заключение, что коэффициент диффузии случайно блуждающего пешехода в одномерном случае выражается при $p = 1/2$ формулой $D = \ell^2/2\tau$.

Хотя соотношения (11.4) и (11.5) можно вывести, используя простые аналитические методы, нам необходимо разработать метод моделирования блужданий, которые не имеют точных решений. Два важных метода моделирования — точный комбинаторный метод (полного перебора) и метод Монте-Карло.

В методе, основанном на полном переборе, количество и вероятность всех блужданий для заданных N и x вычисляются в явном виде. В качестве примера на рис. 11.2 показано восемь блужданий для $N = 3$ и $d = 1$. Заметим, что число блужданий для положительных и отрицательных значений x одинаково. Перебор всех блужданий позволяет вычислить $P_3(x)$ (рис. 11.2), и мы получим

$$\begin{aligned} \langle x_3 \rangle &= \sum x P_3(x) = -3q^3 - 3pq^2 + 3p^2q + 3p^3 = \\ &= 3(p^2 - q^2)(p + q) = \\ &= 3(p + q)(p - q) = 3(p - q), \end{aligned} \quad (11.8a)$$

$$\begin{aligned} \langle x_3^2 \rangle &= \sum x^2 P_3(x) = 9q^3 + 3pq^2 + 3p^2q + 9p^3 = \\ &= 12pq + [3(p - q)]^2. \end{aligned} \quad (11.8b)$$

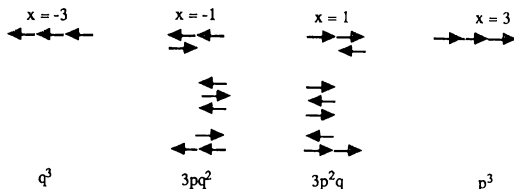


Рис. 11.2. Восемь случайных блужданий в одномерном ($d = 1$) случае при $N = 3$ и полном смещении x . Для наглядности отдельные шаги изображены смещенными по вертикали. Приведена также вероятность $P_3(x)$, где p — вероятность шага вправо, а $q = 1 - p$ — вероятность шага влево.

В формулах (11.8) мы использовали связь $p + q = 1$. Из (11.8) находим, что среднеквадратичное смещение выражается формулой $\langle \Delta x^2 \rangle = \langle x_3^2 \rangle - \langle x_3 \rangle^2 = 12pq$, что совпадает с (11.5). В методе полного перебора компьютер используется как «счетовод», который генерирует возможные блуждания и в конце концов получает точные выражения для рассматриваемых величин. Поскольку полное число возможных N -шаговых блужданий в одномерном случае составляет 2^N , то метод полного перебора, вообще говоря, ограничен малыми значениями N . (Конечно, это ограничение можно преодолеть, если модель решается аналитически.) Мы будем пользоваться результатами полного перебора для проверки наших вычислений методом Монте-Карло.

ЗАДАЧА 11.1. Точный расчет одномерных случайных блужданий

а. Для $N = 3$ имеются три блуждания с $x = 1$ и три блуждания с $x = -1$ (рис. 11.2). Сколько имеется блужданий для $N = 4$ с $x = 0$? Как это число связано с числом блужданий для случая $x = \pm 1$ и $N = 3$? Как связано число четырехшаговых блужданий с $x = 2$ с числом трехшаговых блужданий с $x = 1$ и $x = 3$? Получите точное соотношение между числом N -шаговых блужданий со смещением x и числом $(N - 1)$ -шаговых блужданий со смещением $x \pm 1$. Используя полученное соотношение, определите число возможных блужданий для заданного x и значений $N = 4$ и $N = 5$, а также точные результаты для $P_N(x)$, $\langle x_N \rangle$ и $\langle \Delta x_N^2 \rangle$ при $N = 4$ и $N = 5$.

б. Пользуясь соотношением, полученным в п. «а», составьте программу, в которой перебираются возможные блуждания для заданных значений N и x . Вычислите точные значения $P_N(x)$, $\langle x_N \rangle$ и $\langle \Delta x_N^2 \rangle$.

для N от 1 до 8. Используя полученные результаты точного расчета, покажите, что $P_N(x)$ можно записать в виде

$$P_N(x) = \frac{N!}{(\frac{N}{2} + \frac{x}{2})! (\frac{N}{2} - \frac{x}{2})!} p^{(\frac{N}{2} - \frac{x}{2})} q^{(\frac{N}{2} + \frac{x}{2})}, \quad (11.9)$$

где $(N + \frac{1}{2}x)$ и $(N - \frac{1}{2}x)$ — число шагов вправо и влево соответственно.

В противоположность методу полного перебора всех блужданий из N шагов для малых значений N в методе Монте-Карло моделируются блуждания, состоящие из многих шагов, например N от 100 до 1000. Генерируется большое число блужданий с искомыми вероятностями, которые обеспечивают принадлежность выбранных блужданий множеству всех допустимых блужданий. Понятно, что чем больше используется блужданий, тем точнее получается результат. На практике требуемую точность получают, сравнивая результаты для все большего числа испытаний, пока они не станут меняться с заданной степенью точности.

Ниже приведен пример программы моделирования методом Монте-Карло одномерного случайного блуждания; в подпрограмме **walk** реализован алгоритм случайного блуждания.

```

PROGRAM random_walk1
! моделирование методом Монте-Карло случайного блуждания для d = 1
DIM prob(-64 to 64)
CALL initial(p,N,ntrial)
FOR itrial = 1 to ntrial
  CALL walk(x,p,N)
  CALL data(x,xcum,x2cum,prob) ! сбор данных после N шагов
NEXT itrial
CALL average(N,ntrial,xcum,x2cum,prob)
END

SUB initial(p,N,ntrial)
  RANDOMIZE
  LET ntrial = 100 ! число испытаний
  LET p = 0.5 ! вероятность шага вправо
  INPUT prompt "число шагов = ": N
END SUB

```

```

SUB walk(x,p,N)
  LET x = 0          ! начальное положение для каждого испытания
  FOR istep = 1 to N
    IF rnd <= p then
      LET x = x + 1
    ELSE
      LET x = x - 1
    END IF
  NEXT istep
END SUB

SUB data(x,xcum,x2cum,prob())
  LET xcum = xcum + x
  LET x2cum = x2cum + x*x
  LET prob(x) = prob(x) + 1
END SUB

SUB average(N,ntrial,xcum,x2cum,prob())
! средние значения для N-шагового блуждания
  LET norm = 1/ntrial
  LET xbar = xcum*norm
  LET x2bar = x2cum*norm
  FOR x = -N to N
    LET prob(x) = prob(x)*norm
    IF prob(x) > 0 then PRINT x,prob(x)
  NEXT x
  LET variance = x2bar - xbar*xbar
  LET sigma = sqr(variance)
  PRINT "среднее смещение = "; xbar
  PRINT "sigma = "; sigma
END SUB

```

Для вычисления распределения вероятности $P_N(x)$ необходимо рассмотреть большое число испытаний и зафиксировать относительное число появлений пешехода в точке с координатой x после N шагов. Отдельно вычислять в программе $\langle x_N \rangle$ и $\langle \Delta x_N^2 \rangle$ не нужно, поскольку эти величины можно получить из $P_N(x)$. Однако во многих приложениях величины $\langle x_N \rangle$ и $\langle \Delta x_N^2 \rangle$ уже достаточно информативны и нет необходимости вычислять распределение $P_N(x)$, требующее большого объема памяти.

ЗАДАЧА 11.2. Моделирование одномерных случайных блужданий методом Монте-Карло

а. Модифицируйте программу `random_walk` таким образом, чтобы после каждых 100 испытаний выводилась траектория каждого случайного пешехода и строился график $P_N(x)$. Задайте в своей модифицированной программе параметры $p = 0.5$, $N = 64$ и $ntrial = 1000$. Воспринимаются ли траектории каждого пешехода визуально как «случайные»? Как качественно меняется вид функции $P_N(x)$ в зависимости от числа испытаний?

б. Используя программу `random_walk` с параметром $p = 0.5$, вычислите $P_N(x)$, $\langle x_N \rangle$ и $\langle \Delta x_N^2 \rangle$ при $N = 8, 16, 32$ и 64 . Сравните полученные для $\langle x_N \rangle$ и $\langle \Delta x_N^2 \rangle$ результаты с соответствующими точными ответами (11.4) и (11.5). Если необходимо вычислить $\langle \Delta x_N^2 \rangle$ с одинаковой относительной точностью для $N = 8$ и $N = 64$, то в каком случае потребуется больше испытаний?

в. Постройте график $P_N(x)$ для приведенных выше значений N . Является ли $P_N(x)$ непрерывной функцией? Чему равно значение $P_N(x)$ в точке максимума для каждого значения N ? Чему равна приближенная «ширина» $P_N(x)$ в каждом случае?

г. Используя график $P_N(x)$, покажите, что при достаточно больших N распределение $P_N(x)$ может быть аппроксимировано гауссовым распределением

$$P(x) = \sigma^{-1}(2\pi)^{-1/2} \exp[-(x - \langle x \rangle)^2 / 2\sigma^2], \quad (11.10)$$

где $\sigma^2 = \langle \Delta x^2 \rangle$. Подставьте вычисленные значения $\langle x \rangle$ и $\langle \Delta x^2 \rangle$ в формулу (11.10). Для всех ли значений x выражение (11.10) аппроксимирует рассчитанное распределение одинаково хорошо?

д. Используя анализ погрешностей, рассмотренный в разд. 10.6, оцените число испытаний, необходимое для получения $\langle \Delta x_N^2 \rangle$ при $N = 8$ и $N = 64$ с точностью 5%. Данный анализ приводится в приложении 11А. Чему равно стандартное отклонение $\langle \Delta x_N^2 \rangle$?

е. Положите $p = 0.7$. Вычислите $\langle x_N \rangle$ и $\langle x_N^2 \rangle$ для тех же значений N , что и в п. «б». Как можно интерпретировать $\langle x_N \rangle$ в этом случае?

Во многих приложениях моделей случайного блуждания при больших значениях N используются асимптотические результаты. Например, во многих моделях случайного блуждания величина $\langle \Delta x_N^2 \rangle$ удовлетворяет при достаточно больших N степенному закону, например

$$\langle \Delta x_N^2 \rangle \sim N^{2\nu} \quad (N \gg 1). \quad (11.11)$$

Соотношение (11.11) является примером асимптотического *масштабного закона*. Другими словами, при удвоении числа шагов среднеквадратичное смещение пешехода увеличивается в 2^ν раз. Для одномерного случайного блуждания из (11.5) находим, что $\nu = 1/2$. Во многих задачах, которые будут рассматриваться в этой главе, исследуется, существует ли степенной закон и зависит ли показатель степени ν от структуры и размерности решетки, а также от типа блуждания.

ЗАДАЧА 11.3. Асимптотические свойства одномерного случайного блуждания

а. Постройте в логарифмическом масштабе график зависимости $\langle \Delta x_N^2 \rangle$ от N и оцените показатель степени ν , определенный в формуле (11.11). Метод наименьших квадратов для оценки ν приведен в приложении 11А.

б. Другой важной величиной, характеризующей случайное блуждание, является $\langle S_N \rangle$ — среднее число *различных* узлов решетки, посещаемых во время N -шагового блуждания. Вычислите методом Монте-Карло $\langle S_N \rangle$ и определите асимптотическую зависимость $\langle S_N \rangle$ от N .

11.3. ОБОБЩЕНИЯ МЕТОДА СЛУЧАЙНЫХ БЛУЖДАНИЙ

Модели случайного блуждания не ограничиваются одномерным случаем и применяются не только к блужданию пьяных или движению молекул. В следующих ниже задачах рассматриваются некоторые из более общих и часто используемых моделей случайного блуждания и указываются некоторые приложения. Другие приложения можно найти в литературе, приведенной в конце главы.

Статистический характер задачи случайного блуждания означает, что мы рассматриваем либо большое число последовательных блужданий, как в задачах 11.2 и 11.3, либо большое число одинаковых пешеходов, движущихся одновременно. В следующей задаче мы рассмотрим движение

многих пешеходов, которые перемещаются случайно и независимо друг от друга.

ЗАДАЧА 11.4. Простая задача двумерного случайного блуждания

Рассмотрим «рой» из M «пчел», расположенный изначально в единичном круге с центром в начале координат. На каждом шаге по времени каждая пчела движется случайным образом равновероятно в одном из четырех возможных направлений: на север, юг, восток или запад. В приведенной ниже программе реализуется этот алгоритм и положение пчел изображается точками на экране.

```
PROGRAM random_walk2      ! двумерное случайное блуждание
DIM x(200),y(200)
CALL initial(N,M,x,y)
CALL move(N,M,x,y)
END

SUB initial(N,M,x(),y())
  RANDOMIZE
  INPUT prompt "полное число шагов = ": N
  INPUT prompt "число частиц = ": M
  LET aspect_ratio = 1.5
  LET ymax = 0.5*N
  LET xmax = aspect_ratio*ymax
  SET window -xmax,xmax,-ymax,ymax
  ! частицы располагаются случайно внутри единичной окружности
  FOR i = 1 to M
    LET r = 2*rnd - 1
    LET theta = 2*pi*rnd
    LET x(i) = r*cos(theta)
    LET y(i) = r*sin(theta)
    PLOT POINTS: x(i),y(i)
  NEXT i
END SUB
```

```

SUB  move(N,M,x(),y())
  FOR istep = 1 to N
    FOR i = 1 to M
      CALL choice(i,x,y)
    NEXT i
    CLEAR
    FOR i = 1 to M
      PLOT POINTS: x(i), y(i)
    NEXT i
  NEXT istep
END SUB

```

```

SUB choice(i,x(),y())      ! алгоритм двумерного случайного блуждания
  LET prob = rnd
  IF prob <= 0.25 then
    LET x(i) = x(i) + 1
  ELSEIF prob <= 0.5 then
    LET x(i) = x(i) - 1
  ELSEIF prob <= 0.75 then
    LET y(i) = y(i) - 1
  ELSE
    LET y(i) = y(i) + 1
  END IF
END SUB

```

а. Выполните программу **random_walk2** и качественно опишите характер движения пчелиного роя.

б. Предположим, что каждая пчела расположена случайным образом внутри единичной окружности и имеет случайную начальную скорость в одном из четырех возможных направлений. В течение каждого временного интервала каждая пчела делает шаг единичной длины в направлении своей начальной скорости. Отличается ли движение данного роя пчел от движения роя в п. «а»?

в. Используя первоначальную формулировку задачи случайного блуждания, вычислите $\langle x_N \rangle$, $\langle y_N \rangle$, $\langle \Delta x_N^2 \rangle$ и $\langle \Delta y_N^2 \rangle$ как функции от числа шагов N . Усреднение выполняется по M пчелам. Вычислите также средний квадрат полного смещения $\langle \Delta R_N^2 \rangle$, определяемый как

$$\langle \Delta R_N^2 \rangle = \langle x_N^2 \rangle + \langle y_N^2 \rangle - \langle x_N \rangle \langle x_N \rangle - \langle y_N \rangle \langle y_N \rangle. \quad (11.12)$$

Каковы качественные зависимости этих величин от числа шагов? Вычислите зависимость R_{\max}^2 от N , где R_{\max}^2 — максимальное смещение пчел на N -м шаге. Различаются ли качественно зависимости R_{\max}^2 и $\langle \Delta R_N^2 \rangle$ от N для всех N ?

г. Повторите вычисления п. «в», используя формализм случайной скорости, а не случайного блуждания. Различаются ли качественно усредненные движения пчел? Если да, то объясните природу и причину этого различия.

В задаче 11.4 мы допустили, что каждый шаг имеет равную длину и совершается в одном из четырех направлений. Следовательно, мы неявно предположили, что каждое блуждание происходит на *квадратной* решетке, как показано на рис. 11.3,а. Для квадратной решетки *координационное число* z , число ближайших соседних узлов, равно четырем. Используется и другая двумерная решетка — *треугольная* (рис. 11.3,б), у которой $z = 6$. Как говорилось в гл. 6, двумерные твердые тела обычно имеют треугольную решетку, а не квадратную. Если не оговорено особо, мы считаем период решетки равным единице.

Поскольку мы часто будем рассматривать поведение среднего квадрата полного смещения для больших N [см. (11.12)], то удобно ввести обозначение:

$$R_N \equiv \sqrt{\langle \Delta R_N^2 \rangle} \quad (11.13)$$

Величину R_N называют *среднеквадратичным смещением* и ее не следует путать с абсолютным значением вектора R_N . Показатель степени ν определяется асимптотическим соотношением (большие N)

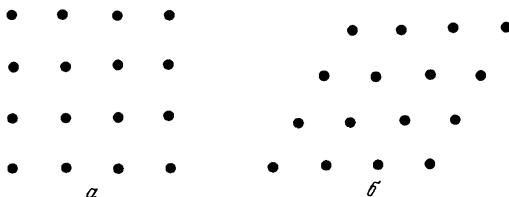


Рис. 11.3. а — пример квадратной решетки: каждый узел решетки имеет четыре соседа. б — пример треугольной решетки: каждый узел решетки имеет шесть соседей. Расстояние между рядами по вертикали равно $(1/2)\sqrt{3}$. Каждый ряд узлов сдвинут по горизонтали на $1/2$ относительно соседних рядов. Период решетки равен единице.

$$R_N \sim aN^\nu, \quad (11.14)$$

где a — постоянная величина порядка единицы.

ЗАДАЧА 11.5. Случайные блуждания на двумерных и трехмерных решетках

а. Перечислите все случайные блуждания на квадратной решетке с $N = 4$ и получите точные значения для $\langle x_N \rangle$, $\langle y_N \rangle$ и $\langle \Delta R_N^2 \rangle$. Напишите программу для вычисления методом Монте-Карло $\langle x_N \rangle$, $\langle y_N \rangle$ и $\langle \Delta R_N^2 \rangle$ и убедитесь в ее правильности, сравнивая полученные с ее помощью результаты с точными. Рассмотрите случай, когда все четыре направления равновероятны.

б. Вычислите методом Монте-Карло R_N для $N = 8, 16, 32$ и 64 с приемлемым числом испытаний для каждого значения N . Заметим, что, вообще говоря, оцененные средние значения $\langle x_N \rangle$ и $\langle y_N \rangle$ не будут в точности равны нулю и наилучшие результаты получатся путем непосредственного вычисления R_N исходя из определений (11.12) и (11.13). Из графика зависимости R_N от N , построенного в дваждылогарифмическом масштабе, оцените показатель степени ν . Если $\nu \approx 1/2$, то оцените величину коэффициента самодиффузии D , воспользовавшись определением (11.6) с заменой «времени» t на N .

*в. Вычислите R_N методом Монте-Карло на треугольной решетке и оцените ν . Можете ли вы сделать вывод о том, что ν не зависит от симметрии решетки? Зависит ли от симметрии решетки коэффициент D ? Если да, то качественно объясните эту зависимость.

*г. Как соответствующим образом обобщить величины $\langle R_N^2 \rangle$, $\langle \Delta R_N^2 \rangle$ и R_N на случай трехмерной решетки? Вычислите R_N для простой кубической решетки ($z = 6$) и оцените ν . Зависит ли показатель ν от размерности пространства?

*д. Рассмотрите случайного пешехода на квадратной решетке, начинающего движение из узла, расположенного на расстоянии h над го-

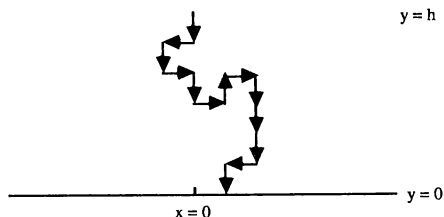


Рис. 11.4. Пример случайного «падения» дождевой капли с высоты h над поверхностью Земли.

ризонтической прямой (рис. 11.4). Примите, что вероятность p_{\downarrow} шага «вниз» больше вероятности p_{\uparrow} шага «вверх». Поскольку $p_{\downarrow} > p_{\uparrow}$, то можно ожидать, что после некоторого числа шагов пешеход «поглотится» (свалится в ров) в некотором узле горизонтальной прямой. Вероятности скачков целесообразно выбрать равными $p_{\downarrow} = 0.5$, $p_{\uparrow} = 0.1$ и $p_{\rightarrow} = p_{\leftarrow} = 0.2$. Проведите моделирование методом Монте-Карло и определите среднее время τ , за которое пешеход достигает горизонтальной прямой, и функциональную зависимость τ от h . Можно ли определить скорость движения в вертикальном направлении? Поскольку пешеход движется не только по вертикали, у него будет некоторое горизонтальное смещение x . Как зависит $\langle \Delta x^2 \rangle$ от h и τ ? Такое случайное блуждание можно рассматривать в качестве идеализированной модели падения дождевой капли при наличии случайных порывов легкого ветра.

ЗАДАЧА 11.6. Непрерывное случайное блуждание

Одна из первых непрерывных моделей случайных блужданий предложена Рейли (1919). В модели Рейли длина каждого шага a является случайной величиной, распределенной с плотностью вероятности $p(a)$, и случайным направлением каждого шага. Для простоты рассмотрим двумерную модель и выберем $p(a)$ так, чтобы каждый шаг был единичной длины. Тогда на каждом шаге пешеход смещается на отрезок единичной длины в случайном направлении. Напишите программу для вычисления методом Монте-Карло $P_N(r)dr$ — вероятности того, что пешеход окажется в интервале от r до $r+dr$, где r обозначает расстояние от начала координат после N шагов. Покажите, что для достаточно больших значений N вычисленную функцию $P_N(r)$ можно аппроксимировать гауссовым распределением. Хорошо ли

гауссово распределение аппроксимирует функцию $P_N(x)$ для малых значений N ? Нужно ли проводить моделирование методом Монте-Карло, чтобы подтвердить справедливость зависимости $R_N \sim N^{1/2}$, или вы можете привести простое объяснение этой зависимости, основанное на формуле для P_N ?

*ЗАДАЧА 11.7. Персистентное случайное блуждание

В «персистентном» случайном блуждании вероятность *перехода*, или «скачка», зависит от последнего перехода. Рассмотрите одномерное случайное блуждание, в котором шаги совершаются только в ближайшие соседние узлы. Предположим, что сделано $N - 1$ шагов. Далее N -й шаг делается в том же направлении с вероятностью α ; шаг в противоположном направлении делается с вероятностью $1 - \alpha$. Физический пример такого блуждания — рассеяние частицы на неподвижных рассеивающих центрах с анизотропными сечениями рассеяния. Другой пример персистентного блуждания рассматривается в п. «в». Модифицируйте программу `random_walk` так, чтобы методом Монте-Карло вычислялись $\langle \Delta x_N^2 \rangle$ и $P_N(x)$. Отметим, что необходимо задавать как начальную координату, так и начальное состояние пешехода. К чему стремится персистентное случайное блуждание при $\alpha = 1/2$?

а. Рассмотрите два случая: $\alpha = 0.25$ и $\alpha = 0.75$ и вычислите величину $\langle \Delta x_N^2 \rangle$ для значений $N = 8, 64, 256$ и 512 . Из асимптотической зависимости $\langle \Delta x_N^2 \rangle$ от N оцените значение ν . Зависит ли ν от α ? Если $\nu \approx 1/2$, вычислите коэффициент самодиффузии D для значений $\alpha = 0.25$ и $\alpha = 0.75$. Дайте физическое объяснение того, почему $D(\alpha \neq 0.5)$ больше (меньше), чем $D(\alpha = 0.5)$.

б. Вычислите функцию $P_N(x)$ для $\alpha = 0.25$ и значений $N = 2, 4, 6$ и 10 . Определите «скорость распространения» c посредством соотношения $P_N(x) = 0$ для $x > cN$. Оцените c , используя графики функций $P_N(x)$ для различных значений N . Сходятся ли полученные оценки c к предельному значению? Проведите аналогичное исследование для обычного случайного блуждания. Сходятся ли оценки c к предельному значению в этом случае?

в. Персистентное случайное блуждание можно рассматривать также как пример блуждания со *многими состояниями*, в котором состояние блуждания определяется последним переходом. Рассмотрим, напри-

мер, пешехода, который может быть в одном из двух состояний; на каждом шаге вероятности остаться в том же состоянии или перейти в другое равны соответственно α и $1 - \alpha$. Одним из первых применений случайного блуждания с двумя состояниями было изучение диффузии в хроматографической колонке. Предположим, что молекула в такой колонке может быть либо в подвижной фазе (постоянная скорость w), либо в захваченной фазе (нулевая скорость). Рассматриваемой в эксперименте величиной является вероятность $P_N(x)$ того, что за N шагов молекула пройдет расстояние x . Положите $w = 1$ и $\alpha = 0.75$ и рассчитайте качественное поведение $P_N(x)$. Хотя молекула не может диффундировать в любое состояние, возможно ли определить понятие ее эффективного коэффициента диффузии?

Падение дождевой капли, рассмотренное в задаче 11.5д, представляет пример ограниченного случайного блуждания, т.е. блуждания при наличии границы. В следующей задаче мы рассмотрим влияние различных типов ограничений, иначе границ, на случайное блуждание в более общей постановке. Другой пример ограниченного случайного блуждания приводится в задаче 11.16.

*ЗАДАЧА 11.8. Ограниченные случайные блуждания

а. Рассмотрите одномерную решетку, у которой имеются «поглощающие» узлы (ловушки) в точках $x = 0$ и $x = a$ ($a > 0$). Пешеход начинает движение из точки x_0 ($0 < x_0 < a$) и с равной вероятностью переходит в ближайшие соседние узлы. Проведите моделирование методом Монте-Карло и убедитесь в том, что среднее время первого прохода τ частицы до ее поглощения выражается формулой

$$\tau = (2D)^{-1}x_0(a - x_0). \quad (11.15)$$

Здесь D обозначает коэффициент самодиффузии в отсутствие ловушек и усреднение производится по всем возможным блужданиям.

б. Модели случайного блуждания при наличии ловушек сыграли важную роль в физике конденсированного состояния. Например, рассмотрим следующую идеализированную модель переноса энергии в твердом теле. Твердое тело представляется в виде решетки с двумя типами узлов: узлы-хозяева и узлы-ловушки. Падающий фотон поглощается узлом-хозяином и возбуждает молекулу. Энергия возбужде-

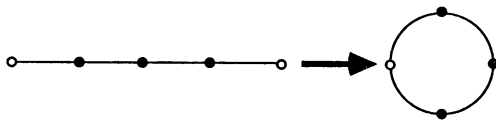


Рис. 11.5. Эквивалентность регулярной решетки с узлами-ловушками, расположенными с периодом 4, и окружности с $N = 3$ узлами-хозяевами и одним узлом-ловушкой. Узлы-ловушки обозначены светлыми кружками.

ния, или *экситон*, случайным образом передается ближайшим соседям, и возбужденная молекула возвращается в основное состояние. Экситон блуждает по решетке до тех пор, пока не достигнет узла-ловушки. Тогда экситон поглощается и происходит некоторый физический процесс, например химическая реакция.

Одним из вариантов данной модели переноса энергии является одномерная решетка с узлами-ловушками, расположенными в периодической подрешетке. Поскольку эти узлы-ловушки расположены на одинаковом расстоянии друг от друга, то можно заменить случайное блуждание на бесконечной решетке случайным блужданием по окружности (рис. 11.5). Рассмотрите окружность с N узлами-хозяевами, или непоглощающими узлами, и одним узлом-ловушкой. Если пешеход с одинаковой вероятностью начинает движение из любого непоглощающего узла и с одинаковой вероятностью перепрыгивает в ближайшие соседние узлы, то как зависит от N среднее время жизни τ (среднее число шагов до попадания в узел-ловушку)? Не моделируйте эту ситуацию, а воспользуйтесь результатами п. «а».

в. Знаменитой версией задачи о времени первого прохода является задача о «разорении игрока». Предположим, что два человека начинают игру, имея каждый начальный капитал в 10 долларов, и после каждого броска игральной кости один из игроков должен выиграть 1 доллар, другой должен проиграть 1 доллар. Как долго в среднем смогут они играть до тех пор, пока капитал одного из них не исчерпается? Как долго смогут они играть, если каждый сядет за стол со 100 долларами? (К сожалению, в этой игре в отличие от реальной жизни ни одному из игроков нельзя залезать в долги.)

г. Кристаллическое твердое тело никогда не является совершенным, а содержит разнообразные *дефекты*. Простейшим дефектом является вакансия решетки, например отсутствие атома в узле решетки и по-

мещение дополнительного атома на поверхность. При конечной температуре в реальном кристалле всегда имеется некоторое число решеточных вакансий. Во многих случаях вакансия диффундирует, меняясь местами с соседними атомами случайным образом. Предположите, что при $t = 0$ вакансия расположена в центре окружности радиусом r . Проведите моделирование методом Монте-Карло и определите среднее время, за которое вакансия достигает поверхности металла, находящейся на расстоянии r . Каково распределение вероятностей для времени первого прохода?

д. Рассмотрите одномерную решетку с отражающими узлами, расположенными в точках $x = -a$ и $x = a$. Другими словами, если пешеход попадает в отражающий узел $x = a$, то на следующем шаге он оказывается в узле с координатой $x = a - 1$. В момент времени $t = 0$ пешеход начинает движение из узла $x = 0$ и с равной вероятностью переходит в ближайшие соседние узлы. Составьте программу вычисления методом Монте-Карло $P_N(x)$ — вероятности того, что через N шагов пешеход окажется в узле x . Сравните вид функции $P_N(x)$ при наличии и в отсутствие отражающих «стенок». Удастся ли вам различить эти два распределения вероятностей, если N порядка a ? При каком минимальном значении N вы можете различить эти распределения?

Хотя во всех вышеприведенных задачах фигурируют случайные блуждания на решетке, хранить координаты узлов решетки или путь пешехода не было необходимости. В следующей задаче мы рассмотрим модель случайного блуждания, в которой требуется хранить координаты решетки, образованной «газом» (коллективом) случайных пешеходов.

*ЗАДАЧА 11.9. Диффузия частицы в решеточном газе

Рассмотрим случай ненулевой концентрации c случайных пешеходов (частиц) на квадратной решетке. Каждая частица перемещается случайным образом в ближайший *свободный* узел, и попадание двух частиц в один узел исключено, иначе говоря, частицы не взаимодействуют. Такая модель является примером *решеточного газа*. Заметим, что движение отдельной частицы коррелировано с движением других частиц. Физическое обоснование этой модели возникает в физике металлов, где диффузия обусловлена температурными вакансиями, концентрация которых зависит от температуры. Основной рассматри-

ваемой физической величиной является коэффициент самодиффузии D отдельной (меченой или индикаторной) частицы. Алгоритм Монте-Карло для вычисления D можно сформулировать следующим образом:

1. Распределите частицы с концентрацией c случайным образом по узлам решетки. (Не забудьте, что $0 < c \leq 1$.) Пометьте каждую частицу (т.е. сделайте ее непохожей на другие) и запомните ее начальное положение в массиве.

2. На каждом шаге случайным образом выберите частицу и один из соседних с ней узлов. Если соседний узел не занят, то частица переходит в него; в противном случае она сохраняет свое текущее положение.

В этом случае «время» измеряется в произвольных единицах. Принятая единица времени, которой мы будем часто пользоваться, соответствует одному шагу метода Монте-Карло на частицу. За один шаг метода Монте-Карло на частицу каждая частица совершает в среднем один переход. Коэффициент диффузии D получается предельным переходом $t \rightarrow \infty$ функции $D(t)$, где $D(t)$ определяется выражением

$$D(t) = \frac{1}{2dt} \langle \Delta R(t)^2 \rangle \quad (11.16)$$

и $\langle \Delta R(t)^2 \rangle$ — полное среднеквадратичное смещение, приходящееся на одну меченую частицу за время t . Приведем пример программы, в которой реализован указанный алгоритм.

```
PROGRAM lattice_gas
DIM site(30,30)
DIM x(200), y(200), xinitial(200), yinitial(200)
CALL initial(L,N,nmcs)
CALL lattice(L,N,site,x,y,xinitial,yinitial)
FOR t = 1 to nmcs
  CALL move(L,N,site,x,y,xinitial,yinitial)
  CALL data(L,N,t,x,y,xinitial,yinitial)
NEXT t
END
```

```

SUB initial (L,N,nmcs)
  RANDOMIZE
  LET L = 30                                ! размер решетки
  INPUT prompt "число частиц = ";N          ! число частиц
  PRINT "концентрация = "; N/(L*L)
  LET nmcs = 10                             ! число шагов Монте-Карло на частицу
  PRINT "время t", "R2(t)"
END SUB

SUB lattice (L,N,site(,),x(,),y(,),xinitial(,),yinitial(,))
  DO
    LET xadd = int(L*rnd + 1)
    LET yadd = int(L*rnd + 1)
    IF site(xadd,yadd) = 0 then
      LET i = i + 1                          ! число добавляемых частиц
      LET site(xadd,yadd) = -1              ! узел занят
      LET xinitial(i),x(i) = xadd           ! координата x при t = 0
      LET yinitial(i),y(i) = yadd
    END IF
  LOOP until i = N
END SUB

SUB move (L,N,site(,),x(,),y(,),xinitial(,),yinitial(,))
  FOR imove = 1 to N
    LET itrial = int(N*rnd + 1)
    LET xtrial = x(itrial)
    LET ytrial = y(itrial)
    CALL random(xtrial,ytrial,L)
    IF site(xtrial,ytrial) = 0 then          ! узел свободен
      LET site(x(itrial),y(itrial)) = 0
      LET x(itrial) = xtrial
      LET y(itrial) = ytrial
      LET site(x(itrial),y(itrial)) = -1    ! занят новый узел
    END IF
  NEXT imove
END SUB

```

```

SUB random (xtemp,ytemp,L)
! выбор случайного направления с учетом периодических краевых условий
LET direction = int(4*rnd + 1)
SELECT CASE direction
CASE 1
LET xtemp = xtemp + 1
IF xtemp > L then LET xtemp = xtemp - L
CASE 2
LET xtemp = xtemp - 1
IF xtemp < 1 then LET xtemp = L - xtemp
CASE 3
LET ytemp = ytemp + 1
IF ytemp > L then LET ytemp = ytemp - L
CASE 4
LET ytemp = ytemp - 1
IF ytemp < 1 then LET ytemp = L - ytemp
END SELECT
END SUB

SUB separation (dx,dγ,L)
LET L2 = 0.5*L ! учет в расстоянии периодических краевых условий
IF abs(dx) > L2 then LET dx = dx - sgn(dx)*L
IF abs(dγ) > L2 then LET dγ = dγ - sgn(dγ)*L
END SUB

SUB data (L,N,t,x(),y(),xinitial(),yinitial())
FOR i = 1 to N
LET dx = x(i) - xinitial(i)
LET dγ = y(i) - yinitial(i)
CALL separation(dx,dγ,L)
LET R2 = R2 + dx*dx + dγ*dγ
NEXT i
LET R2cum = R2cum + R2
LET R2bar = R2cum/N
PRINT t,R2bar
END SUB

```

а. Вычислите методом Монте-Карло величину D на квадратной решетке для значений $c = 0.1, 0.2, 0.3, 0.5$ и 0.7 . Хотя D определяется как предел при $t \rightarrow \infty$ выражения (11.16), на практике $D(t)$

флуктуирует со временем и путем увеличения t точность не повышается. Лучшую статистику для D можно получить, усредняя D по как можно большему числу меченых частиц и, следовательно, рассматривая решетку как можно большего размера L . Почему необходимо ограничивать число шагов в методе Монте-Карло так, чтобы величина $\langle \Delta R(t)^2 \rangle$ была меньше, чем $(L/2)^2$? Точность вычисления D можно также увеличить, усредняя $\langle \Delta R(t)^2 \rangle$ по различным стартовым временам. Покажите, что отклонение $D(t)$ от точного значения обратно пропорционально квадратному корню из числа частиц, по которым производится усреднение в (11.16).

б. Почему D является монотонно убывающей функцией концентрации c ? Определите зависимость от концентрации вероятности того, что если частица переходит в вакансию в момент времени t , то она возвращается в начальное положение в момент времени $t+1$. Попробуйте определить качественную зависимость между величиной D как функцией концентрации и этой вероятностью.

в. Рассмотрите модель одномерной решетки, на которой частицы движутся случайным образом, но исключается попадание двух частиц в один узел. Последнее ограничение означает, что частицы не могут проходить сквозь друг друга. Вычислите $\langle \Delta x^2 \rangle$ в зависимости от t . Диффундируют ли эти частицы, т.е. пропорциональна ли величина $\langle \Delta x^2 \rangle$ времени t ? Если нет, то как зависит $\langle \Delta x^2 \rangle$ от t ?

ЗАДАЧА 11.10. Центральная предельная теорема

Рассмотрим случайную величину x с плотностью вероятности $f(x)$. Момент m -го порядка функции $f(x)$ определяется как

$$\langle x^m \rangle = \int dx x^m f(x). \quad (11.17)$$

Среднее значение $\langle x \rangle$ равно значению выражения (11.17) при $m = 1$. Дисперсия σ_x^2 функции $f(x)$ определяется как

$$\sigma_x^2 = \langle x^2 \rangle - \langle x \rangle^2. \quad (11.18)$$

Рассмотрим сумму y_n , соответствующую среднему n значений x :

$$y_n = (x_1 + x_2 + \dots + x_n)/n. \quad (11.19)$$

Обозначим $y = y_n$. Предположим, что мы проделали много измерений y . Известно, что значения y не получатся одинаковыми, а будут распределены в соответствии с некой плотностью вероятности $P(x)$, где $P(x)\Delta y$ — вероятность попадания измеряемой величины в интервал $[y, y + \Delta y]$. Основными рассматриваемыми величинами являются среднее $\langle y \rangle$, стандартное отклонение $\sigma_y^2 = \langle y^2 \rangle - \langle y \rangle^2$ и сама функция $P(y)$.

а. Предположите, что $f(x)$ равномерно распределена на отрезке $[-1, 1]$. Вычислите аналитически $\langle x \rangle$ и σ_x . Используя метод Монте-Карло, проведите достаточное количество измерений y для определения $P(y)$, $\langle y \rangle$ и σ_y с приемлемой точностью. (Например, выберите $n = 500$ и проведите 100 измерений y .) Покажите, что σ_y приблизительно равно σ_x/\sqrt{n} . Постройте график функции $P(y)$ и качественно обсудите его форму. Сильно ли меняется форма графика $P(y)$ с ростом n ? Меняется ли форма графика $P(y)$ при увеличении числа измерений y ?

б. Чтобы проверить, насколько результаты п. «а» носят общий характер, рассмотрите экспоненциальную плотность вероятности

$$f(x) = \begin{cases} \exp(-x), & \text{если } x \geq 0, \\ 0 & , \text{если } x < 0. \end{cases} \quad (11.20)$$

Вычислите аналитически $\langle x \rangle$ и σ_x . Модифицируйте программу метода Монте-Карло из п. «а» и вычислите $P(y)$, $\langle y \rangle$ и σ_y . Связана ли σ_y с σ_x так же, как в п. «а»? Постройте график $P(y)$ и обсудите качественно его форму и зависимость от количества измерений y .

в. Пусть y будет оценкой интеграла

$$4 \int_0^1 dx \sqrt{1 - x^2}, \quad (11.21)$$

полученной методом Монте-Карло (см. задачу 10.5а.) В этом случае y находится путем получения выборки подынтегральной функции $f(x) = 4\sqrt{1 - x^2}$ в n точках. Выберите $n = 1000$ и проведите 100 измерений y . Покажите, что распределение значений y является гауссовым. Как связано стандартное отклонение $P(y)$ со стандартным отклонением $f(x)$?

г. Рассмотрите лоренцову плотность вероятности

$$f(x) = \frac{a}{\pi} \frac{1}{x^2 + a^2} \quad (11.22)$$

с $a = 1$. Чему равно среднее значение $\langle x \rangle$? Существует ли второй момент, а тем самым и дисперсия $f(x)$? Вычислите методом Монте-Карло $P(y)$, $\langle y \rangle$ и σ_y . Постройте график функции $P(y)$ и обсудите его вид. Как зависит функция $P(y)$ от числа испытаний?

ЗАДАЧА 11.11. Случайные блуждания с переменным шагом

а. Рассмотрите одномерное случайное блуждание со всеми допустимыми длинами прыжков. Вероятность того, что длина шага равна j , обозначим $p(j)$. Если $p(j)$ имеет вид $p(j) = e^{-j}$, то как выглядит $P_N(x)$? Указания: для формирования шагов с длиной, отвечающей плотности вероятности $p(j)$, примените метод обратного преобразования из разд. 10.7. Затем сгенерируйте N -шаговое случайное блуждание и найдите полное смещение x . Сгенерируйте много таких блужданий и определите $P_N(x)$. Постройте график $P_N(x)$ и убедитесь в том, что его форма соответствует гауссову распределению. Эквивалентно ли такое случайное блуждание процессу диффузии?

б. Рассмотрите случайное блуждание на решетке, как в п. «а», с вероятностью распределения длины шагов $p(j) = A/j^2$. Определите нормировочную постоянную A , используя соотношение

$$\sum_{j=1}^{\infty} \frac{1}{j^2} = \frac{\pi^2}{6} \quad (11.23)$$

и требуя, чтобы $p(j)$ была нормирована на единицу. Существует ли второй момент $p(j)$? Ожидаете ли вы, что вероятность $P_N(x)$ будет гауссовой? Проведите моделирование методом Монте-Карло, как в п. «а», и убедитесь в том, что функция $P_N(x)$ имеет вид

$$P_N(x) \sim \frac{bN}{x^2 + b^2 N^2}. \quad (11.24)$$

Чему равно значение постоянной b ? Существует ли у функции $P_N(x)$ дисперсия? Эквивалентно ли такое случайное блуждание процессу диффузии?

В п.п. «а» — «г» задачи 11.10 приведены примеры *центральной предельной теоремы*, которая утверждает, что распределение вероятности большого числа измерений y будет гауссовым со средним $\langle y \rangle$ и стандартным отклонением, равным стандартному отклонению $f(x)$, умноженному на $1/\sqrt{n}$. Единственные налагаемые требования состоят в том, чтобы первый и второй моменты функции $f(x)$ были конечны, чтобы измерения y были статистически независимыми и n было велико. Используя центральную предельную теорему, объясните результаты, полученные в задачах 11.10 и 11.11а. Как связаны между собой настоящее вычисление функции $P_N(x)$ и вычисление распределений вероятностей в рассмотренных ранее моделях случайного блуждания?

ЗАДАЧА 11.12. Распределение Гаусса

Рассмотрите сумму вида

$$y = \sum_{i=1}^{12} r_i \quad (11.25)$$

где r_i равномерно распределены на отрезке $[0, 1]$. Прodelайте много «измерений» величины y и покажите, что распределение вероятности y аппроксимирует гауссово распределение со средним, равным 6, и дисперсией 1. Обсудите возможность использования этого результата для генерирования гауссова распределения с произвольными средним и дисперсией.

11.4. ПРИЛОЖЕНИЯ В ФИЗИКЕ ПОЛИМЕРОВ

Немного найдется областей исследования, в которых модели случайного блуждания играют столь же важную роль, как в физике полимеров (де Жен). Самое первое исследование (1934) конфигураций полимеров было проведено с использованием моделей случайных блужданий. На протяжении последних 50 лет изучение статистических свойств длинных гибких полимерных цепочек и моделей случайных блужданий развивалось параллельно.

Полимер состоит из очень большого числа N ($N \sim 10^3 - 10^5$) повторяющихся звеньев, или мономеров. Например, повторяющиеся звенья цепочки полиэтилена можно изобразить как $\dots -CH_2-CH_2-CH_2-\dots$. Зна-

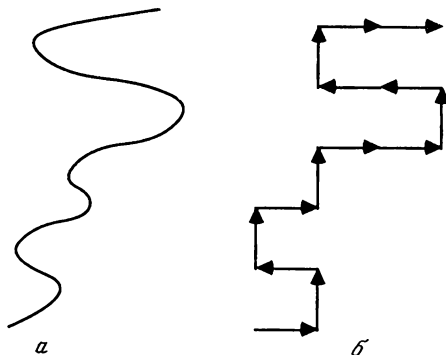


Рис 11.6. *а* — схематическое представление линейного полимера, помещенного в хороший растворитель. *б* — пример соответствующего блуждания на квадратной решетке (*б*).

ние детальной структуры полимера важно для многих практических приложений. Если, например, мы хотим усовершенствовать производство резины, то нужно хорошо разбираться в свойствах локальных движений цепочек каучука. Однако если нас интересуют *глобальные* свойства полимера, то детали строения цепочки можно не принимать во внимание.

Рассмотрим известный пример о кусочке спагетти, помещенном в теплую воду. Через короткий промежуток времени спагетти становится гибким и ведет себя подобно полимерной цепочке в хорошем растворителе, т.е. спагетти как не сворачивается в маленький комок, так и не вытягивается полностью. Скорее кусочек приобретает случайную структуру, как схематично показано на рис. 11.6. Мы говорим, что спагетти ведет себя как слабый раствор полимерных цепочек в хорошем растворителе. Разбавленность раствора означает, что мы можем не принимать во внимание запутывание цепочек и рассматривать каждую цепочку отдельно. Наличие «хорошего» растворителя означает, что полимеры могут свободно двигаться и, следовательно, принимать всевозможные конфигурации. Одной из фундаментальных геометрических характеристик, которую можно использовать для описания полимера, является среднеквадратичное расстояние между концами цепочки R_N , где R_N определяется формулой (11.13) и N — количество мономеров. Известно, что для слабого раствора полимерных цепочек в хорошем растворителе асимптотическая зависимость R_N выражается формулой (11.14) с показателем степени $\nu \approx 3/5$ для *всех* гибких цепочек. Для двумерного случая известен точный результат $\nu = 3/4$. Множитель a в выражении

(11.14) определяется структурой мономеров и самим растворителем.

Обсудим теперь модель случайного блуждания, которая описывает глобальные свойства линейных полимеров в растворе. Мы уже рассмотрели модель полимерной цепочки, состоящей из прямолинейных отрезков одинаковой длины, соединенных под случайными углами (см. задачу 11.6). Дальнейшая идеализация заключается в размещении полимерной цепочки на решетке (рис. 11.6,б). Известно, что если пренебречь взаимодействием мономеров между собой, то независимо от размерности и структуры решетки получается гауссова цепочка с $\nu = 1/2$. Поскольку этот результат для ν не соответствует экспериментальным данным, необходимо рассмотреть более реалистичную модель цепочки, в которой учитывается самое важное физическое свойство любого полимера — два мономера не могут находиться в одном месте пространства. Данное ограничение, известное как условие «исключенного объема», означает, что само блуждание невозможно адекватно описать с помощью только модели случайного блуждания. Стандартная решеточная модель гибкой полимерной цепочки называется блужданием «без самопересечений» (ББС). Рассмотрите множество всех N -шаговых блужданий, исходящих из определенной начальной точки, при условии глобального ограничения, что в каждом блуждании никакой узел решетки не может посещаться более одного раза; данное ограничение отвечает условию исключенного объема. Расчет характеристик ББС — трудная задача, и вычисления на компьютере комбинаторным методом и методом Монте-Карло сыграли важную роль в получении современных представлений об этих явлениях. В задаче 11.13 мы рассмотрим двумерное моделирование ББС методом Монте-Карло.

ЗАДАЧА 11.13. Моделирование двумерного ББС методом Монте-Карло

а. Рассмотрите ББС на квадратной решетке. Удобно представить решетку в виде двумерного массива; этот массив используется для записи узлов, которые посещались. Выберите произвольный узел в качестве начального и предположите, что первый шаг делается на «север», поскольку блуждания, порождаемые тремя другими возможными начальными шагами, различаются только направлением. Заметим, что второй шаг можно сделать в трех возможных направлениях. Для получения несмещенных результатов мы генерируем случайное число (1, 2 или 3), которое и определяет выбор одного из трех направлений движения. Последующие шаги генерируются аналогичным образом. К сожалению, обычно при таком способе моделирования

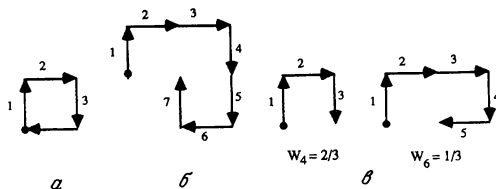


Рис. 11.7. Примеры блужданий без самопересечений на квадратной решетке. Начальная точка обозначена черным кружком. *а*—запрещенное $N = 4$ -шаговое блуждание. *б*—такое $N = 8$ -шаговое блуждание приводит к самопересечению, и его вес равен нулю. *в*—примеры весов блужданий в улучшенном методе Розенблата.

блуждание не продолжается бесконечно. Рассмотрите $N = 3$ -шаговое блуждание, показанное на рис. 11.7, *а*. Следующий шаг, показанный на рисунке, приводит к самопересечению и тем самым нарушает принятое ограничение. Для получения несмещенных результатов нам нужно, как обычно, сгенерировать случайное число (1, 2, 3). Если следующий шаг приводит к самопересечению, то блуждание прекращается и начинается новое блуждание из начальной точки. Напишите программу, в которой реализуется этот алгоритм, и зарегистрируйте долю $f(N)$ успешных попыток построения полимерных цепочек, составленных из всех N звеньев (шагов). Какова качественная зависимость $f(N)$ от N ? Какое максимальное значение N вы в состоянии еще рассмотреть? Вычислите R_N для нескольких значений N .

б. Недостатком приведенного выше метода моделирования является его неэффективность для длинных цепочек, т.е. доля успешных попыток уменьшается экспоненциально. Для преодоления этого недостатка было предложено несколько «улучшений» метода. Обсудим сначала относительно простую процедуру, предложенную М. Розенблатом и А. Розенблатом (1955), в которой каждому N -шаговому блужданию соответствует весовая функция W_N . Поскольку первый шаг на север всегда возможен, мы имеем $W_1 = 1$. Чтобы в равной степени учесть все допустимые при данном N конфигурации, веса W_N для $N > 1$ определяются в соответствии со следующими возможностями:

1. Все три допустимых шага нарушают условие самонепересечения (см., например, рис. 11.7, *б*). Блуждание прекращается с весом $W_N = 0$, и генерируется новое блуждание из начальной точки.
2. Все три шага допустимы, и $W_N = W_{N-1}$.

3. Допустимы только m шагов с $1 \leq m < 3$ (рис. 11.7, в). Тогда $W_N = (m/3)W_{N-1}$ и случайное число используется для выбора одного из m допустимых шагов.

Правильное значение $\langle R_N^2 \rangle$ получается умножением $R_{N,i}^2$ т.е. значения R_N^2 , полученного в i -м испытании, на значение весовой функции W_N , которое мы обозначим $W_{N,i}$. Тогда получим

$$\langle R_N^2 \rangle = \sum_i W_{N,i} R_{N,i}^2 / \sum_i W_{N,i}, \quad (11.26)$$

где суммирование производится по всем испытаниям. Включите процедуру Розенблата в свою программу метода Монте-Карло и вычислите R_N для значений $N = 4, 8, 16$ и 32 . Оцените показатель степени ν из графика зависимости R_N от N , построенного в дваждылогарифмическом масштабе. Отличается ли найденная оценка ν , от значения, полученного из случайного блуждания $\nu = 1/2$?

*ЗАДАЧА 11.14. Применение метода рептаций¹⁾

Одним из наиболее эффективных улучшений метода является процедура «рептаций» (см. список литературы). Для простоты рассмотрим модель полимерной цепочки, у которой все углы связи равны 90° . В качестве примера такой модели на рис. 11.8 показаны пять $N=5$ -звенных независимых полимерных цепочек. (Другие цепочки совпадают с приведенными с точностью до преобразований поворота или отражения.) Метод рептаций можно сформулировать следующим образом:

1. Случайным образом выбираем цепочку и удаляем хвостовое звено.
2. Пробуем добавить звено в начало цепочки. Возможны не более двух направлений присоединения нового головного звена.
3. Если добавление нового звена приводит к нарушению условия самонепересечения, возвращаемся к исходной цепочке и меняем местами голову и хвост. Включаем эту цепочку в статистическую выборку.

¹⁾От латинского reptatio — ползающий, пресмыкающийся. — Прим. перев.

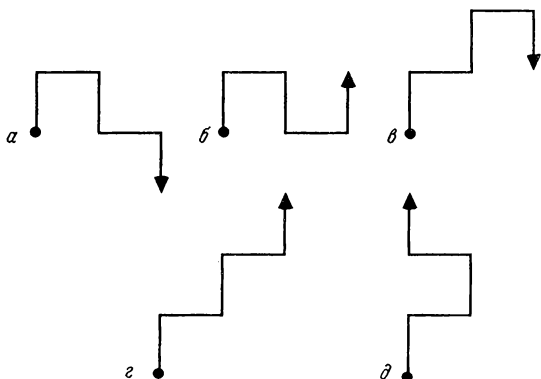


Рис. 11.8. Пять независимых возможных блужданий из $N = 5$ шагов на квадратной решетке с углами связи $\pi/2$. Голова и хвост каждого блуждания обозначаются стрелкой и кружком соответственно.

Для получения статистического среднего R_N^2 приведенные выше шаги повторяем много раз.

В качестве примера использования метода рептации предположим, что мы выбрали цепочку a на рис. 11.8. Новое звено можно добавить в двух направлениях (рис. 11.9, a), так что в среднем получается $a \rightarrow \frac{1}{2}b + \frac{1}{2}г$. По сравнению с этим в цепочке $б$ звено можно добавить только в одном направлении, и получается $б \rightarrow \frac{1}{2}д + \frac{1}{2}б$, где хвост и голова цепочки $б$ меняются местами (рис. 11.9, $б$). Убедитесь в том, что остальные цепочки преобразуются по схемам

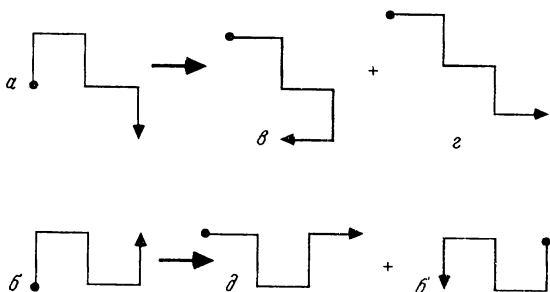
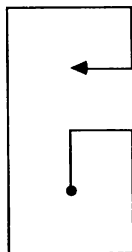


Рис. 11.9. Допустимые преобразования цепочек a и $б$. Одно из возможных преобразований цепочки $б$ нарушает условие самонепересечения, и таким образом вместо этого меняются местами голова и хвост.



$a \rightarrow \frac{1}{2}d + \frac{1}{2}a$, $g \rightarrow \frac{1}{2}a + \frac{1}{2}g$ и $d \rightarrow \frac{1}{2}a + \frac{1}{2}b$ и что в результате все пять цепочек равновероятны, т.е. преобразования, применяемые в методе рептаций, сохраняют собственные статистические веса цепочек. Существует лишь одна трудность: если не начать с конфигурации, имеющей «тупики» на обоих концах, как, например, показанной на рис. 11.10, то такую конфигурацию мы никогда не получим, используя описанное выше преобразование. Отсюда видно, что метод рептаций приводит к небольшому смещению нашей статистической выборки и вычисленное среднее расстояние между концами цепочки будет получаться немного больше, чем если бы рассматривались все конфигурации. Однако вероятность таких «тупиковых» конфигураций очень мала и смещением в большинстве случаев можно пренебречь.

Считая углы между звеньями равными 90° , вычислите вручную точное значение $\langle R_N^2 \rangle$ для $N = 5$. Затем составьте программу метода Монте-Карло, в которой реализуется метод рептаций. Сгенерируйте одно $N = 5$ -шаговое блуждание и с помощью метода рептаций сгенерируйте статистическую выборку цепочек. Для проверки своей программы Монте-Карло рассчитайте $\langle R_N^2 \rangle$ и сравните оба полученных значения. Затем распространите свои расчеты $\langle R_N^2 \rangle$ методом Монте-Карло на большие значения N .

Моделью, которую легче изучать и которая описывает статистику линейных полимеров специального типа в растворе, является «истинное» блуждание без самопересечений (ИББС). ИББС описывает путь случайного пешехода, которому предписано не посещать любой узел решетки с вероятностью, являющейся функцией от количества уже происшедших посещений данного узла. Это условие приводит к уменьшению иск-

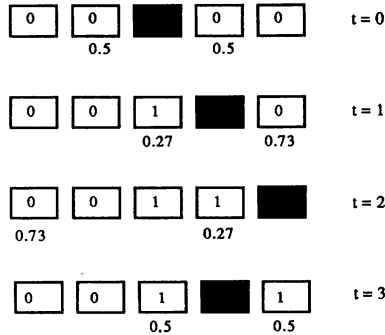


Рис. 11.11. Пример временной эволюции истинного блуждания без самопересечений с $g = 1$. Зачерненные ячейки изображают положения пешехода в момент времени t . Числа, приведенные внутри каждой ячейки, указывают количество их посещений, а числа под ними — вероятность перехода в соседнюю ячейку.

люченного объема по сравнению с обычным блужданием без самопересечения. ИББС рассматривается в задаче 11.15.

ЗАДАЧА 11.15. Одномерное «истинное» блуждание без самопересечений

Одномерное ИББС соответствует пешеходу, который может «перескакивать» в один из двух соседних узлов с вероятностью, зависящей от числа уже происшедших посещений этих узлов. Предположим, что в момент времени t пешеход находится в i -м узле. Он уже посетил $(i+1)$ -й узел n_{i+1} раз и $(i-1)$ -й узел n_{i-1} раз. Тогда вероятность того, что на шаге $t+1$ пешеход перепрыгнет в $(i+1)$ -й узел, равна

$$P_{i+1} = \frac{\exp(-gn_{i+1})}{\exp(-gn_{i+1}) + \exp(-gn_{i-1})}. \quad (11.27)$$

Вероятность скачка в $(i-1)$ -й узел составляет $P_{i-1} = 1 - P_{i+1}$. Параметр g представляет меру «желательности» избежать данного пути. На рис. 11.11 показаны несколько первых шагов типичного ИББС. Основной рассматриваемой величиной является показатель степени ν . Нам известно, что $g = 0$ соответствует обычному случайному блужданию с $\nu = 1/2$, а предельное значение $g \rightarrow \infty$ соответствует блужданию без самопересечения. Каково значение ν для

одномерного блуждания без самопересечения? Отличается ли это значение ν для любого конечного g от указанных двух предельных случаев? Ниже мы исследуем эти вопросы.

а. Напишите программу моделирования методом Монте-Карло одномерного ИББС. Предусмотрите запоминание числа посещений каждого узла. На каждом шаге вычислите вероятность P скачка вправо. Сгенерируйте случайное число r и сравните его с P . Если $r \leq P$, то сдвиньте пешехода вправо, в противном случае — влево. Используйте одну и ту же последовательность случайных чисел для сравнения путей ИББС при $g = 0.1$ и обычного случайного блуждания ($g = 0$).

б. Вычислите $\langle \Delta x_N^2 \rangle$, где x — расстояние пешехода от начальной точки, как функцию от числа шагов N . Постройте в дваждылогарифмическом масштабе график зависимости $\langle \Delta x_N^2 \rangle$ от N и оцените показатель степени ν . Отличается ли показатель степени ν от соответствующих значений ν для случайного блуждания и блуждания без самопересечений? Приемлемыми значениями параметров являются $g = 0.1$ и $N = 1000$ шагов. Для получения качественных результатов вычисляются средние по 100 испытаниям. Для сравнения можно обратиться к работе Бернаскони и Пьетронеро, где приведены результаты вычислений для $N = 10\,000$ шагов и 1000 испытаний; уточненные результаты для $g = 2$ получены для 200 000 шагов и 10 000 испытаний.

Во многих из рассмотренных выше задач обнаруживается медленная сходимость метода Монте-Карло и возникают трудности получения количественных результатов для асимптотических величин, таких, как показатель степени ν . Мы завершаем этот раздел определенным предостережением с рассмотрением «простой» задачи, в которой обычные методы Монте-Карло дают неверные асимптотические результаты.

***ЗАДАЧА 11.16. Случайное блуждание на решетке, содержащей узлы-ловушки**

а. Мы уже рассмотрели среднее время жизни пешехода в одномерной задаче случайного блуждания при наличии периодически распределенных узлов-ловушек (см. задачу 11.8а). Теперь предположим, что узлы-ловушки распределены на одномерной решетке случайным образом с концентрацией c . Если пешеход помещен случайным образом в узел, не являющийся ловушкой, то определите среднее время его

жизни τ , т.е. среднее число шагов до попадания в узел-ловушку. Положите, что на каждом шаге пешеход с равной вероятностью переходит в ближайшие соседние узлы, и используйте периодические краевые условия. Данная задача гораздо труднее, чем может показаться на первый взгляд, а в действительности вас подстерегает ряд тупиков! Главная проблема заключается в том, что необходимо выполнить *три* усреднения: по распределению ловушек, по начальной координате пешехода и по всем различным возможным блужданиям для данной начальной координаты и распределения узлов-ловушек. Выберите подходящие значения для числа испытаний, связанных с каждым усреднением, и найдите среднее время жизни τ из моделирования методом Монте-Карло. Если τ изменяется по степенному закону, например $\tau \approx \tau_0 c^{-z}$, то оцените показатель степени z . Предложите простое объяснение такого значения z .

б. По-видимому, прямым обобщением п. «а» является вычисление методом Монте-Карло вероятности выживания P_n для n -шагового блуждания. Положите $c = 0.5$ и вычислите методом Монте-Карло P_n для максимально возможного значения n . [Имеются опубликованные результаты (Хевлин и др.) для 50 000 испытаний и 2000 шагов на решетке из 50 000 узлов.] Положите, что для $n \gg 1$ формула P_n имеет вид

$$\ln P_n \sim -bn^\alpha \quad (n \rightarrow \infty), \quad (11.28)$$

где b — постоянная, зависящая от концентрации. Согласуются ли ваши результаты с формулой (11.28)? Можно ли получить осмысленную оценку показателя степени α ?

в. Цель п. «б» заключалась в том, чтобы убедить вас, что на практике методом Монте-Карло невозможно получить правильное асимптотическое поведение функции P_n . Проблема связана с тем, что мы пытаемся оценить P_n в асимптотической области, где значения P_n очень малы; присущие методу Монте-Карло флуктуации не позволяют получить осмысленные результаты за разумное время. С помощью аналитических методов было доказано, что асимптотическая зависимость P_n от n действительно имеет вид (11.28), но с показателем степени $\alpha = 1/3$. Соответствуют ли результаты, полученные вами с помощью метода Монте-Карло, этому значению α ?

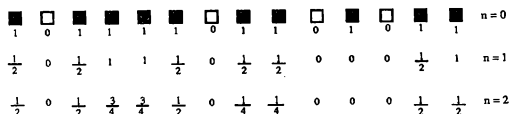


Рис. 11.12. Пример полного перебора блужданий для заданного распределения узлов-ловушек. Зачерненные и светлые квадраты обозначают регулярные узлы и узлы-ловушки соответственно. На шаге $n = 0$ пешеход помещается в каждый регулярный узел. Число под каждым узлом с номером i указывает количество пешеходов W_i . Используются периодические краевые условия. В данном примере начальное число пешеходов составляет $N_0 = 10$. Найдено, что средние значения вероятности выживания на шагах $n = 1$ и $n = 2$ равны 0.6 и 0.45 соответственно.

К счастью, существует более хороший метод, который уменьшает количество усреднений, и поэтому уменьшаются флуктуации. Суть этого метода заключается в точном вычислении вероятности того, что после n шагов пешеход оказывается в i -м узле с учетом данного распределения ловушек. Метод иллюстрируется на рис. 11.12. В первой строке показано случайное распределение узлов-ловушек на одномерной решетке. Во второй строке показано размещение пешехода во всех регулярных узлах решетки; узлам-ловушкам присваивается значение 0. Поскольку каждый пешеход с вероятностью $1/2$ переходит в каждый ближайший узел, количество пешеходов $W_{n+1}(i)$ в i -м узле на $(n+1)$ -м шаге задается выражением

$$W_{n+1}(i) = \frac{1}{2}[W_n(i+1) + W_n(i-1)]. \quad (11.29)$$

[Сравните соотношение (11.29) с соотношением, полученным в задаче 11.1а.] Вероятность выживания P_n после n шагов для заданного распределения узлов-ловушек выражается точной формулой:

$$P_n = \frac{1}{N_0} \sum_i W_n(i), \quad (11.30)$$

где N_0 — начальное количество пешеходов, и суммирование проводится по всем узлам решетки. Напишите программу для вычисления P_n , используя эту процедуру полного перебора, и вычислите среднее $\langle P_n \rangle$ по нескольким распределениям узлов-ловушек. Положите $c = 0.5$ и определите вероятность выживания для $n = 32, 64, 128, 512$ и 1024 . Выберите периодические краевые условия и решетку достаточного размера. С какой точностью можно оценить показатель

степени α ? [Для сравнения: Хевлин и др. рассматривали решетку из 50 000 узлов и значения n вплоть до 10^7 .]

11.5. НЕПРЕРЫВНЫЙ ПРЕДЕЛ

Поучительно рассмотреть непрерывный предел модели одномерного случайного блуждания, изложенной в разд. 11.2. Если с равной вероятностью делается шаг вправо или влево, то случайное блуждание можно переписать в виде простого «порождающего» уравнения

$$P_n(i) = \frac{1}{2} P_{n-1}(i+1) + \frac{1}{2} P_{n-1}(i-1), \quad (11.31)$$

где $P_n(i)$ — вероятность того, что пешеход оказывается в i -м узле после n шагов. Для получения дифференциального уравнения для плотности вероятности $P(x, t)$ положим $t = n\tau$, $x = ia$ и $P_n(i) = aP(x, t)$, где τ — время между шагами, a — период решетки. Эти обозначения позволяют переписать (11.31) в эквивалентной форме:

$$P(x, t) = \frac{1}{2} P(x+a, t-\tau) + \frac{1}{2} P(x-a, t-\tau). \quad (11.32)$$

Вычтем $P(x, t-\tau)$ из обеих частей (11.32), разделим на τ и перепишем (11.32) в виде

$$\frac{1}{\tau} [P(x, t) - P(x, t-\tau)] = \frac{a^2}{2\tau} [P(x+a, t-\tau) - 2P(x, t-\tau) + P(x-a, t-\tau)] a^{-2}. \quad (11.33)$$

Разлагая $P(x, t-\tau)$ и $P(x \pm a, t-\tau)$ в ряды Тейлора и переходя к пределу $a \rightarrow 0$ и $\tau \rightarrow 0$ при конечном отношении $D = a^2/2\tau$, получим уравнение диффузии

$$\frac{\partial P(x, t)}{\partial t} = D \frac{\partial^2 P(x, t)}{\partial x^2}. \quad (11.34a)$$

Обобщение (11.34a) на трехмерный случай выглядит следующим образом:

$$\frac{\partial P(x, y, z, t)}{\partial t} = D \nabla^2 P(x, y, z, t), \quad (11.34b)$$

где $\nabla^2 = \partial^2/\partial x^2 + \partial^2/\partial y^2 + \partial^2/\partial z^2$ — оператор Лапласа. Уравнение вида (11.34) называется уравнением *диффузии* или *Фоккера — Планка* и часто

используется для описания динамики молекул жидкости. В качестве «простого» упражнения найдите с помощью аналогичного метода дифференциальное уравнение, которому удовлетворяет $P(x, t)$ при $p \neq q$.

Можно показать, что решением уравнения (11.34а) для свободного пространства является распределение Гаусса с шириной, пропорциональной $t^{1/2}$:

$$P(x, t) = (2\pi Dt)^{-1/2} \exp(-x^2/4Dt), \quad (11.35)$$

где $P(x, t)$ — плотность вероятности того, что в момент времени t частица находится в точке x , если при $t = 0$ она находилась в точке $x = 0$. Прямой подстановкой вы можете убедиться в том, что (11.35) является решением (11.34а). Используя (11.35), можно получить

$$\langle x(t) \rangle = \int_{-\infty}^{\infty} dx x P(x, t) = 0 \quad (11.36)$$

и

$$\langle x^2(t) \rangle = \int_{-\infty}^{\infty} dx x^2 P(x, t) = 2Dt. \quad (11.37)$$

Мы получили, что величина $\langle x^2(t) \rangle$ пропорциональна t так же, как и для случайного блуждания на решетке. Обобщение (11.37) на d -мерный случай дает: $\langle R^2(t) \rangle = 2dDt$ [см. (11.6)], где R^2 обозначает квадрат смещения частицы.

Численное решение уравнений в частных производных *параболического типа*, к которым принадлежит (11.34), относится к нетривиальным задачам численного анализа (Пресс и др., Кунин.). Косвенный метод анализа уравнения (11.34) заключается в применении метода Монте-Карло, т.е. замене (11.34) на соответствующее случайное блуждание на решетке с дискретными временными шагами. Поскольку асимптотическое поведение уравнения в частных производных и модели случайного блуждания одинаково, то в данном случае метод Монте-Карло выступает как метод *численного анализа*. В противоположность этому если наша цель заключается в непосредственном изучении модели случайного блуждания на решетке, то метод Монте-Карло является методом *моделирования*. Иногда различие между моделированием и численным анализом определяется только точкой зрения исследователя.

11.6. СЛУЧАЙНЫЕ ЧИСЛА

До сих пор мы со спокойной совестью пользовались генератором случайных чисел, предусмотренным нашим языком программирования, для получения в приложениях метода Монте-Карло требуемых «случайных» чисел. В принципе можно было бы генерировать эти числа, используя какой-нибудь случайный физический процесс, как, например, радиоактивный распад. Однако такая последовательность чисел не воспроизводима и, следовательно, не может использоваться для проверки наших программ. Поэтому на практике пользуются цифровым компьютером, который является полностью детерминированной машиной, и получают последовательности «случайных» чисел по вполне конкретному алгоритму. Конечно, эти последовательности не являются истинно случайными, и в действительности их часто называют *псевдослучайными*. Тем не менее мы называем последовательности случайными, если они имеют равномерное распределение и удовлетворяют всем нашим критериям случайности. В дальнейшем мы увидим, что фактически многие из критериев случайности можно сформулировать на языке случайного блуждания.

Мы изложим кратко только некоторые свойства хорошего генератора случайных чисел. В большинстве генераторов случайных чисел получается последовательность, в которой каждое число используется для нахождения последующего по вполне определенному алгоритму. Таким образом, последовательность определяется *начальным значением* — первым ее числом. Все генераторы случайных чисел дают последовательность, повторяющуюся после некоторого количества членов, называемого *периодом*. Во всех обычных методах максимально возможный период связан с конечной длиной машинного «слова». В основе наиболее распространенного генератора случайных чисел лежит метод *вычетов*, или *линейный конгруэнтный метод*. А именно, если задано начальное число x_0 , то каждое число последовательности определяется «отображением»

$$x_n = (ax_{n-1} + c) \bmod m, \quad (11.38)$$

где a , c и m — натуральные числа. (Обозначение $y = z \bmod m$ означает, что m вычитается из z до тех пор, пока не получится $0 \leq y \leq m$. Например, $412 \bmod 50$ равно 12.) Отображение (11.38) характеризуется тремя параметрами: *множителем* a , *инкрементом* c и *модулем* m . Поскольку m является наибольшим целым числом, генерируемым с помощью (11.38), наибольший возможный период равен m . Однако в общем случае период зависит от всех трех параметров. Например, если $a = 3$, $c =$

$= 4$, $m = 32$ и $x_0 = 1$, то с помощью соотношения (11.38) генерируется последовательность 1, 7, 25, 15, 17, 23, 9, 31, 1, 7, 25, ... и период равен 8, а не максимально возможному значению 32. Если мы тщательно выберем a , c и m так, чтобы получался максимальный период, то в последовательности будут встречаться все целые числа от 0 до $m-1$. Поскольку обычно нужны случайные числа на отрезке $[0, 1]$, а не случайные целые числа, то генераторы случайных чисел выдают, как правило, отношение x_{n+1}/m , которое всегда меньше единицы. (Заметим, что $x_n = 0$ появляется в последовательности один раз.)

Достоинством метода вычетов является его высокое быстродействие. Некоторые его свойства исследуются в задаче 11.17. Как показывает приведенный пример, чтобы получить оптимальные результаты, необходимо тщательно выбрать параметры m , a и c . Разработаны некоторые правила получения наибольшего периода (Кнут).

Не существует необходимых и достаточных критериев случайности конечной последовательности чисел; самое большее, что мы можем сказать о конечной последовательности чисел, — это то, что она «выглядит» случайной. Поскольку ни один статистический критерий не является надежным индикатором случайности, в следующей задаче мы рассмотрим несколько критериев.

ЗАДАЧА 11.17. Статистические критерии случайности

а. Один из способов определения длительности периода генератора случайных чисел основан на построении графика зависимости полного смещения случайного блуждания от числа шагов. Каждый очередной шаг с равной вероятностью имеет длину ± 1 . На рис. 11.13 показан пример случайного блуждания, полученного с помощью формулы

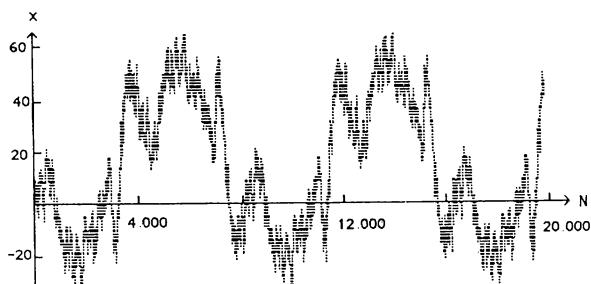


Рис. 11.13. Случайное блуждание, генерируемое с помощью соотношения (11.38) с $a = 899$, $c = 0$, $m = 32\,768$ и начальным числом $x_0 = 12$.

(11.38) с параметрами $a = 899$, $c = 0$, $m = 32\,768$ и $x_0 = 12$. Определите период последовательности, получаемой из (11.38) с приведенными значениями a , c и m . Определите также период генератора случайных чисел, имеющегося в вашем языке программирования.

б. В случайной последовательности должны содержаться числа, равномерно распределенные на отрезке $[0, 1]$. Однако последовательные числа могут появляться не идеально равномерно, а проявляют тенденцию к образованию групп, или коррелируют. В одном из тестов на наличие такой корреляции требуется заполнить случайным образом простую кубическую решетку из L^3 узлов. Рассмотрите массив $n(x, y, z)$, где $1 \leq x_i, y_i, z_i \leq L$, который сначала пуст. В качестве координат x_i , y_i и z_i каждой случайной точки берутся три последовательных случайных числа. Если узел пустой, то он заполняется и $n(x_i, y_i, z_i) = 1$; в противном случае состояние узла не меняется. Эта процедура повторяется tL^3 раз, где t — число шагов Монте-Карло на один узел. Поскольку данный процесс аналогичен распаду радиоактивного ядра, то можно ожидать, что доля пустых узлов решетки будет убывать как e^{-t} . Определите долю пустых узлов, используя генератор случайных чисел, предусмотренный в вашем языке программирования, для значений $L = 10, 15$ и 20 . Соответствуют ли полученные результаты ожидаемым? Если возможно, то повторите тот же тест, используя формулу (11.38) с параметрами $a = 65\,549$, $c = 0$ и $m = 231$.

в. Другой тест на равномерность заключается в делении отрезка $[0, 1]$ на M равных отрезков, или «корзин», и помещении каждого члена случайной последовательности в одну из корзин. В качестве примера рассмотрите первые $n = 10\,000$ чисел, сгенерированных по формуле (11.38) с параметрами $a = 106$, $c = 1283$ и $m = 6075$ (см. Пресс и др.). Поместите каждое число в одну из 100 корзин. Соответствует ли распределение чисел в корзинах законам статистики? Наиболее известным критерием такого соответствия является критерий *хи-квадрат*, или χ^2 . Пусть y_i — наблюдаемое количество чисел в i -й корзине и E_i — математическое ожидание. В нашем примере $E_i = 100$. Статистика *хи-квадрат* определяется выражением

$$\chi^2 = \frac{1}{M} \sum_{i=1}^M \frac{(y_i - E_i)^2}{E_i}. \quad (11.39)$$

(Заметим, что в учебниках определение χ^2 отличается множителем M .) Число χ^2 служит показателем согласия между наблюдаемым и ожидаемым распределениями. Если $\chi^2 \approx 0$, то согласие полное. При большом числе корзин или большом числе членов в каждой корзине значение $\chi^2 < 1$ указывает на то, что y_i распределены в соответствии с гауссовым распределением. Вычислите χ^2 для значений параметров a , c , и m , приведенных выше, и для генератора случайных чисел, предусмотренного вашим языком программирования.

г. Другой мерой коротких корреляций является автокорреляционная функция

$$C(k) = \frac{\langle x_i x_{i+k} \rangle - \langle x_i \rangle \langle x_{i+k} \rangle}{\langle x_i x_i \rangle - \langle x_i \rangle \langle x_i \rangle}, \quad (11.40)$$

где x_i есть i -й член последовательности. Величина $\langle x_i x_{i+k} \rangle$ вычисляется для каждого конкретного значения k суммированием всех возможных произведений $x_i x_{i+k}$ и делением на число произведений. Если x_i и x_{i+k} не коррелированы, то $\langle x_i x_{i+k} \rangle = \langle x_i \rangle \langle x_{i+k} \rangle$ и $C(k) = 0$. Равна ли тождественно нулю $C(k)$ для любой конечной последовательности? Вычислите функцию $C(k)$ для значений параметров $a = 106$, $c = 1283$ и $m = 6075$.

*ЗАДАЧА 11.18. «Улучшенный» генератор случайных чисел

Одним из способов уменьшить последовательную корреляцию и увеличить период является «перемешивание», или «тасование», двух различных генераторов случайных чисел. Нижеследующая процедура иллюстрирует этот метод для двух генераторов случайных чисел, которые мы обозначим RAN1 и RAN2.

1. С помощью генератора RAN1 составьте список или таблицу 256 случайных чисел. (Число 256 выбрано произвольно, но оно должно быть меньше периода генератора RAN1.)
2. Выберите из этого списка случайное число x , генерируя с помощью RAN2 случайный порядковый номер от 1 до 256.
3. Замените это число в списке новым случайным числом, сформированным генератором RAN1.

Искомым результатом является *x*. Рассмотрите два генератора случайных чисел с относительно короткими периодами и сильной последовательной корреляцией и покажите, что приведенная выше схема тасования улучшает качество случайных чисел.

ЛИТЕРАТУРА

J. Bernasconi, L. Pietronero, True self-avoiding walk in one dimension, *Phys. Rev. B* **29**, 5196 (1984). Научная работа. Авторы приводят значение показателя ν с 1% точностью.

S. Chandrasekhar, Stochastic problems in physics and astronomy, *Rev. Mod. Phys.* **15**, 1 (1943). Эта статья перепечатана в *Selected Papers on Noise and Stochastic Processes*, ed. M. Wax, Dover Publications, 1954. [Имеется перевод: *Чандрасекар С.*, Стохастические проблемы в физике и астрономии. — М.: ИЛ, 1947.]

A. K. Dewdney, Computer Recreations (Five easy peaces for a do loop and random-number generator), *Sci. Amer.* **252**, 20 (April 1985).

R. Ehrlich, *Physics and Computers*, Houghton Mifflin, 1973. В гл. 4 обсуждается метод вычетов.

P.-G. de Gennes, *Scaling Concepts in Polymer Physics*, Cornell University Press, 1979. Важная, но трудная для чтения книга.

S. Havlin, G. H. Weiss, J. Kiefer, M. Dishon, Exact enumeration of random walks with traps, *J. Phys. A: Math. Gen.* **17**, L347 (1984). Авторы обсуждают метод вычисления времени жизни случайных пешеходов, основанный на полном переборе, на решетке со случайным распределением ловушек.

D. E. Knuth, *Seminumerical Algorithms*, 2nd ed., vol. 2 of *The Art of Computer Programming*, Addison-Wesley, 1981. [Имеется перевод: *Д. Е. Кнут*, Искусство программирования на ЭВМ, т. 2. — М.: Мир, 1979.] Стандартная ссылка на генераторы случайных чисел.

E. W. Montroll, M. F. Shlesinger, On the wonderful world of random walks. In: *Nonequilibrium Phenomena 2: From Stochastic to Hydrodynamics*, *J. L. Lebowitz, E. W. Montroll*, eds., North-Holland Press, 1984. В первой части этого прекрасного обзора освещаются исторические аспекты случайных блужданий.

W. H. Press, B. P. Flannery, S. A. Teukolsky, W. T. Vetterling, *Numerical Recipes*, Cambridge University Press, 1986. [Имеется перевод: *В. П. Пресс, Б. П. Фланнери, С. А. Тьюкольски, В. Т. Веттерлинг*. Численные рецепты. — М.: Мир, 1990.]

F. Reif, Fundamentals of Statistical and Thermal Physics, McGraw-Hill, 1965. В этом широко известном учебнике по статистической физике хорошо рассмотрены случайные блуждания (гл. 1) и диффузия (гл. 12).

M. N. Rosenbluth, A. W. Rosenbluth, Monte Carlo calculation of the average extension of molecular chains, J. Chem. Phys. **23**, 356 (1955). Один из первых расчетов методом Монте-Карло блужданий без самопересечений.

J. R. Taylor, An Introduction to Error Analysis, University Science Books, Oxford University Press, 1982. [Имеется перевод: Дж. Тейлор. Введение в теорию ошибок. — М.: Мир, 1985.] В гл. 12 обсуждается критерий χ^2 .

F. T. Wall, F. Mandel, Macromolecular dimensions obtained by efficient Monte Carlo method without sample attrition, J. Chem. Phys., **63**, 4592 (1975). Изложение модели репаций, или «ползущей змеи».

G. H. Weiss, R. P. Rubin, Random Walks: Theory and Selected Applications, Adv. Chem. Phys. **52**, 507 (1983). Несмотря на исследовательский характер, многое в этом обзоре доступно студентам.

G. H. Weiss, S. Havlin, Trapping of random walks on the line, J. Stat. Phys. **37**, 17 (1984). Авторы обсуждают аналитический метод вычисления асимптотического поведения одномерных случайных пешеходов в случае случайного распределения узлов-ловушек.

C. A. Whitney, Generating and Testing Pseudorandom Numbers, Byte, pp. 128–464 (October 1984). Доступная и информативная статья, содержащая тесты для функции `rnd` языка IBM BASICA.

ДОПОЛНИТЕЛЬНАЯ ЛИТЕРАТУРА

А. Ю. Гросберг, А. Р. Хохлов, Физика в мире полимеров. — М.: Наука, 1989. В этой просто написанной книжке обсуждаются проблема исключенного объема, метод репаций и многое другое.

А. Ю. Гросберг, А. Р. Хохлов, Статистическая физика макромолекул. — М.: Наука, 1989. Книга рассчитана на подготовленного читателя. Приведена обширная библиография.

И. М. Соболев, Численные методы Монте-Карло. — М.: Наука, 1973. В гл. 1 обсуждаются вопросы получения случайных последовательностей и статистические критерии.

ПРИЛОЖЕНИЕ 11А. МЕТОД НАИМЕНЬШИХ КВАДРАТОВ

В задаче 11.2 мы рассчитывали величины $\langle x_N \rangle$ и $\langle x_N^2 \rangle$ для $N = 8, 16, 32$ и 64 . Результаты представлены в табл. 11.1. Приведены также среднеквадратичное смещение $R_N = \sqrt{\langle x_N^2 \rangle - \langle x_N \rangle^2}$ и величина $\langle x_N^4 \rangle$. Из табл. 11.1 можно заключить, что общий результат $\langle x_N \rangle = 0$ при $p = 1/2$ согласуется с нашим, полученным для небольшого числа испытаний.

ТАБЛИЦА 11.1. Значения $\langle x_N \rangle$, $\langle x_N^2 \rangle$, $R_N = \sqrt{\langle x_N^2 \rangle - \langle x_N \rangle^2}$ и $\langle x_N^4 \rangle$, полученные методом Монте-Карло для одномерного случайного блуждания. Величины усреднялись по 1000 испытаний для каждого значения N . Вероятность шага равна $p = 1/2$

N	$\langle x_N \rangle$	$\langle x_N^2 \rangle$	R_N	$\langle x_N^4 \rangle$
8	-0.010	7.980	2.825	183.984
16	-0.068	16.360	4.044	794.080
32	-0.006	30.284	6.830	2708.530
64	-0.022	65.348	8.084	12110.900

Основной рассматриваемой величиной в табл. 11.1 является зависимость R_N от N . Поскольку мы предполагаем, что для достаточно больших N величина R_N ведет себя как N^{ν} , мы можем построить график зависимости $\log R_N$ от $\log N$ и попытаться найти прямую, которая проходит как можно ближе к этим точкам. Чему равна оценка величины наклона ν , полученная из ваших результатов? Хотя часто визуальная подгонка может давать замечательные результаты, желательно как-то систематизировать процедуру подгонки.

Аналитический метод нахождения наилучшей прямой, проходящей через ряд экспериментальных точек, называется методом *линейной регрессии* или *наименьших квадратов*. Предположим, у нас имеется n пар измерений $(x_1, y_1), (x_2, y_2), \dots, (x_n, y_n)$ и ошибки полностью содержатся в значениях y . Предположим также на минуту, что все статистические ошибки в y одинаковые по величине. Наша цель состоит в том, чтобы наилучшим образом подобрать функцию

$$y = mx + b. \quad (11.41)$$

Задача заключается в вычислении значений параметров m и b для наи-

лучшей прямой, проходящей через n точек. Если бы не было ошибок в значениях y_i , мы имели бы $y_i - mx_i - b = 0$. Остаток d_i определяется выражением

$$d_i = y_i - mx_i - b \quad (11.42)$$

и является мерой ошибки в y_i . Следовательно, представляется разумным предположить, что наилучшие значения m и b — это те, которые минимизируют величину

$$S = \sum_{i=1}^n (y_i - mx_i - b)^2. \quad (11.43)$$

Для нахождения минимума S продифференцируем эту величину соответственно по m и b :

$$\frac{\partial S}{\partial m} = -2 \sum_{i=1}^n x_i (y_i - mx_i - b) = 0, \quad (11.44)$$

$$\frac{\partial S}{\partial b} = -2 \sum_{i=1}^n (y_i - mx_i - b) = 0. \quad (11.45)$$

(В дальнейшем мы не будем писать у знака суммы пределы суммирования $i=1$ и n .) Из (11.44) и (11.45) получаем систему двух уравнений:

$$m \sum x_i^2 + b \sum x_i = \sum x_i y_i, \quad (11.46)$$

$$m \sum x_i + bn = \sum y_i. \quad (11.47)$$

Решениями уравнений (11.46) и (11.47) являются

$$m = \frac{\sum (x_i - \bar{x}) y_i}{\Delta}, \quad (11.48)$$

$$b = \bar{y} - m\bar{x}, \quad (11.49)$$

где

$$\bar{x} = \frac{1}{n} \sum x_i, \quad \bar{y} = \frac{1}{n} \sum y_i \quad (11.50)$$

и

$$\Delta = \sum (x_i - \bar{x})^2. \quad (11.51)$$

Из уравнений (11.48) и (11.49) вычисляются наклон и отрезок, отсекаемый на оси, для наилучшей прямой, проведенной через n точек.

В нашем примере случайного блуждания можно преобразовать нелинейную зависимость $R_N = aN^v$ в линейную

$$\ln R_N = \ln a + v \ln N. \quad (11.52)$$

По значениям R_N и N из табл. 11.1, получаем значения $x = \ln N$ и $y = R_N$, приведенные в табл. 11.2 и на рис. 11.14.

ТАБЛИЦА 11.2. Вычисленные значения $\ln N$ и $\ln R_N$ для одномерного случайного блуждания с 1000 испытаний

$\ln N$	$\ln R_N$
2.079	1.039
2.773	1.397
3.466	1.921
4.159	2.090

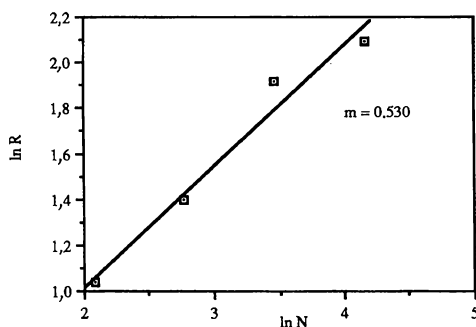


Рис. 11.14. График зависимости $\ln R_N$ от $\ln N$ для одномерного случайного блуждания. Значения $\ln R_N$ взяты из табл. 11.2. Прямая, проходящая через точки, найдена путем минимизации выражения (11.43).

Используя формулы (11.48)–(11.51) и значения $\ln N$ и $\ln R_N$ из табл. 11.2, находим, что $\bar{x} = 3.119$, $\bar{y} = 1.612$, $\Delta = 2.403$, $m = 0.530$ и $b = -0.043$. Следовательно, можно сделать вывод, что наши ограниченные данные для R_N дают оценку $\nu = 0.53$, которая находится в соответствии с точным результатом $\nu = 1/2$.

Приведем без доказательства оценки доверительных интервалов для m и b :

$$(\Delta m)^2 \approx \frac{1}{D} \frac{\sum d_i^2}{D(n-2)}, \quad (11.53)$$

$$(\Delta b)^2 \approx \left(\frac{1}{n} + \frac{\bar{x}^2}{\Delta}\right) \frac{\sum d_i^2}{n-2}. \quad (11.54)$$

В нашем случае $\Delta m = 0.07$ и мы делаем вывод, что наилучшая оценка ν есть $\nu = 0.53 \pm 0.07$.

Если статистические ошибки σ_i в y_i различны, то данные необходимо умножить на веса $w_i = 1/\sigma_i^2$. В этом случае следует минимизировать величину

$$S = \sum w_i (y_i - mx_i - b)^2. \quad (11.55)$$

Можно показать, что наилучшие оценки величин m и b равны

$$m = \frac{1}{\Delta} \sum w_i (x_i - \bar{x}) y_i, \quad (11.56)$$

$$b = \bar{y} - m\bar{x}, \quad (11.57)$$

$$\bar{x} = \frac{\sum w_i x_i}{\sum w_i}, \quad \bar{y} = \frac{\sum w_i y_i}{\sum w_i}, \quad (11.58)$$

$$\Delta = \sum w_i (x_i - \bar{x})^2. \quad (11.59)$$

Как можно оценить ошибку $\ln R_N$? В гл. 10 мы установили, что вероятная ошибка $\langle x_N \rangle$ равна σ_x/\sqrt{n} , где $\sigma_x^2 = \langle x_N^2 \rangle - \langle x_N \rangle^2$. Используя аналогичные соображения, можно показать, что вероятная ошибка $\langle x_N^2 \rangle$ равна $\tilde{\sigma}/\sqrt{n}$, где

$$\tilde{\sigma}^2 = \langle x_N^4 \rangle - \langle x_N^2 \rangle^2. \quad (11.60)$$

Ошибки $\ln R_N$ вычисляются в задаче 11.19.

***ЗАДАЧА 11.19.** Оценка вероятной ошибки $\ln R_N$

а. Используя результаты, полученные для $\langle x_N^4 \rangle$ и $\langle x_N^2 \rangle$, вычислите стандартное отклонение $\tilde{\sigma}$ и $\sigma_m = \tilde{\sigma}/\sqrt{n}$ для каждого значения N . Величина σ_m является наиболее вероятной ошибкой одного измерения величины $\langle x_N^2 \rangle$, состоящего из n испытаний. Равны ли приблизительно значения σ_m для всех N ? Если нет, то качественно объясните поведение σ_m . Если желательно иметь приблизительно равные оценки ошибки для всех значений N , то сколько испытаний требуется для случая $N = 64$ по сравнению с $N = 8$?

б. Поскольку рассматриваемой величиной служит *получаемая* величина $\ln R_N$, а не *исходная* $\langle x_N^2 \rangle$, необходимо связать оцененную ошибку $\langle x_N^2 \rangle$ с ошибкой $\ln R_N$. Можно показать, что если U и A связаны соотношением $U = A^p$, то ошибка ΔU связана с ошибкой ΔA следующим образом:

$$\frac{\Delta U}{U} = p \frac{\Delta A}{A}. \quad (11.61)$$

(U и A — средние значения.) Аналогично, если $U = \ln A$, то ошибки связаны соотношением

$$\Delta U = \frac{\Delta A}{A}. \quad (11.62)$$

С помощью формул (11.61) и (11.62) оцените ошибки $\ln R_N$ для каждого значения N , приведенного в табл. 11.1. Поскольку значения $\ln R_N$ имеют разные ошибки, используйте формулы (11.56)–(11.59) для получения улучшенной оценки ν .

Для простого случайного блуждания, рассмотренного здесь, соотношение $R_N = AN^p$ справедливо для всех значений N . Однако во многих других задачах настоящей главы это соотношение выполняется только асимптотически для больших N . Поэтому, возможно, неправильно придавать дополнительный вес результатам для R_N при небольших N .

ЛИТЕРАТУРА

W. S. Cleveland, R. McGill, Graphical perception and graphical methods for analyzing scientific data, Science **229**, 828 (1985). Кроме метода наименьших квадратов существуют другие методы анализа данных.

ЗАДАЧА О ПЕРКОЛЯЦИИ

12

В этой главе на примере задачи о перколяции вводятся понятия, относящиеся к критическим явлениям.

12.1. ВВЕДЕНИЕ

В этой главе, посвященной геометрическим фазовым переходам, не требуется обширных познаний физики, например не требуется знания классической или квантовой механики и статистической физики. В самом деле, все, что требуется, ограничивается некоторыми понятиями геометрии и теории вероятностей. Главной притягательной силой геометрических фазовых переходов являются их игровые аспекты и интуитивная простота. Кроме того, эти модели служат прекрасным введением в компьютерное моделирование и указывают на важность методов графического анализа. С другой стороны, знание основ физики сделает эту главу более значительной и может послужить введением в задачи фазовых переходов и в такие важные представления, как масштабирование, критические показатели и ренорм-группа.

Вам может быть знаком термин «просачивание» в контексте задачи о заваривании кофе. Однако мы будем использовать это термин в более узком смысле. Для ознакомления с явлением просачивания мы рассмотрим другой важный (кухонный) пример. Представьте себе большой противень, на котором случайным образом размещены кружки жидкого теста. Затем противень с печеньем помещается в духовку. Допускается, что в процессе выпечки каждая капля теста может расплыться до максимального размера радиусом a . По своему опыту вы, вероятно, знаете, что если два печенья соприкасаются, то они сливаются и образуется одно печенье. Как вы думаете, что произойдет? Если вы не следите за выпечкой, то можете получить одно гигантское печенье, которое будет занимать значительную часть противня (рис. 12.1).

Давайте абстрагируемся от этого примера для уяснения концепции просачивания (перколяции). Представьте себе большую шахматную дос-

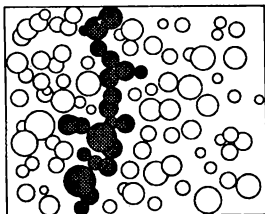


Рис. 12.1. Кружки (печенья) различного радиуса размещаются случайным образом на большом листе. Заметим, что существует путь из перекрывающихся кружков (закрашены), который соединяет верхнюю и нижнюю стороны «противня». Если такой путь существует, мы говорим, что этот путь «просачивается» (перколирует) сквозь структуру.

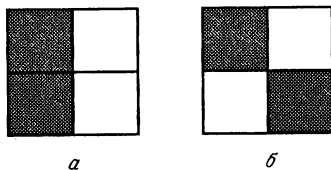


Рис. 12.2. Пример ячеечного перколяционного кластера на квадратной решетке со стороной $L = 2$. Две ближайшие занятые ячейки (закрашенные квадраты) в случае *a* являются частью одного кластера; в случае *б* две занятые ячейки не являются соседними квадратами и не принадлежат одному кластеру.

ку, а не противень. Будем представлять эту шахматную доску как квадратную решетку и предположим, что каждый квадрат, или «ячейка», этой решетки может находиться в двух состояниях: «занято» или «пусто». Каждая ячейка занимает с вероятностью p независимо от состояния соседних ячеек. Эта модель называется *ячеечной* перколяцией. Занятые ячейки («печенья») либо изолированы друг от друга, либо образуют группы, состоящие из ближайших соседей. Мы определим *кластер* как группу занятых ячеек решетки, связанных с ближайшим соседом по стороне ячейки (рис. 12.2). Две занятые ячейки принадлежат одному кластеру, если они соединены путем, состоящим из занятых ячеек.

Один из простых способов изучения перколяции основан на использовании генератора случайных чисел карманного калькулятора. Вся процедура сводится к тому, чтобы сгенерировать случайное число, а затем занять ячейку решетки, если случайное число меньше p . Выполним эту процедуру для каждой ячейки решетки. Если вероятность занятия ячейки мала, то можно ожидать, что будут присутствовать только небольшие изолированные кластеры (рис. 12.3, *a*). По сравнению с этим если $p \sim 1$, то ожидается, что большинство занятых ячеек образуют один большой кластер, который протянется от одной стороны решетки до другой (рис. 12.3, *г*). О таком кластере говорят, что он «перекидывается» через решетку, и называют *соединяющим кластером*. Что произойдет для промежуточных значений p , например для p от 0.4 до 0.7 (рис. 12.3, *б* и *в*)? Мы увидим, что в пределе бесконечной решетки существует вполне определенная «пороговая» вероятность p такая, что

Для $p \geq p_c$ существует один соединяющий кластер, или путь;
 для $p < p_c$ нет ни одного соединяющего кластера и все кластеры конечны.

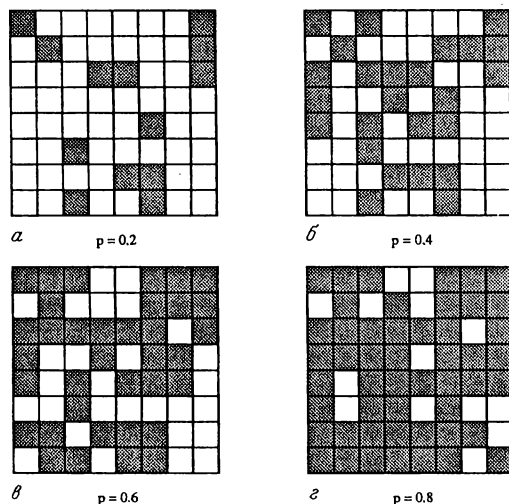


Рис. 12.3. Примеры ячеечных перколяционных кластеров на квадратной решетке со стороной $L = 8$ для значений $p = 0.2, 0.4, 0.6$ и 0.8 . В среднем доля занятых ячеек (закрашенные квадраты) равна p . Заметим, что для значения $p = 0.6$ существует кластер, который «соединяет» стороны решетки в вертикальном направлении, но не в горизонтальном; при значении $p = 0.8$ кластер соединяет стороны решетки и по вертикали, и по горизонтали.

Следует подчеркнуть, что характерной особенностью, присущей перколяции, является *связность*. Поскольку связность обнаруживает качественное изменение при конкретном значении некоторого параметра, который можно менять непрерывно, мы увидим, что переход из состояния, не содержащего соединяющий кластер, в состояние с одним соединяющим кластером представляет собой *фазовый переход*.

Конечно, наш интерес не связан с печеньем или даже ячеечной перколяцией. Примером приложения понятий перколяции является электропроводность сложных систем, состоящих из проводящих и непроводящих материалов. Простой способ изготовления такой системы в лабораторных условиях заключается в помещении в контейнер смеси маленьких металлических и пластмассовых шариков (см. Фицпатрик и др.). Особое внимание надо уделить случайному размещению шариков. Если металлические шарiki составляют малую долю объема системы, то электрический ток не может пройти через комбинированную систему и она будет

изолятором. Однако если металлические шарики составляют достаточно большую часть объема контейнера, то электрический ток будет в состоянии протекать через области, занимаемые этими шариками, и система будет проводником. Описание протекания электрического тока через комбинированные материалы можно сделать более точным, вводя параметр ϕ — долю объема контейнера, занимаемую металлическими шариками. Переход из одного режима в другой (изолятор и проводник) происходит внезапно по мере увеличения параметра ϕ и соответствует отсутствию или наличию *соединяющего пути* из металлических шариков. Более реалистичные комбинированные системы обсуждаются Цалленом.

Явления протекания можно также наблюдать в лаборатории на куске проволоочной сетки. Возможно, вам захочется проделать эксперимент Уотсона и Лиса, которые измеряли электропроводность большого куска однородной стальной сетки в зависимости от доли вырубленных ячеек. Координаты ячеек, которые будут вырубаться, определяются с помощью генератора случайных чисел. Уотсон и Лис установили, что измеряемая электропроводность является быстро убывающей функцией части оставшихся ячеек p и стремится к нулю при значениях ниже критического порогового. Недавно выполнено аналогичное измерение проводимости листа проводящей бумаги со случайно расположенными «дырками» (см. Мер и др.).

Приложения явлений протекания относятся к переходам металл — изолятор и проводимости электрической сетки (случайная резисторная цепь), для учета распространения эпидемий в популяции, поведения магнитов, содержащих примеси, и описания гелей. Чем отличается шарик геля от бульонного шарика? В дальнейшем мы сконцентрируем наше внимание на изучении некоторых простых моделей перколяции, которые апеллируют к нашей интуиции. Многие приложения концепции перколяции обсуждаются в литературе, приведенной в конце главы.

12.2. ПОРОГ ПЕРКОЛЯЦИИ

Поскольку неудобно генерировать перколяционные конфигурации с помощью калькулятора, мы разработаем простую программу. Рассмотрим квадратную решетку со стороной L и присвоим каждой ячейке этой решетки случайные числа от нуля до единицы. Ячейка занимается, если присвоенное ей случайное число меньше p . Программа *site*, распечатка которой приведена ниже, порождает ячеечную перколяционную конфигу-

рацию и выводит ее на экран компьютера для наглядного представления кластеров. В массиве r основной программы хранятся случайные числа, присваиваемые каждой ячейке. Заметим, что каждой ячейке решетки присваивается случайное число, так что если p увеличивается, то занятые ячейки остаются занятыми.

```

PROGRAM site          ! рисуются конфигурации ячеечной перколяции
DIM r(50,50)
RANDOMIZE
CALL initial(L)       ! задаются параметры решетки и экрана
CALL lattice(L,r)     ! придание каждой ячейке случайного числа
CALL configuration(L,r) ! занятие ячеек с данной вероятностью p
END

SUB initial(L)
INPUT prompt "размер решетки = ": L
LET aspect_ratio = 1.5    ! значение для компьютера Macintosh
LET margin = 0.1*L
LET mx = aspect_ratio*margin
LET bx = aspect_ratio*L
SET window -mx,bx + mx,-margin,L + margin
BOX LINES 0,L,0,L
END SUB

SUB lattice(L,r(,))
FOR row = 1 to L          ! рисование ячеек решетки
  LET y = row - 0.5
  ! связывает с каждой ячейкой квадрат со стороной 1
  FOR col = 1 to L
    LET x = col - 0.5
    LET r(col,row) = rnd ! каждой ячейке придается случайное число
    PLOT POINTS: x,y
  NEXT col
NEXT row
END SUB

```

```

SUB configuration (L,r(.))
  DIM s(50,50)
  DO while p >= 0
    SET cursor 1,1
    INPUT prompt "вероятность p = ": p
    LET size = 0.4      ! половина стороны квадрата для занятых ячеек
    FOR row = 1 to L
      LET y = row - 0.5
      FOR col = 1 to L
        IF r(col,row) < p and s(col,row) <> 1 then ! ячейку занимаем
          LET x = col - 0.5
          BOX AREA x - size,x + size, y - size,y + size
          LET s(col,row) = 1 ! ячейка занята
        END IF
      NEXT col
    NEXT row
  LOOP
END SUB

```

Порог перколяции p_c определяется как такая вероятность p , при которой появляется первый бесконечный кластер на бесконечной решетке. Однако для конечной решетки со стороной L , которую мы можем промоделировать на компьютере, всегда существует ненулевая вероятность того, что будет появляться соединяющий кластер, связывающий одну сторону решетки с другой. Для малых значений p эта вероятность порядка p^L (рис. 12.4). По мере увеличения L эта величина стремится к нулю, и для достаточно малых значений p будут существовать только

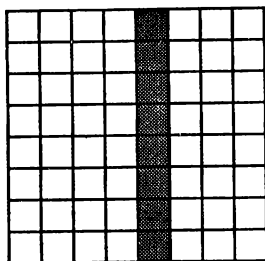


Рис. 12.4. Пример соединяющего кластера, появляющийся с вероятностью p^L на конечной решетке с $L = 8$. Сколькими способами можно представить соединяющий кластер, состоящий из L ячеек?

конечные кластеры. Поскольку нам необходимо применить правило «протекания» для конечной решетки, мы определим $p_c(L)$ как среднее значение p , при котором впервые появляется соединяющий кластер. Для конечной решетки определение протекания произвольно и, следовательно, вычисленное значение p_c зависит от критерия протекания. Например, мы можем определить соединяющий путь одним из способов: он связывает решетку либо в горизонтальном, либо в вертикальном направлении; соединяет решетку в выбранном направлении (например, в вертикальном); соединяет решетку в обоих направлениях. Все эти правила протекания должны приводить к одному и тому же экстраполированному значению p_c при $L \rightarrow \infty$. В следующей ниже задаче мы получим приближенное значение p_c с точностью около 10%. Более тонкий анализ, который называется конечномерным масштабированием, позволит нам экстраполировать результаты p_c на предельный случай $L \rightarrow \infty$. Этот метод обсуждается в разд. 12.4.

ЗАДАЧА 12.1. Порог перколяции для квадратной решетки

а. Воспользуйтесь программой `site` для генерирования ячеечной перколяционной конфигурации на квадратной решетке. Оцените p_c , находя такое значение p , при котором впервые появляется соединяющий кластер. Сначала рассмотрите случай с $L = 4$ и начните со значения p , для которого вы уверены, что маловероятно наличие соединяющего кластера. Затем увеличивайте значение p с шагом 0.025 до появления соединяющего кластера. Запомните значение p , при котором впервые появляется соединяющий кластер для каждого критерия протекания. Помните, что каждому испытанию соответствуют различные наборы случайных чисел. Повторите эту процедуру для десяти испытаний. Оценка p_c является средним значением по десяти испытаниям для $L = 4$. Совпадают ли ваши значения p_c с предполагаемыми значениями для каждого критерия протекания?

б. Повторите вычисления п. «а» для значений $L = 16$ и 32 . Вычисляется ли значение p_c точнее в случае больших L , т.е. является ли разброс численных значений p_c меньше? Насколько быстро вы можете визуально определять наличие соединяющего кластера? Воспользуйтесь, насколько это возможно, сущностью своего визуального «алгоритма» для определения наличия соединяющего кластера.

Значение p_c зависит как от размеров решетки, так и от ее симмет-

рии. Кроме квадратной решетки наиболее известной двумерной решеткой является треугольная. Как обсуждалось в гл. 11, существенным различием между квадратной и треугольной решетками является число ближайших соседей.

ЗАДАЧА 12.2. Ячеечная перколяция на треугольной решетке

а. Модифицируйте программу `site` таким образом, чтобы рассмотреть ячейечные перколяционные конфигурации на треугольной решетке. Простой способ модификации вашей программы заключается в учете узлов, имеющих симметрию квадратной решетки, но считая ближайшими соседями узлы, расположенные по диагонали.

б. Предположите, что соединяющий путь связывает верхние и нижние узлы решетки (рис. 12.5). Считаете ли вы, что значение p_c для треугольной решетки будет меньше или больше значения p_c для квадратной решетки? Вычислите $p_c(L)$ для случаев $L = 16$ и 32 . Совпадают ли ваши значения p с предполагаемыми?

Еще один тип просачивания — *цепная* перколяция¹⁾. Представьте, что каждый узел решетки занят и существуют узлы, соединенные со своими ближайшими соседями (рис. 12.6). В цепной перколяции каждое звено либо занимает с вероятностью p , либо остается свободным с вероятностью $1 - p$. Кластер представляет собой группу узлов, соединенных между собой занятыми звеньями. Простым примером цепной перколяции служит проволочная сетка, описанная в разд. 12.1. На этот раз мы можем представить вырезание звеньев между центрами пересече-

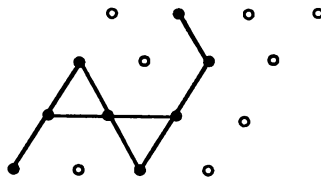


Рис. 12.5. Пример соединяющего кластера в случае треугольной решетки с $L = 4$. Звенья между занятыми узлами изображены, чтобы уточнить структуру решетки. Заметим, что иногда удобно связать соседние вершины решетки с ячейками.

¹⁾В литературе на русском языке иногда используется термин *задача о связях*. — Прим. перев.

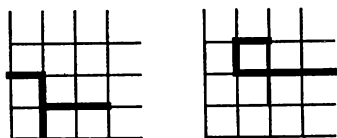


Рис. 12.6. Два примера цепных кластеров. Занятые звенья показаны жирными прямыми.

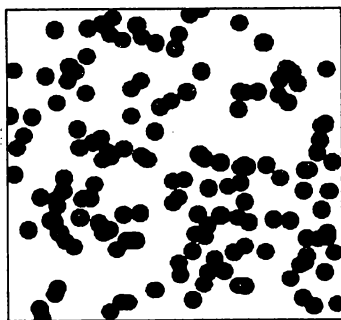


Рис. 12.7. Модель непрерывной перколяции, реализованная размещением дисков единичного радиуса случайным образом в квадратном ящике со стороной L . Существует ли соединяющий кластер? Количество дисков равно 150, $L = 20$ и плотность дисков $x = 150/(20)^2 = 0.375$.

ний, а не удаление самих узлов. Если бы мы вздумали измерить электропроводность, то нашли бы, что она пренебрежимо мала, когда доля оставленных звеньев меньше или равна примерно 0.5. Хотя цепная перколяция и другие модели решеточной перколяции представляют интерес, мы будем для простоты рассматривать только ячеечную перколяцию.

Кроме моделей решеточной перколяции мы также можем рассмотреть *непрерывные* перколяционные модели. Например, в приведенной ниже программе диски единичного радиуса помещаются случайным образом в ящик. Два диска принадлежат одному кластеру, если они касаются друг друга или перекрываются. Типичная непрерывная перколяционная конфигурация изображена на рис. 12.7. Мы снова можем задаться вопросом: «Какая минимальная плотность дисков необходима для существования связанного пути дисков?».

```

PROGRAM continuum      ! рисуются конфигурации непрерывной перколяции
RANDOMIZE
CALL initial(L)        ! см. распечатку подпрограммы в программе site
CALL disks(L)          ! случайное размещение дисков в квадрате
END

SUB disks(L)
  LET r = 0.5           ! радиус дисков
  LET area = L*L        ! L - сторона квадрата
  DO while n >= 0
    SET cursor 5,48
    INPUT prompt "число новых дисков = ": n
    FOR i = 1 to n
      LET x = rnd*L     ! центр диска внутри квадрата
      LET y = rnd*L
      BOX CIRCLE x - r, x + r, y - r, y + r
    NEXT i
    LET number = number + n
    SET cursor 1,1
    PRINT "                                     ";    ! стирание строки
    SET cursor 1,1
    PRINT "плотность дисков = "; number/area;
  LOOP
END SUB

```

ЗАДАЧА 12.3. Непрерывная перколяция

а. Воспользуйтесь программой `continuum` для вычисления значения порога перколяции x_c , где параметр x представляет собой плотность дисков.

*б. Модифицировав программу `continuum` так, чтобы диски произвольного размера от 0 до 1 случайно располагались в квадратном ящике, рассмотрите простую модель задачи о «противне», обсуждаемую в разд. 12.1. Как изменится значение x_c ?

При обсуждении перколяции мы подчеркнули, что существует порог перколяции p_c и появляется соединяющий путь, или кластер, при $p \geq p_c$. Более полную информацию можно получить из *распределения среднего размера кластеров* $n_s(p)$, определяемого формулой

$$n_s(p) = \frac{\text{среднее число кластеров размером } s}{\text{полное число ячеек решетки}}. \quad (12.1)$$

При $p \geq p_c$ соединяющий кластер исключается из n_s . (По историческим причинам под *размером* кластера подразумевается *число* ячеек в кластере, а не его пространственная протяженность.) Тщательное изучение рис. 12.3,а показывает, что $n_s(p=0) = 5/64$, $1/64$ и $2/64$ для $s = 1, 2$ и 3 соответственно и равно нулю в противном случае. Поскольку $\sum_s s n_s$ представляет собой полное число занятых ячеек, а $s n_s$ — количество занятых ячеек в кластере размером s , величина

$$w_s = \frac{s n_s}{\sum_s n_s} \quad (12.2)$$

является вероятностью того, что занятый узел, выбранный случайным образом, принадлежит кластеру размером s . Следовательно, *средний размер кластера* S определяется как

$$S = \sum_s s w_s = \frac{\sum_s s^2 n_s}{\sum_s s n_s}. \quad (12.3)$$

В качестве примера рассмотрим рис. 12.3,а. Средний размер кластера, соответствующий восьми кластерам на этом рисунке, равен $S = 27/13$.

Другой величиной, характеризующей перколяцию, является $P_\infty(p)$ — вероятность того, что занятая ячейка принадлежит соединяющему кластеру. Функция $P_\infty(p)$ определяется следующим образом:

$$P_\infty = \frac{\text{число ячеек в соединяющем кластере}}{\text{полное число занятых ячеек}}. \quad (12.4)$$

В случае бесконечной решетки $P_\infty(p) = 0$ при $p < p_c$ и $P_\infty(p) = 1$ при $p = 1$. Внимательное изучение рис. 12.3,в показывает, что $P_\infty(p = 0.6) = 36/47$ для приведенной конфигурации.

ЗАДАЧА 12.4. Качественное поведение функций $n_s(p)$, $S(p)$ и $P_\infty(p)$

а. Вычислите распределение размеров кластеров $n_s(p)$ на квадратной решетке с $L = 16$ и $p = 0.4$, $p = p_c$, $p = 0.8$. Положите $p_c = 0.5927$. Проведите по пять испытаний для каждого значения p и усредните $n_s(p)$ по этим пяти испытаниям. Согласующиеся результаты можно получить, отбросив конфигурации с соединяющими кластерами для случая $p < p_c$ и конфигурации без соединяющих кластеров для случая $p \geq p_c$. Постройте график функции n_s в зависимости от s для каждого значения p и качественно объясните полученную зависимость. В каком из случаев, $p = p_c$ или $p \neq p_c$, величина n_s будет расти быстрее по мере увеличения s ?

б. Используйте конфигурацию из п. «а» для вычисления зависимости среднего размера кластера S от p . Не забудьте, что при $p > p_c$ соединяющие кластеры не учитываются.

в. Вычислите $P_\infty(p)$ для случая с $L = 16$ и различных значений $p > p_c$. Постройте график функции $P(p)$ и качественно обсудите ее поведение.

***12.3. МАРКИРОВКА КЛАСТЕРОВ**

Вероятно, ваш визуальный алгоритм определения наличия соединяющего пути и вычисления количества кластеров является очень изощренным. Включает ли он в себя параллельную обработку данных? Однако выполнение вашего алгоритма утомляет и поглощает много времени, и мы хотели бы переложить эту работу на компьютер. Как мы увидим в дальнейшем, такая работа трудна даже для компьютера. Эта трудность связана с тем, что принадлежность ячейки некоторому кластеру является *глобальным* свойством ячейки, а не *локальным*.

Мы рассмотрим метод многократной маркировки кластеров Хошена и Копельмана (см. литературу в конце главы). Алгоритм лучше всего описать на примере. Рассмотрим конфигурацию, изображенную на рис. 12.8. Мы присваиваем ячейкам кластерные метки, двигаясь из нижнего левого угла вправо. Поскольку *ячейка(1,1)* занята, мы присваиваем ей кластерную метку 1. Следующая ячейка пустая, а значит, не маркируется. Следующей занятой ячейкой в первой строке является *ячейка(3,1)*. Поскольку соседняя ячейка слева пустая, мы присваиваем

7	7		9	3	3	
7				8	3	3
	6	3			3	3
6	6	5	5	3	3	3
		5		4	3	3
1	1			4	3	3
1		2	2		3	

a

7	7		3	3	3	
7				3	3	3
	3	3			3	3
3	3	3	3	3	3	3
		3		3	3	3
1	1			3	3	3
1		2	2		3	

б

Рис. 12.8. Перколяционная конфигурация на квадратной решетке $L = 7$. Координаты ячеек определяются относительно левого нижнего угла (1,1). *a*—показаны неправильные кластерные метки, присвоенные сначала программой *cluster*, в которой реализован модифицированный алгоритм Хошена—Копельмана. *б*—показаны правильные кластерные метки.

ей следующую допустимую кластерную метку, т.е. метку 2. Точно так же присваиваются кластерные метки остальным ячейкам первой строки. Затем мы возобновляем процедуру с *ячейке*(1,2) во второй строке. Поскольку она занята и ближайшая соседняя ячейка из первой строки имеет метку 1, мы присваиваем *ячейке*(1,2) метку 1. Продолжаем в том же духе, двигаясь слева направо по второй строке и проверяя занятость каждой ячейки. Если ячейка занята, то мы проверяем на занятость ее ближайших соседей в предыдущих строке и колонке. Если соседние ячейки пусты, то мы присваиваем ей следующую доступную кластерную метку. Если только одна из соседних ячеек занята, то присваиваем ей метку занятой ячейки. Например, *ячейке*(2,2) присваиваем метку 1, поскольку соседняя занятая *ячейка*(1,2) имеет метку 1.

Проблема возникает тогда, когда мы попадаем в занятую ячейку, в которой соединяются два кластера, и необходимо изменить кластерные метки. Этот случай впервые возникает в *ячейке*(6,2), у которой две соседние ячейки в предыдущих строке и столбце имеют метки 3 и 4 соответственно. Ясно, что *правильное* присваивание кластерной метки *ячейке*(6,2) заключается в выборе меньшей из меток 3 и 4. Следовательно, *ячейке*(6,2) присваивается метка 3, а метка 4 должна замеситься на метку 3. Однако, поскольку в дальнейшем могут проводиться другие изменения присвоенных меток, мы не будем проводить изменение меток до тех пор, пока не будет обследована вся решетка. Эта процедура изменения меток выполняется с помощью разделения *правильных* меток, таких, как 3, и *неправильных*, таких, как 4. Мы введем дополнительный массив *пр*(*i*), в котором разделяются правильные и непра-

ильные метки и учитываются их связи. Вернемся к конфигурации, показанной на рис. 12.8, чтобы объяснить, как используется этот массив. Прежде чем мы попадем в *ячейку*(6,2), метки от 1 до 4 считаются правильными и мы полагаем

$$pr(1) = 0, \quad pr(2) = 0, \quad pr(3) = 0, \quad pr(4) = 0.$$

Однако в *ячейке*(6,2), где метки 3 и 4 связаны, мы положим $pr(4) = 3$. Этот ненулевой элемент $pr(4)$ массива показывает, что метка 4 является неправильной, а численное значение $pr(4)$ показывает, что метка 4 связана с меткой 3. Заметим, что индекс массива всегда больше самого значения pr .

Приведенная выше процедура несовершенна. Что мы должны сделать, когда попадем в ячейку, в которой предшествующие отмаркированные соседи имеют неправильные метки? Рассмотрим, например, *ячейку*(5,4), у которой имеются занятые соседние ячейки с метками 5 и 4. Соблазнительно было бы присвоить *ячейке*(5,4) метку 4 и положить $pr(5) = 4$. Однако вместо присваивания ячейке минимальной метки ее соседей мы должны присвоить ей минимальную *правильную* метку одной из двух ближайших ячеек. К тому же, если две ближайшие ячейки имеют разные правильные метки, мы должны присвоить элементу массива pr максимальную правильную метку, равную минимальной правильной метке.

Приведенная выше версия алгоритма присваивания меток Хошена—Копельмана реализуется в программе `cluster` для квадратной решетки. В программе имеется также «меню», так что различные характеристики кластеров можно вычислять, не вызывая другие подпрограммы. В подпрограмме `assign` каждой ячейке присваивается случайное число и записывается в массив r . Эта подпрограмма вызывается в 1-м и 11-м пунктах меню. Занятость ячеек определяется в подпрограмме `occupy` и записывается в массив s . Присваивание кластерных меток производится в подпрограммах `cluster_label`, `newcluster`, `neighbors`, `label_min` и `proper`. Каждой ячейке присваиваются кластерные метки и записываются в одномерный массив cl , а связи кластерных меток хранятся в одномерном массиве pr . Подпрограмма `plot_label` изображает правильные или неправильные метки. Подпрограмма `plot_conf` рисует занятые ячейки, подпрограмма `plot_cluster` изображает координаты ячейки в кластере, правильный номер которой является аргументом подпрограммы.

После маркировки кластеров мы можем получить некоторые интересные нас геометрические характеристики. В подпрограмме `span` опреде-

ляется, содержит ли конфигурация вертикально соединяющий кластер, путем проверки наличия общих правильных меток в верхней и нижней строках. Если имеется соединяющий кластер, то его правильная метка выводится на экран, в противном случае переменная *ispan* остается равной 0.

В подпрограмме **compute_mass** определяется размер (число занятых ячеек) каждого кластера. Распределение кластеров n_s можно найти, подсчитывая количество кластеров размером n_s , и нормировать результаты, разделив их на L^2 .

Вероятность того, что занятая ячейка принадлежит соединяющему кластеру, определяется в подпрограмме **Pinfinity**. В этой подпрограмме для вычисления размера соединяющего кластера и полного числа занятых ячеек используются соответственно метка кластера, полученная в подпрограмме **span**, и распределение кластеров, полученное в подпрограмме **mass**. Отношение этих двух величин представляет собой P_∞ .

В подпрограмме **mean_size** для вычисления среднего размера кластера S в несоединяющих кластерах используются данные, полученные в подпрограмме **mass**. В подпрограмме **mean_size** значение $ns(i)$ представляет собой размер кластера с правильной меткой i . Для получения S мы суммируем по правильным меткам, а не по размеру s , как это сделано в определении (12.3) величины S . Убедитесь в том, что эти два метода вычисления S эквивалентны. Подчеркнем, что соединяющие кластеры не включаются в сумму.

Программа **cluster** не является самой эффективной реализацией алгоритма Хошена—Копельмана. Более эффективная версия программы на Фортране приводится у Стауффера (см. список литературы в конце главы). Хотя можно показать, что алгоритм Хошена—Копельмана является самым эффективным способом маркировки кластеров для двумерной решетки, не ясно, будет ли он самым эффективным для решеток большей размерности. Можете ли вы предложить другой метод идентификации кластеров?

```
PROGRAM cluster
DIM s(64,64), r(64,64), cl(64,64), np(1000), mass(1000)
RANDOMIZE
DO
  CLEAR
  CALL menu(choice)
  CLEAR
  SELECT CASE choice
  CASE 1          ! генерирование новой конфигурации
    INPUT prompt "длина стороны решетки = ": L
    INPUT prompt "вероятность занятия новой ячейки = ": p
    CALL assign(L,r) ! придание случайного числа каждой ячейке
    CALL occupy(L,p,s,r) ! проверка ячеек на занятость
  CASE 2          ! изменение p с сохранением случайных чисел
    INPUT prompt "новая вероятность занятия ячейки = " : p
    CALL occupy(L,p,s,r)
  CASE 3          ! рисование конфигурации
    CALL plot_setup(L)
    CALL draw_conf(L,s)
  CASE 4          ! определение кластерных меток
    CALL cluster_label(L,s,cl,np) ! присваивание меток кластерам
  CASE 5          ! список распределения кластеров
    CALL compute_mass(L,cl,np,mass,nmax) ! расчет массы кластеров
  CASE 6          ! рисование правильных меток кластеров
    CALL plot_setup(L)
    CALL print_label(L,cl,np,"Y")          ! рисуются правильные метки
  CASE 7          ! рисование меток кластеров до переписывания
    CALL plot_setup(L)
    CALL print_label(L,cl,np,"n")          ! рисуются неправильные метки
  CASE 8          ! определение протекания
    CALL span(L,cl,np,isper)
    IF isper <> 0 then
      PRINT "метка кластера - ",isper,"протекание вертикальное"
    ELSE
      PRINT "нет вертикального протекания"
    END IF
  END IF
```

```
CASE 9          ! вычисление радиуса циркуляции
  INPUT prompt "метка кластера = ": label
  CALL radius_gyration(L,cl,np,label)
CASE 10         ! рисование координат требуемого кластера
  INPUT prompt "метка кластера = ": label
  CALL plot_cluster(L,cl,np,label)
CASE 11         ! оценка вероятности первого протекания
  CALL first_span(L,p,s,r,cl,np)
CASE 12         ! вероятность занятия ячейки в соединяющем кластере
  CALL Pinfinity(mass,isper,nmax,prob)
  PRINT "вероятность ячейки в бесконечном кластере = "; prob
CASE 13         ! вычисление среднего размера кластеров
  CALL mean_size(mass,isper,nmax,mean)
  PRINT "средний размер кластера = ", mean
CASE 14
  STOP
CASE ELSE
END SELECT
PRINT "для выдачи на экран меню нажмите клавишу пробела"
DO
LOOP until key input
LOOP
END
```

SUB menu (choice)

PRINT "задайте один из следующих вариантов"

PRINT " 1. инициализация решетки"

PRINT " 2. изменение p , сохранение случайных чисел"

PRINT " 3. рисование занятых ячеек"

PRINT " 4. определение кластерных меток"

PRINT " 5. список распределения кластеров"

PRINT " 6. рисование правильных меток кластеров"

PRINT " 7. рисование пустых кластерных меток"

PRINT " 8. определение (вертикального) протекания"

PRINT " 9. вычисление радиуса циркуляции"

PRINT "10. рисование отдельного кластера"

PRINT "11. вычисление вероятности первого протекания"

PRINT "12. вычисление вероятности ячейки в соединяющем кластере"

PRINT "13. расчет среднего размера кластера S по конечным кластерам"

PRINT "14. выход"

PRINT

INPUT prompt "вариант? ": choice

END SUB

SUB assign (L, r(,))

! присваивание каждой ячейке случайного числа

FOR col = 1 to L

FOR row = 1 to L

LET r(col,row) = rnd

NEXT row

NEXT col

END SUB

SUB occupy (L, p, s(,), r(,))

FOR col = 1 to L

FOR row = 1 to L

IF r(col,row) < p then ! если $p < r(i,j)$, ячейка занимает

LET s(col,row) = -1 ! занятые ячейки - отрицательные

ELSE

LET s(col,row) = 0

END IF

NEXT row

NEXT col

END SUB

```

SUB cluster_label (L,s(,),cl(,),np())
  ! метки кластеров
  MAT cl = 0
  MAT np = 0
  LET ncluster = 0           ! метка кластера
  IF s(1,1) < 0 then CALL newcluster (ncluster,cl,np,1,1)
  FOR col = 2 to L           ! сначала метится нижняя строка
    IF s(col,1) < 0 then
      LET col_left = col - 1
      IF s(col_left,1) < 0 then
        LET cl(col,1) = cl(col_left,1)
      ELSE
        CALL newcluster (ncluster,cl,np,col,1)
      END IF
    END IF
  NEXT col
  FOR row = 2 to L           ! метки кластеров в остальных строках
    IF s(1,row) < 0 then     ! первый столбец анализируется отдельно
      LET row_down = row - 1 ! занятость ячейки предыдущей строки
      IF s(1,row_down) < 0 then
        LET cl(1,row) = cl(1,row_down)
      ELSE
        CALL newcluster (ncluster,cl,np,1,row)
      END IF
    END IF
    FOR col = 2 to L
      IF s(col,row) < 0 then
        LET row_down = row - 1
        LET col_left = col - 1
        IF cl(col,row_down) + cl(col_left,row) = 0 then
          CALL newcluster (ncluster,cl,np,col,row)
        ELSE
          CALL neighbor (s,cl,np,col,row)
        END IF
      END IF
    NEXT col
  NEXT row
END SUB

```

```
SUB newcluster ( ncluster, cl(, ), np(, ), col, row)
  LET ncluster = ncluster + 1
  LET cl(col, row) = ncluster
  LET np(ncluster) = 0                                ! правильная метка
END SUB

SUB neighbor ( s(, ), cl(, ), np(, ), col, row)      ! определение занятости соседей
  LET row_down = row - 1
  LET col_left = col - 1
  IF s(col, row_down) * s(col_left, row) > 0 then    ! оба соседа заняты
    CALL label_min(cl, np, col, row, col_left, row_down)
  EXIT SUB
END IF
IF cl(col, row_down) > 0 then
  LET cl(col, row) = cl(col, row_down)
  EXIT SUB
END IF
! сосед слева занят
LET cl(col, row) = cl(col_left, row)
END SUB

SUB label_min ( cl(, ), np(, ), col, row, col_left, row_down)
  ! оба соседа заняты, определение минимального кластерного числа
  IF cl(col_left, row) = cl(col, row_down) then
    ! у обоих соседей одинаковая кластерная метка
    LET cl(col, row) = cl(col_left, row)
  ELSE
    ! определение новой кластерной метки
    LET left = cl(col_left, row)
    LET down = cl(col, row_down)
    CALL proper(np, left)
    CALL proper(np, down)
    LET nmax = max(left, down)
    LET nmin = min(left, down)
    LET cl(col, row) = nmin
    IF nmin <> nmax then
      LET np(nmax) = nmin ! присваивание неправильной метке nmax = nmin
    END IF
  END IF
END SUB
```



```

SUB proper (np(), label)
DO
  IF np(label) = 0 then EXIT SUB
  LET label = np(label)
LOOP
END SUB

SUB plot_setup (L)
  ! окно для подпрограмм рисования
  LET aspect_ratio = 1.5      ! значение для компьютера Macintosh
  LET margin = 0.1*L
  LET mx = aspect_ratio*margin
  LET Lx = aspect_ratio*L
  SET window -mx, Lx + mx, -margin, L + margin
END SUB

SUB print_label (L, cl(, ), np(), proper$)
  LET size = 0.4
  FOR col = 1 to L
    LET x = col - 0.5
    FOR row = 1 to L
      LET y = row - 0.5
      LET label = cl(col, row)
      IF label > 0 then
        IF proper$ = "γ" then CALL proper(np, label)
        BOX CLEAR x - size, x + size, y - size, y + size
        BOX LINES x - size, x + size, y - size, y + size
        PLOT TEXT, at x - 0.25, y - 0.25: using$("###", label)
      END IF
    NEXT row
  NEXT col
END SUB

```

```
SUB span (L, cl(, ), np( ), ispan)
! проверка существования вертикального соединяющего кластера
LET ispan = 0
FOR side1 = 1 to L
  LET arg1 = side1
  LET arg2 = 1
  IF cl(arg1, arg2) > 0 then
    LET nbot = cl(arg1, arg2)
    CALL proper(np, nbot)      ! определяется правильность метки
    FOR side2 = 1 to L
      LET arg1 = side2
      LET arg2 = L
      IF cl(arg1, arg2) > 0 then
        LET ntop = cl(arg1, arg2)
        CALL proper(np, ntop)
        IF ntop = nbot then
          LET ispan = ntop
          EXIT SUB
        END IF
      END IF
    NEXT side2
  END IF
NEXT side1
END SUB

SUB compute_mass (L, cl(, ), np( ), mass( ), nmax)
! в элементе массива mass(i) - масса кластера с правильной меткой i
FOR n = 1 to nmax
  LET mass(n) = 0
NEXT n
LET nmax = 0
FOR row = 1 to L
  FOR col = 1 to L
    IF cl(col, row) > 0 then
      LET label = cl(col, row)
      CALL proper(np, label)      ! определение правильной метки
      IF label > nmax then LET nmax = label
      LET mass(label) = mass(label) + 1
    END IF
  
```

```

    NEXT col
NEXT row
PRINT "метка", "масса"
FOR i = 1 to nmax
    IF mass(i) > 0 then
        PRINT USING "*****": i, mass(i)
    END IF
NEXT i
PRINT
END SUB

SUB radius_gyration (L, cl(, ), np( ), label)
    FOR row = 1 to L
        FOR col = 1 to L
            IF cl(col, row) > 0 then
                LET label_temp = cl(col, row)
                CALL proper(np, label_temp)      ! определение правильной метки
                IF label_temp = label then        ! правильная метка
                    LET xcm = xcm + col
                    LET ycm = ycm + row
                    LET R2 = R2 + row*row + col*col
                    LET mass = mass + 1          ! число ячеек в кластере
                END IF
            END IF
        NEXT col
    NEXT row
    IF mass > 0 then
        LET radius = sqr((R2/mass) - ((xcm*xcm + ycm*ycm)/(mass*mass)))
        PRINT "радиус циркуляции кластера "; "метка"; label; " = "; radius
    ELSE
        PRINT "нет кластера с такой меткой"
    END IF
END SUB

```

```
SUB plot_cluster (L, cl(, ), np( ), label)
  CLEAR
  BOX LINES 0, L, 0, L
  LET size = 0.4      ! размер квадрата, рисуемого в каждом узле
  FOR col = 1 to L
    LET x = col - 0.5  ! с каждой ячейкой связан квадрат со стороной 1
    FOR row = 1 to L
      LET y = row - 0.5
      BOX LINES x - size, x + size, y - size, y + size
      LET label_temp = cl(col, row)
      IF label_temp > 0 then
        CALL proper(np, label_temp)
        IF label_temp = label then      ! занятая ячейка
          BOX AREA x - size, x + size, y - size, y + size
        END IF
      END IF
    NEXT row
  NEXT col
END SUB

SUB draw_conf (L, s(, ))
  BOX LINES 0, L, 0, L
  LET size = 0.4      ! размер квадрата, рисуемого в каждом узле
  FOR col = 1 to L
    LET x = col - 0.5  ! с каждой ячейкой связан квадрат со стороной 1
    FOR row = 1 to L
      LET y = row - 0.5
      PLOT POINTS: x, y      ! рисование узлов решетки
      IF s(col, row) = -1 then      ! занятая ячейка
        BOX AREA x - size, x + size, y - size, y + size
      ELSE
        BOX LINES x - size, x + size, y - size, y + size
      END IF
    NEXT row
  NEXT col
END SUB
```

```

SUB first_span (L,p,s(.),r(.),cl(.),np())
  DIM prob(100)
  INPUT prompt "длина стороны решетки = ": L
  INPUT prompt "начальное p = ": p0
  INPUT prompt "приращение p = ": dp
  INPUT prompt "число испытаний = ": ntrial
  FOR itrial = 1 to ntrial
    LET p = p0
    LET ip = 1
    LET ispan = 0          ! метка вертикально соединяющего кластера
    CALL assign(L,r)
    DO
      CALL occupy(L,p,s,r)      ! занятие дополнительных ячеек
      CALL cluster_label(L,s,cl,np)
      IF ispan = 0 then
        CALL span(L,cl,np,span)
        IF ispan <> 0 then LET prob(ip) = prob(ip) + 1
      END IF
      LET p = p + dp
      LET ip = ip + 1
    LOOP until ispan <> 0
  NEXT itrial
  LET p = p0
  LET ip = 1
  DO          ! нормировка вероятности первого протекания
    IF prob(ip) <> 0 then
      LET prob(ip) = prob(ip)/ntrial
      PRINT "p","вертикальная"
      PRINT p,prob(ip)
    END IF
    LET p = p + dp
    LET ip = ip + 1
  LOOP until p > 0.9
END SUB

```

```
SUB Pinfinity(mass(), ispan, nmax, prob)
! вероятность принадлежности занятой ячейки соединяющему кластеру
IF ispan = 0 then
    LET prob = 0
    EXIT SUB
END IF
LET sum = 0
FOR i = 1 to nmax
    LET sum = sum + mass(i)
NEXT i
LET prob = mass(ispan)/sum
END SUB

SUB mean_size(mass(), ispan, nmax, mean)
! определение среднего размера кластера S
LET sum = 0
LET sum2 = 0
FOR i = 1 to nmax
    IF i <> ispan then
        LET sum = sum + mass(i)
        LET sum2 = sum2 + mass(i)*mass(i)
    END IF
NEXT i
LET mean = sum2/sum
END SUB
```

Для более систематического изучения ячеечной перколяции в задаче 12.5 используется алгоритм маркировки кластеров Хошена—Копельмана. В разд. 12.4 мы будем применять конечномерный масштабный анализ, чтобы получить качественные результаты для относительно малых систем. Подпрограммы, которые изображают конфигурации и метки кластеров на экране, необходимо использовать только в задаче 12.5а.

ЗАДАЧА 12.5. Применение алгоритма маркировки кластеров

а. Запустите программу `cluster` и опишите несколько примеров метода маркировки кластеров. Подробно объясните, как работает алгоритм.

б. Вычислите $F(p)dp$ — вероятность *первого* протекания на решетке размером $L \times L$ в интервале от p до $p + dp$. Проведите минимум 100

испытаний для каждого значения L и постройте график изменения $F(p)$ с ростом L . При каком значении p в 50% испытаний уже имеются соединяющие кластеры для каждого правила протекания и для каждого размера решетки. Обозначьте это значение $p_c(L)$. Зависит ли p_c от L ? Как сильно $p_c(L)$ зависит от правила протекания? (Заметим, что вам необходимо модифицировать подпрограммы `span` и `first_span`, чтобы вычислить $F(p)$ для различных правил протекания.)

в. Модифицируйте программу `cluster` так, чтобы величина P_∞ усреднялась по меньшей мере по 100 испытаниям для каждого значения p . Вычислите P_∞ для значений $p = p_c$, $p = 0.65$, $p = 0.75$ и $p = 0.65$ при $L = 4, 16$ и 32 . Используйте либо оцененное значение $p_c(L)$, полученное в п. «а», либо наилучшую известную оценку $p_c = 0.5927$. Какова качественная зависимость P_∞ от p ? Является ли $P_\infty(p = p_c)$ возрастающей или убывающей функцией от L ? Не забудьте отбросить конфигурации, в которых нет соединяющего кластера.

г. Напишите подпрограмму для вычисления $n_s(p)$, используя массив $ns(i)$ (i — метка кластера). Рассмотрите случаи $p = p_c$ и $p = p_c \pm 0.1$ для значений $L = 4, 16$ и 32 и усредните по крайней мере по десяти испытаниям. Почему n_s является убывающей функцией s ? Когда n_s убывает более быстро: при $p = p_c$ или при $p \neq p_c$?

д. Вычислите средний размер кластера S при $p = p_c$ и $p = p_c \pm 0.1$ для значений $L = 4, 16$ и 32 . Усредните по крайней мере по десяти испытаниям. Какова качественная зависимость $S(p)$ от p ? Как $S(p = p_c)$ зависит от L ? Для значений $p < p_c$ отбросьте те конфигурации, которые содержат соединяющий кластер, а для $p > p_c$ отбросьте конфигурации, которые не содержат соединяющий кластер.

12.4. КРИТИЧЕСКИЕ ПОКАЗАТЕЛИ И КОНЕЧНОМЕРНОЕ МАСШТАБИРОВАНИЕ

Из повседневного опыта нам известны различные фазовые состояния вещества. Наиболее известным примером является вода, которая может существовать в виде пара, жидкости и льда. Известно, что вода может переходить из одной фазы в другую при определенных давлении и температуре, например переход из твердой фазы в жидкую происходит при атмосферном давлении и температуре 0°C . Такое изменение фаз пред-

ставляет собой пример *термодинамического фазового перехода*. Для большинства веществ существует *критическая точка*, т.е. при давлении и температуре выше конкретных значений температуры и давления невозможно различить газовую и жидкую фазы (Рейф).

Другим важным, но менее известным примером является существование критической точки в магнетиках при температуре Кюри T_c . Известно, что при низких температурах некоторые тела ведут себя как ферромагнетики, у которых в отсутствие внешнего поля намагниченность произвольна. Если повышать температуру, спонтанная намагниченность непрерывно убывает, обращаясь в нуль при «критической» температуре T_c . При температурах $T > T_c$ система становится парамагнетиком. В гл. 16 мы используем методы Монте-Карло для исследования поведения магнетиков в окрестностях критической точки.

Поскольку для понимания сути термодинамических фазовых переходов требуются солидные знания по статистической физике, интересно изучить фазовые переходы для явлений перколяции. Конечно, перколяция не является фазовым переходом в обычном смысле, поскольку для ее описания не используется температура. Однако в дальнейшем мы убедимся в том, что свойства *геометрических* фазовых переходов в задачах перколяции и термодинамических фазовых переходов качественно подобны. Следовательно, дальнейшее изучение фазовых переходов в задачах перколяции также может служить простым введением в теорию термодинамических фазовых переходов. Основной вывод, который мы сделаем, заключается в том, что вблизи точки фазового перехода поведение системы обусловлено наличием дальнедействующих корреляций.

Нам известно, что в окрестности порога перколяции поведение системы тесно связано с наличием больших, но конечных кластеров. Например, в задаче 12.5г мы нашли, что n_s убывает экспоненциально по мере роста s в случае $p < p_c$, а в случае $p > p_c$ n_s убывает более быстро с ростом s . Однако для $p = p_c$ зависимость n_s от s совершенно другая и убывание более медленное. Это особое поведение n_s в случае $p = p_c$ обусловлено наличием всех масштабов длины, например «бесконечный» кластер и кластеры всех размеров.

Более прямой способ наблюдения влияния длины связан с введением характерного линейного размера, или *средней длины связности* $\xi(p)$. Два возможных определения $\xi(p)$ изучаются в задаче 12.6.

ЗАДАЧА 12.6. Длина связности

а. Одно из определений величины ξ заключается в присваивании ей

значения *радиуса циркуляции* R_s . Запишем радиус циркуляции R_s для единственного кластера, состоящего из s ячеек, в виде

$$R_s^2 = \frac{1}{s} \sum_{i=1}^s (r_i - \bar{r})^2, \quad (12.5)$$

где

$$\bar{r} = \frac{1}{s} \sum_{i=1}^s r_i \quad (12.6)$$

и r_i — координата i -й ячейки в этом кластере. Величина \bar{r} является аналогом центра масс для кластера. Свяжем величину ξ с радиусом циркуляции *наибольшего* несоединяющего кластера, а не со взвешенным средним всех R_s по всем несоединяющим кластерам. Сгенерируйте перколяционную конфигурацию для данного значения p , вычислите R_s для наибольшего несоединяющего кластера и усредните R_s по нескольким конфигурациям. Рассмотрите значения p , принадлежащие интервалам $[p_c - 0.05, p_c - 0.01]$ и $[p_c + 0.01, p_c + 0.05]$, положив $p_c = 0.5927$ и шаг 0.01. Положите $L = 32$ и рассмотрите минимум 50 конфигураций для каждого значения p . Для $p < p_c$ отбросьте конфигурации, содержащие соединяющий кластер, а для $p > p_c$ отбросьте конфигурации, не содержащие соединяющий кластер. Постройте график функции ξ в зависимости от p и обсудите качественную зависимость от p . Является ли $\xi(p)$ монотонно возрастающей или убывающей функцией p для случаев $p < p_c$ и $p > p_c$?

*б. Свяжите длину связности с максимальным расстоянием между двумя узлами в наибольшем несоединяющем кластере (рис. 12.9). Модифицируйте программу `cluster` так, чтобы максимальное расстояние можно было вычислить и усреднить по нескольким испытаниям. Используйте конфигурацию из п. «а» и вычислите среднюю максималь-

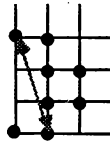


Рис. 12.9. Максимальное расстояние между двумя узлами показанного кластера равно $\sqrt{10}$. Координаты центра масс (\bar{x}, \bar{y}) и радиуса циркуляции кластера равны $\bar{x} = 1$, $\bar{y} = 1.5$, $R_s^2 = 14/8$.

ную длину между узлами в наибольшем несоединяющем кластере. Отождествите полученное значение этой длины с ξ , постройте график функции $\xi(p)$ и обсудите зависимость от p . Приводят ли оба определения ξ к одинаковой качественной зависимости?

На основании результатов, полученных в задаче 12.6, о поведении функции $\xi(p)$ на конечной решетке можно сделать вывод, что для больших значений L $\xi(p)$ — убывающая функция в диапазоне $p < p_c$ и возрастающая для $p > p_c$ (рис. 12.10). Кроме того, нам известно, что $\xi(p = p_c)$ приблизительно равна L и, следовательно, является расходящейся при $L \rightarrow \infty$. Качественное поведение функции ξ не зависит от определения этой функции и соответствует нашему физическому представлению о кластерах: по мере приближения p к p_c возрастает вероятность того, что два занятых узла находятся в одном кластере. Такое качественное рассмотрение наталкивает нас на мысль, что в пределе $L \rightarrow \infty$ $\xi(p)$ сингулярна в критической области $|p - p_c| \ll 1$. Количественно можно описать сингулярность $\xi(p)$, вводя критический показатель степени ν , определяемый соотношением

$$\xi(p) \sim |p - p_c|^{-\nu}. \quad (12.7)$$

Конечно, *a priori* нельзя сказать, почему расхожимость $\xi(p)$ имеет простой степенной характер. Возможно, вам захочется проанализировать свои данные о зависимости $\xi(p)$ от p и попытаться оценить показатель степени ν . Однако, поскольку ваши данные получены только для относительно малых решеток («эффекты конечных размеров») и ограниченного числа испытаний, ваша оценка ν будет очень приблизительной.

Как ведут себя в критической области другие величины, которые мы рассматривали, при $L \rightarrow \infty$? В соответствии с определением P_∞ (12.4), $P_\infty = 0$ при $p < p_c$ и возрастает при $p > p_c$. Мы предполагаем, что в

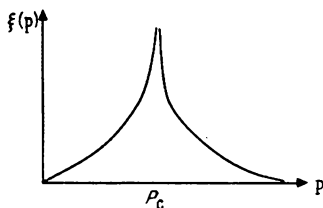


Рис. 12.10. Качественная зависимость длины средней связности $\xi(p)$ от p . Стремление $\xi(p)$ к бесконечности в критической области происходит по степенному закону с показателем ν [см. (12.7)].

критической области рост P_∞ характеризуется другим показателем степени β , определяемым соотношением

$$P_\infty \sim (p_c - p)^\beta. \quad (12.8)$$

В терминологии критических явлений P_∞ называется *параметром порядка* системы. Критический показатель степени β описывает стремление к нулю связности бесконечного кластера при пороговом значении для перколяции. Другой рассматриваемой величиной является средний размер кластера $S(p)$. Критическое поведение $S(p)$ можно описать следующим выражением:

$$S(p) \sim |p - p_c|^{-\gamma}, \quad (12.9)$$

которое определяет критический показатель. Известные критические показатели для перколяции приводятся в табл. 12.1. Для сравнения приведены аналогичные критические показатели для магнетиков.

Поскольку мы можем моделировать только конечные решетки, прямая подгонка измеряемых величин ξ , P_∞ и $S(p)$ по формулам (12.7)–(12.9) не будет давать хорошие оценки для соответствующих показателей степени. Главная проблема заключается в том, что мы не можем взять p достаточно близким к p_c без получения эффектов конечного размера. В

ТАБЛИЦА 12.1. Некоторые критические показатели степени для перколяционных и магнитных фазовых переходов в двумерном и трехмерном случаях. Рациональные числа представляют известные точные результаты. Данные для магнетиков относятся к модели Изинга (см. гл. 15.)

Величина	Зависимость	Показатель	$d = 2$	$d = 3$
Перколяция				
Параметр порядка	$P_\infty \sim (p - p_c)^\beta$	β	5/36	0.4
Средняя длина конечного кластера	$S(p) \sim p - p_c ^{-\gamma}$	γ	43/18	1.8
Длина корреляции	$\xi(p) \sim p - p_c ^{-\nu}$	ν	4/3	0.9
Число кластеров	$n_s \sim s^{-\tau}$	τ	187/91	2.2
Магнетизм				
Параметр порядка	$M(T) \sim (T - T_c)^\beta$	β	1/8	0.32
Восприимчивость	$\chi(T) \sim T - T_c ^{-\gamma}$	γ	7/4	1.24
Длина корреляции	$\xi(T) \sim T - T_c ^{-\nu}$	ν	1	0.63

противоположность этому значение $\xi(p)$ мало по сравнению с L для значений p , лежащих вдали от p_c , и на измеряемое значение ξ , а следовательно, и на другие физические величины не влияют конечные размеры решетки. Значит, для значений $p \ll p_c$ и $p \gg p_c$ свойства системы неотличимы от соответствующих свойств истинно макроскопической системы ($L \rightarrow \infty$). Однако если значение p близко к p_c , то $\xi(p)$ сравнима с L и поведение системы отличается от поведения макроскопической системы. В частности, на конечной решетке не могут происходить истинные фазовые переходы, описываемые расходящимися физическими величинами. Вместо этого ξ и S достигают конечного максимума при значении $p = p_c(L)$.

Эффекты конечного размера системы можно количественно описать с помощью следующих рассуждений. Рассмотрим, например, предполагаемое критическое поведение величины P_∞ , описываемой (12.8). До тех пор пока ξ много меньше, чем L , предполагается, что степенной закон поведения (12.8) выполняется. Однако если величина ξ сравнима с L , то ξ не может существенно изменяться и выражение (12.8) становится неприменимым. Эти качественные изменения поведения P_∞ и других физических величин будут происходить для зависимости

$$\xi(p) \sim L \sim |p - p_c|^{-\nu}. \quad (12.10)$$

Заметим, что выражение (12.10) можно обратить и записать

$$|p - p_c| \sim L^{-1/\nu}. \quad (12.11)$$

Следовательно, если ξ и L приблизительно одинаковы, то можно заметить (12.8) на следующее соотношение:

$$P_\infty(p = p_c) \sim L^{-\beta/\nu} \quad (L \rightarrow \infty). \quad (12.12)$$

Соотношение (12.12) между P_∞ и L при $p = p_c$ соответствует тому факту, что фазовый переход определяется только для бесконечных систем.

Одно из применений соотношения (12.12) состоит в том, что с его помощью можно определить критические показатели. Такой метод анализа известен как *конечномерное масштабирование* и является важным при изучении критических показателей. Предположим, что мы сгенерировали перколяционные конфигурации при $p = p_c$ для различных значений L и исследуем зависимость P_∞ от L . Если значение L достаточно велико, то можно воспользоваться соотношением (12.12) для оценки отношения

β/ν . Аналогичное исследование можно провести для $S(p)$ и других интересующих нас физических величин. Мы пользуемся этим методом в задаче 12.7.

*ЗАДАЧА 12.7. Конечномерный анализ критических показателей

а. Воспользуйтесь программой `cluster` для вычисления P_∞ при $p = p_c$, проводя по меньшей мере 100 испытаний. Рассмотрите случаи $L = 10, 20, 40$ и 60 . Усредняйте только по конфигурациям, которые содержат соединяющий кластер. Хотя наши качественные рассуждения о соотношении (12.12) не определяют, должны ли мы выбрать p равным $p_c(L)$ или $p_c(L \rightarrow \infty)$, наилучшие результаты получаются с использованием последнего значения $p_c = 0.5927$ на квадратной решетке. Постройте график зависимости $\ln P_\infty$ от $\ln L$ и оцените отношение β/ν .

б. Воспользуйтесь конечномерным масштабированием для установления зависимости среднего размера кластера от L при $p = p_c$. Модифицируйте программу `cluster` таким образом, чтобы S усреднялась по тем же конфигурациям, что и в п. «а». Не забудьте, что S является средним числом ячеек в несоединяющих кластерах.

в. Найдите зависимость массы (числа частиц) M в соединяющем кластере при $p = p_c$ от L . Используйте такие же конфигурации, что и в п. «а». Определите показатель степени с помощью графика зависимости $\ln P_\infty$ от $\ln L$. Значение этого показателя называется фрактальной размерностью кластера и обсуждается в гл. 13.

В разд. 12.2 мы нашли, что численное значение порога перколяции p_c зависит от симметрии и размеров решетки, например $p_c \approx 0.5927$ для квадратной решетки и $p_c = 1/2$ для треугольной решетки. Удивительное свойство степенных зависимостей, приведенных в табл. 12.1, заключается в том, что значения критических показателей не зависят от симметрии решетки и даже от наличия самой решетки, например они равны показателям для модели непрерывной перколяции, которая обсуждалась в задаче 12.3. Более того, не надо делать различий между этими показателями для ячеечной и цепной перколяций. Используя терминологию теории фазовых переходов, мы скажем, что непрерывная, ячеечная и цепная перколяции принадлежат одному классу универсальности и их критические показатели равны.

Другой важной идеей в области критических явлений считается существование зависимостей между критическими показателями. Примером такого закона масштабирования будет соотношение

$$2\beta + \gamma = \nu d, \quad (12.13)$$

где d — размерность решетки. Более детальное рассмотрение конечномерных и масштабных законов можно найти в списке литературы, который приводится в конце раздела.

12.5. РЕНОРМ-ГРУППА

В разд. 12.4 мы воспользовались свойствами связанных между собой величин в задачах перколяции при различных масштабах длины для вычисления значений критических показателей. Идею изучения некоторых физических величин в окрестности критических точек на различных масштабах длины можно применять не только для конечномерного масштабирования, но и взять за основу в методе *ренорм-группы*, который, вероятно, является одним из важнейших новых методов теоретической физики за последнее двадцатилетие. Первое объяснение критических явлений с помощью метода ренорм-группы было опубликовано К. Ж. Вильсоном в 1971 г. В 1981 г. автор был удостоен Нобелевской премии по физике за вклад в разработку метода ренорм-группы. Хотя впервые этот метод был использован в теории термодинамических фазовых переходов, проще познакомиться с ним на примере задач перколяции. Мы увидим, что этот прямой метод получения критических показателей в соединении с методами Монте-Карло часто представляет собой гораздо более мощное средство, чем обычные методы Монте-Карло.

Для ознакомления с этим методом рассмотрим фотографию перколяционной конфигурации, сгенерированной для значения $p = p_0 < p_c$. Что мы увидим, если будем рассматривать фотографию со все большего расстояния? Убедитесь в том, что рассматривание фото с большого расстояния не позволит вам увидеть кластеры, состоящие из одной ячейки, и различить близлежащие ячейки. К тому же области между большими кластерами и узкие переемы, соединяющие крупные «кляксы», не будут различаться на фотографии с такого расстояния. Следовательно, при $p_0 < p_c$ расположенная вдалеке фотография будет похожа на перколяционную конфигурацию, сгенерированную для значения p_1 такого, что $p_1 < p_0$. Кроме того, длина связности $\xi(p_1)$ оставшихся кластеров бу-

дет меньше, чем $\xi(p_0)$. Если мы отойдем еще дальше, то новые кластеры на фотографии будут казаться еще меньше и соответствовать значению $p = p_2$, причем $p_2 < p_1$. Обычно мы не в состоянии различить любые кластеры и фотография будет казаться такой, как будто она сделана для тривиальной неподвижной точки $p = 0$.

Обсудим что мы могли бы увидеть в случае $p_0 > p_c$. При внимательном изучении фотографии мы могли бы различить только малые области незанятых ячеек. По мере удаления фотографии эти пустоты становятся менее различимыми и конфигурация будет выглядеть все более однородной. Следовательно, фото будет выглядеть как конфигурация сгенерированная для значения $p = p_1$ с $p_1 > p_0$ и $\xi(p_1) < \xi(p_0)$. Мы заключаем, что по мере удаления фотографии она будет казаться такой, как будто она сделана при значении, равном другой тривиальной устойчивой точке, $p = 1$.

Что произойдет в случае $p_0 = p_c$? Нам известно, что при значении параметра, равном пороговому, имеются все масштабы длины, и не играет роли, какую длину мы используем для изучения системы. Таким образом, фото будет выглядеть одинаково (хотя и менее чисто) независимо от расстояния, с которого мы его рассматриваем. В этом смысле p_c является особой *нетривиальной* устойчивой точкой.

Рассмотрим операционный метод использования компьютера для изменения конфигурации, которое аналогично удалению фотографии. Наше изложение в точности соответствует статье Рейнольдса и др. (см. список литературы). Рассмотрим квадратную решетку, которая разбита на *клетки*, или *блоки*, покрывающие решетку (рис. 12.11). Если мы посмотрим на решетку издалека, то узлы поглощаются клетками и образуют новый суперузел или «ренормированный» узел, тогда новая решетка обладает той же симметрией, что и исходная решетка. Однако замена узлов новыми изменяет масштаб длины — все расстояния уменьшаются в b раз, где b — линейный размер клетки. Таким образом, эффект «ренормализации» заключается в замене каждой клетки единственным ренормированным узлом и перемасштабировании длины связности для ренормированной решетки в b раз.

Как установить, будут ли ренормированные узлы заняты или свободны? Поскольку мы хотим сохранить основные особенности исходной решетки, а значит, ее связность мы предположим, что ренормированный узел занимается, если исходная группа узлов связывает эту клетку. Будем пользоваться для удобства правилом вертикального связывания. Эффект выполнения масштабного преобразования типичных перколяционных конфигураций для значений p больше и меньше p_c иллюстрируется

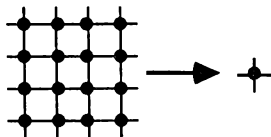


Рис. 12.11. Пример клетки с $b = 4$, используемой для квадратной решетки. Эта клетка, содержащая b^2 узлов, после ренормализации преобразуется в единственный узел.

соответственно на рис. 12.12а и 12.12б. В обоих случаях преобразования ренормализации удаляют систему от состояния при p_c . Видно, что для $p = 0.7$ эти преобразования возвращают систему к случаю с $p = 1$. Для $p = 0.5$ имеется тенденция к возвращению системы обратно к случаю с $p = 0$. Поскольку мы начали на конечной решетке, невозможно продолжать преобразование перенормировки бесконечно.

Программа `gc` в соединении с вашей зрительной интуицией генерирует конфигурации, типа приведенной на рис. 12.12, и выполняет простой вариант метода ренорм-группы. Программа делит экран на четыре окна и рисует три ренормированные решетки в окнах 2, 3, 4.

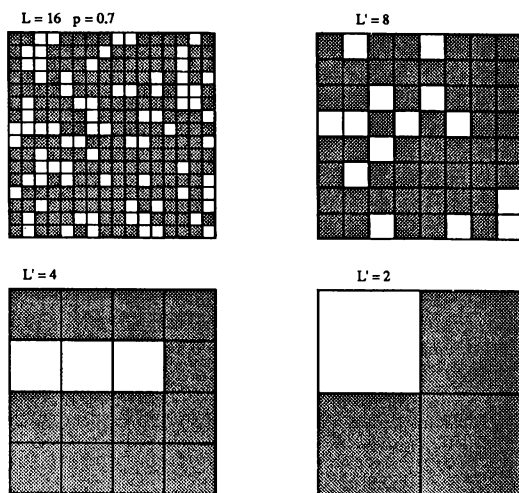


Рис. 12.12а. Перколяционная конфигурация, сгенерированная для значения $p = 0.7$. Исходная конфигурация ренормировалась три раза с помощью программы `gc`, которая преобразует четырехъячеечные клетки в одну новую. Что получилось бы после еще одного преобразования?

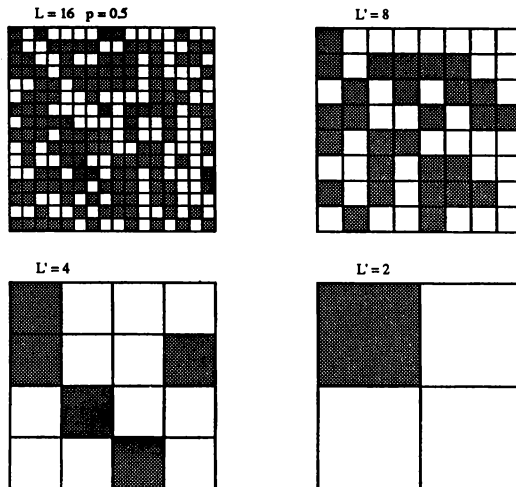


Рис. 12.126. Перколяционная конфигурация, сгенерированная для случая $p = 0.5$. Исходная конфигурация ренормировалась три раза с помощью программы `rg`, которая преобразует четырехъячейчные клетки в одну новую. Что получилось бы после еще одного преобразования?

```

PROGRAM rg
DIM r(16,16)
CALL initial(*1,*2,*3,*4,L,b,r) ! присваивание случайных чисел ячейкам
CALL configuration(*1,*2,*3,*4,L,b,r)
END

SUB initial(*1,*2,*3,*4,L,b,r(,))
  RANDOMIZE
  LET L = 16
  LET b = 2
  FOR row = 1 to L
    FOR col = 1 to L
      LET r(col,row) = rnd ! присваиваем каждой ячейке случайное число
    NEXT col
  NEXT row
  OPEN #1: screen 0,0.5,0.5,1
  CALL lattice(*1,L) ! рисование исходной решетки в окне 1
  OPEN #2: screen 0.5,1,0.5,1

```

```

CALL lattice(#2,L/b)
OPEN #3: screen 0,0.5,0,0.5
CALL lattice(#3,L/(b*b))
OPEN #4: screen 0.5,1,0,.5
CALL lattice(#4,L/(b*b*b))
END SUB

SUB lattice (#1,L)
  LET aspect_ratio = 1.5
  LET margin = 0.2*L
  LET mx = aspect_ratio*margin
  LET bx = aspect_ratio*L
  SET window -mx,bx,-margin,L + margin
  BOX LINES 0,L,0,L
  FOR row = 1 to L
    FOR col = 1 to L
      PLOT POINTS: col - 0.5,row - 0.5
    NEXT col
  NEXT row
END SUB

SUB configuration (#1,#2,#3,#4,L,b,r(,))
  DIM s(16,16),s1(8,8),s2(4,4),s3(2,2)
  DO while p <=1
    WINDOW #1
    SET cursor 1,1
    PRINT "L ="; L;
    SET cursor 1,14
    PRINT " " " ! стирание предыдущего значения p
    SET cursor 1,10
    INPUT prompt "p = ": p
    FOR row = 1 to L
      FOR col = 1 to L
        IF r(col,row) < p and s(col,row) <> 1 then
          CALL occupy(#1,col,row)
          LET s(col,row) = 1
        END IF
      NEXT col
    NEXT row
  
```

```

CALL block(#2,L/b,s,s1)
CALL block(#3,L/(b*b),s1,s2)
CALL block(#4,L/(b*b*b),s2,s3)
LOOP
END SUB

SUB block (#1,L,w(,),wr(,))
WINDOW #1
FOR row = 1 to L
  LET y = 2*row - 1
  FOR col = 1 to L
    LET x = 2*col - 1
    ! при вертикальном связывании ренормированная ячейка занимается
    IF w(x,y)*w(x,y + 1) = 1 then LET wr(col,row) = 1
    IF w(x + 1,y)*w(x + 1,y + 1) = 1 then LET wr(col,row) = 1
    IF wr(col,row) = 1 then CALL occupy(#1,col,row)
  NEXT col
NEXT row
SET cursor 1,1
PRINT "L' ="; L;
END SUB

SUB occupy (#1,col,row)
LET x = 0.5
LET d = 0.4
BOX AREA col - x - d, col - x + d, row - x - d, row -x + d
END SUB

```

ЗАДАЧА 12.8. Наглядная ренорм-группа

Воспользуйтесь программой **rg** с $L = 32$ и $b = 2$ и оцените величину порога перколяции. Например, покажите, что для небольших значений p , например $p \approx 0.4$, ренормированная решетка обычно преобразуется в несоединяющий кластер. Что произойдет в случае больших p , например $p \approx 0.8$? Как можно использовать свойства ренормированных решеток для вычисления p_c ?

Хотя визуальное выполнение метода ренорм-группы позволяет нам грубо оценить p_c , оно не дает возможности оценить критические показатели. В дальнейшем мы проведем анализ, основанный на методе ре-

норм-группы, который позволит нам получить p_c и критический показатель ν , связанный с длиной связности. Этот анализ близко следует методу, предложенному Рейнольдсом и др. (см. список литературы).

Реализация метода ренорм-группы распадается на две части: усреднение по всем основным переменным и точное определение параметров, определяющих ренормированную конфигурацию. Мы применим такое же усреднение, как и ранее, т.е. сгруппируем b^d ячеек внутри блока размером b и заменим эту клетку единственной, которая изображается или нет в зависимости от того связывали или нет исходные ячейки эту клетку. Второй шаг заключается в определении параметров, которые характеризуют новую конфигурацию после усреднения. Мы допустим, что каждая клетка независима от остальных и характеризуется только значением p' — вероятностью того, что клетка занята. Поскольку преобразование ренормализации, связывающее между собой p' и p , должно отражать тот факт, что основным свойством перколяции является связность, т.е. наличие соединяющего пути; мы считаем клетку занятой, если она содержит множество ячеек, которые «пересекают» эту клетку. Следовательно, если ячейки занимаются с вероятностью p , то клетки занимаются с вероятностью p' , где p' определяется *рекурсивным соотношением* или *преобразованием перенормировки* вида

$$p' = R(p). \quad (12.14)$$

Функция $R(p)$ — полная вероятность того, что ячейки образуют соединяющий путь. Приведем пример, проясняющий формальную запись (12.14). На рис. 12.13 показаны семь соединяющих конфигураций для случая клетки с $b = 2$. Вероятность p' того, что ренормированная ячейка будет занята, равна сумме вероятностей всех возможных вариантов:

$$p' = R(p) = p^4 + 4p^3(1 - p) + 2p^2(1 - p)^2. \quad (12.15)$$

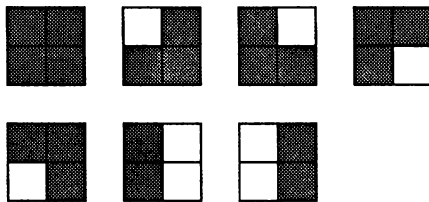


Рис. 12.13. Семь (вертикально) соединяющих конфигураций для клетки с $b = 2$.

Заметим, что вообще вероятность занятия ренормированных ячеек p' отличается от вероятности занятия исходных ячеек p . Например, предположим, что начинаем со значения $p = p_0 = 0.5$. После выполнения одного преобразования ренормализации значение p , получаемое по формуле (12.15), равно $p_1 = R(p_0 = 0.5) = 0.44$. Если выполнить второе преобразование, получим $p_2 = R(p_1) = 0.35$. Нетрудно сделать вывод, что дальнейшее применение преобразования будет приближать систему к неподвижной точке $p = 0$. Точно так же, начиная с $p = 0.7$, после последовательного применения преобразований попадем в неподвижную точку $p = 1$. Для нахождения нетривиальных устойчивых точек, соответствующих критическому порогу p_c , нам необходимо найти конкретное значение p такое, что

$$p^* = R(p^*). \quad (12.16)$$

Из рекурсивного соотношения (12.15) мы найдем, что решение уравнения четвертой степени относительно p^* имеет два тривиальных значения неподвижных точек $p^* = 0$ и $p^* = 1$ и нетривиальную неподвижную точку $p^* = 0.61804$, которую мы свяжем с p_c . Это вычисленное значение p^* для случая с $b = 2$ нужно сравнить с наилучшей известной оценкой $p_c = 0.5927$.

Для вычисления критического показателя ν с помощью преобразования перенормировки мы повторим, что на ренормированной решетке все длины уменьшаются в b раз по сравнению с длинами исходной решетки. Следовательно, длина связности преобразуется в соответствии с

$$\xi' = \xi/b. \quad (12.17)$$

Поскольку $\xi(p) = \text{const } |p - p_c|^{-\nu}$ для $p \sim p_c$ и p_c соответствует p^* , мы имеем

$$|p' - p_c|^{-\nu} = b^{-1} |p - p^*|^{-\nu}. \quad (12.18)$$

Для нахождения связи между p' и p вблизи p_c разложим ренормализационное преобразование (12.14) в ряд вблизи p^* и получим в первом порядке

$$p' - p^* = R(p^*) \approx \lambda(p - p^*) \quad (12.19)$$

где

$$\lambda = \left. \frac{dR}{dp} \right|_{p=p^*} . \quad (12.20)$$

Необходимо проделать несложные алгебраические выкладки для получения явного выражения для ν . Возведем сначала (12.19) в степень $-\nu$ и запишем

$$|p^* - p_c|^{-\nu} = \lambda^{-\nu} |p - p^*|^{-\nu} . \quad (12.21)$$

Затем сравним (12.21) и (12.18) и получим

$$b^{-1} = \lambda^{-\nu} . \quad (12.22)$$

Наконец, прологарифмируем обе части (12.22) и получим требуемое соотношение для критического показателя ν :

$$\nu = \frac{\log b}{\log \lambda} . \quad (12.23)$$

В качестве примера вычислим λ для $b = 2$, записав (12.15) в виде $R(p) = -p^2 + 2p^2$. Производная $R(p)$ по p дает $\lambda = 4p(1 - p^2) = 1.5279$ при $p = p^* = 0.61804$. Затем используем соотношение (12.23) и получим

$$\nu = \log 2 / \log 1.5279 = 1.635... . \quad (12.24)$$

Сравнение (12.24) с точным значением $\nu = 4/3$ для двумерного случая показывает удивительное соответствие для таких простых вычислений. Какое заключение мы могли бы сделать, если бы измеряли непосредственно $\xi(p)$ на решетке 2×2 ? Наши вычисления неконтролируемы, поскольку мы не имеем *a priori* оценок точности наших вычислений. В чем заключается суть наших приближений? Нашим основным предположением была независимость занятия каждой ячейки от состояния других. Это предположение верно для исходной ячейки, но после ренормализации мы теряем некоторые исходные соединяющие пути и приобретаем соединяющие пути, отсутствующие в исходной решетке. Пример такой «пограничной» проблемы приведен на рис. 12.14. Поскольку такой поверхностный эффект становится менее вероятным с увеличением размера клеток, один из способов улучшения вычислений методом ренорм-группы заключается в рассмотрении больших клеток. Мы рассматриваем вычисления для $b = 3$ в задаче 12.9.

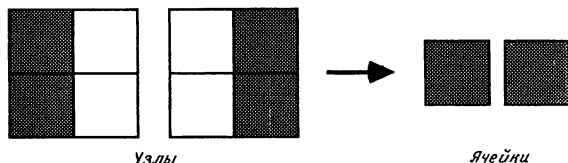


Рис. 12.14. Пример проблемы границ между клетками. Эти две клетки не связаны между собой в исходной конфигурации на ячейечном уровне, но связаны на клеточном.

ЗАДАЧА 12.9. Пространственный метод ренорм-группы для малых клеток

а. Получите соединяющую конфигурацию для $b = 2$ -связанной клетки, предположив, что клетка связывается, если соединяющий путь существует для вертикального или горизонтального направления. Получите рекурсивную зависимость и решите ее для нахождения неподвижной точки значения p^* . Простейшим способом решения (12.16) является метод проб и ошибок. Другой метод заключается в построении функции $R(p) - p$ и нахождении значения p , при котором эта функция пересекает горизонтальную ось. Как соотносятся значения p^* и ν со значениями, полученными при использовании правила вертикального протекания?

б. Переберите возможные соединяющие конфигурации для клетки размером 3×3 , вычислите вероятность каждой конфигурации и получите преобразование перенормировки $R(p)$. Предположите, что клетка занимает, если кластер пересекает клетку вертикально и если кластер пересекает клетку и вертикально, и горизонтально. Затем решите рекурсивное соотношение (12.16) для p^* . Если значение p^* известно, то наклон λ и показатель ν можно вычислить аналитически. Вычислите $p^*(b = 3)$ и $\nu(b = 3)$ для этих двух правил связывания. Близки ли ваши значения p^* и ν к известным результатам для случая $b = 2$?

Возможно улучшить значения $p_c(b)$ и $\nu(p)$ перебором соединяющих кластеров для достаточно больших значений b . Однако, поскольку имеется 2^{b^2} возможных конфигураций для клетки размером $b \times b$, число которых быстро растет с увеличением b , полный перебор невыполним для $b > 5$, и мы должны использовать методы Монте-Карло. Два связанных между собой метода Монте-Карло обсуждаются в задаче 12.10.

***ЗАДАЧА 12.10. Метод Монте-Карло для ренорм-группы**

а. Один из способов улучшения оценки $R(p)$, полной вероятности всех соединяющих кластеров, можно понять, записывая $R(p)$ в виде

$$R(p) = \sum_{n=1}^N \binom{N}{n} p^n q^{(N-n)} S(n), \quad (12.25)$$

где $N = b^2$. Биномиальный коэффициент $\binom{N}{n}$ представляет собой число возможных конфигураций n занятых ячеек и $(N - n)$ пустых. Величина $S(n)$ представляет собой вероятность того, что конфигурация с n занятыми ячейками протекает через клетку. Сравнение (12.15) и (12.25) показывает, что для $b = 2$ и вертикального протекания $S(1) = 0$, $S(2) = 2/6$, $S(3) = 1$ и $S(4) = 1$. Чему равны значения $S(n)$ для $b = 3$?

Поскольку $S(n)$ — вероятность, мы можем оценить ее простыми методами Монте-Карло. Простейший способ провести выборочное испытание $S(n)$ заключается в добавлении частицы произвольным образом в незанятый узел и проверке, существует ли соединяющий путь. Если соединяющий путь существует после добавления s частиц, то $S(n) = S(n) + 1$ для $n \geq s$ и начинается новое испытание. После приемлемого числа испытаний значения $S(n)$ можно нормировать. Конечно, эту процедуру можно сделать более эффективной, проверяя соединяющий кластер только после добавления полного числа частиц вблизи $s \sim p^* N$ и проверяя соединяющий кластер после добавления нескольких частиц.

Напишите программу выборочного испытания $S(n)$ методом Монте-Карло. Запомните положения незанятых узлов в массиве. Для контроля вашей программы сначала произведите выборку $S(n)$ для $b = 2$ и $b = 3$, сравните свои результаты с точными для $S(n)$. Рассмотрите большие значения b и вычислите $S(n)$ для значений $b = 5, 8, 16$ и 32. Для значений $b \cong 16$ функцию $R(p)$ можно найти, используя (12.25) и гауссову аппроксимацию

$$P_N(p) = \binom{N}{n} p^n q^{(N-n)} S(n) \approx (2\pi Npq)^{-1/2} \exp(-(n-pN)^2/2Npq). \quad (12.26)$$

Поскольку $P_N(p)$ имеет резкий пик для больших значений b , необхо-

димо проводить выборку $S(n)$ только вблизи $n = p^*N$.

б. Заметим, что в п. «а» варьировалось число частиц, а не вероятность занятия ячейки p . Эквивалентная процедура метода Монте-Карло заключается в варьировании p и производстве выборки $F(p)dp$ — вероятности появления *первого* соединяющего кластера в клетке размерами $b \times b$ в диапазоне от p до $p + dp$. Поскольку ренорм-групповое преобразование определяет p' как *полную* вероятность соединения при значении p , величину p' можно интерпретировать как *интегральную функцию распределения*, а значит, она связана с $F(p)$ соотношением

$$p' = R(p) = \int_0^p F(p)dp. \quad (12.27)$$

Выборка $F(p)$ для конечной длины dp подразумевает, что интеграл в выражении (12.27) переходит в конечную сумму. Поскольку $\lambda = dR(p=p^*)/dp$, мы имеем $\lambda = F(p^*)$. Простейший способ вычислить λ заключается в том, чтобы положить $\lambda = F(p_{\max})$, где p_{\max} — значение p , при котором $F(p)$ достигает максимума. Как соотносятся между собой величины p_c и p_{\max} ? Вычислите $p_c(b)$ и $\nu(b)$ для значений $b = 5, 8, 16$ и 32 . Как соотносятся эти результаты с теми, которые получены в п. «а»? В каком из методов ошибки оценок величин p_c и ν меньше?

в. Представляется возможным проэкстраполировать значения $p_c(b)$ и $\nu(b)$ при $b \rightarrow \infty$. Воспользуйтесь асимптотическими соотношениями

$$\nu(b)^{-1} \approx \nu^{-1} - c_1/\ln b \quad (12.28a)$$

и

$$p^*(b) \approx p_c - c_2 b^{-\nu} \quad (12.28b)$$

для оценки экстраполированных значений ν и p_c . Заметим, что необходимо рассмотреть клетки размером порядка $b \approx 500$ и провести более тонкое исследование $\nu(b)$ и $p^*(b)$ для получения результатов, отвечающих точному значению $\nu = 4/3$ и наилучшей известной оценке $p_c = 0.5927$.

ЛИТЕРАТУРА

C. Domb, E. Stoll, T. Schneider, Percolation clusters, *Contemp. Phys.* **21**, 577 (1980). В этой статье обсуждается природа перколяционных переходов с использованием иллюстраций из фильма о процессе перколяции, промоделированном методом Монте-Карло.

J. W. Essam, Percolation theory, *Reports on Progress in Physics* **53**, 883 (1980). Обзорная статья, ориентированная на математиков.

J. P. Fitzpatrick, R. B. Malt, F. Spaepen, Percolation theory of the conductivity of random close packed mixtures of hard spheres, *Phys. Letts.* **A47**, 207 (1974). Авторы описывают демонстрационный эксперимент, выполняемый студентами-физиками первого года обучения в Гарвардском университете.

E. T. Gawlinski, H. E. Stanley, Continuum percolation in two dimensions: Monte Carlo tests of scaling and universality for non-interacting discs, *J. Phys. A: Math. Gen.* **14**, L291 (1981). В этой работе авторы показали, что погрешность метода Монте-Карло для критических показателей в случае непрерывной перколяции та же, что и для двумерной решетки.

J. Hoshen, R. Kopelman, Percolation and cluster distribution. 1. Cluster multiple labeling technique and critical concentration algorithm, *Phys. Rev.* **B14**, 3488 (1976). Оригинальная статья, в которой описывается эффективный алгоритм маркировки кластеров.

R. Mehr, T. Grossman, N. Kristianpoller, Y. Gefen, Simple percolation experiment in two dimensions, *Am. J. Phys.* **54**, 271 (1986). Предложен простой опыт для начинающих экспериментаторов.

P. J. Reynolds, H. E. Stanley, W. Klein, Large-cell Monte Carlo renormalization group for percolation, *Phys. Rev.* **B21**, 1223 (1980). Очень ясно написанная статья. Наше рассмотрение ренорм-группы в разд. 12.5 основано на этой статье.

D. Stauffer, Percolation clusters as teaching aid for Monte Carlo simulation and critical exponents, *Am. J. Phys.* **45**, 1001 (1977).

D. Stauffer, Scaling theory of percolation clusters, *Physics Reports* **54**, 1 (1979). Важная обзорная статья.

D. Stauffer, *Introduction to Percolation Theory*, Taylor & Francis, 1985. Отличная небольшая книга одного из ведущих специалистов в этой области. В приложении приводится эффективная реализация алгоритма Хошена—Копельмана на Фортране.

B. P. Watson, P. L. Leath, Conductivity in the two-dimensional-site percolation problem, *Phys. Rev.* **B9**, 4893 (1974). Исследовательская ра-

работа, посвященная проводимости проволоочной сетки.

K. G. Wilson, Problems in physics with many scales of length, Sci. Am. **241**, 158 (1979). Доступная статья, посвященная методу ренорм-группы и его приложениям в физике частиц и конденсированного состояния. См. там же. The renormalization group and critical phenomena, Rev. Mod. Phys. **55**, 583 (1983). Последняя статья является текстом лекции Вильсона, прочитанной по случаю вручения Нобелевской премии по физике. В этой лекции он заявил, что он «... пришел к заключению, что очень полезно потребовать, чтобы корректно сформулированная теория поля решалась компьютером тем же способом, что обыкновенные дифференциальные уравнения...».

R. Zallen, The Physics of Amorphous Solids, Wiley-Interscience, 1983. В гл. 4 этой развлекательной и информативной книжки обсуждается много приложений понятий перколяции к реальным системам.

ДОПОЛНИТЕЛЬНАЯ ЛИТЕРАТУРА

Эфрос А. Л., Физика и геометрия беспорядка.—М.: Наука, 1982. Простая и увлекательная книжка, в которой изложены многие идеи теории перколяции.

Шкловский Б. И., Эфрос А. Л., Электронные свойства легированных полупроводников.—М.: Наука, 1979. Гл. 5 посвящена перколяции. Рассматриваются различные типы перколяции на решетках и метод критических показателей.

Методы Монте-Карло в статистической физике/ Под ред. К. Биндера. М: Мир, 1982. Во введении и гл. 5 рассматриваются метод Хошена—Копельмана и техника ренормализации с использованием метода Монте-Карло. Рассчитана на подготовленного читателя.

ФРАКТАЛЫ, МОДЕЛИ КИНЕТИЧЕСКОГО РОСТА И КЛЕТОЧНЫЕ АВТОМАТЫ

13

В этой главе мы знакомимся с понятием фрактальной размерности и обсуждаем модели перколяции и кинетического роста. Рассматриваются также некоторые модели клеточных автоматов.

13.1. ФРАКТАЛЬНАЯ РАЗМЕРНОСТЬ

Одним из наиболее интересных геометрических свойств объектов является их форма. В качестве примера на рис. 13.1 показан перколяционный кластер, полученный для значения, равного пороговому. Хотя описание соединяющего кластера, основанное на зрительном восприятии, субъективно, о таких кластерах говорят как о разветвленных, воздушных, разреженных и волокнистых. В противоположность этому перколяционный кластер никак нельзя охарактеризовать как плотный или заполненный.

В последние годы Мандельбротом и другими авторами (см. список литературы) для описания таких разветвленных объектов была разработана новая *фрактальная* геометрия. В качестве количественной меры структурности этих объектов выступает *фрактальная размерность* d_f . Для определения d_f вспомним сначала некоторые простые понятия обычной евклидовой геометрии. Рассмотрим круговой или сферический объект массой M и радиусом R . Объект может либо быть сплошным (однородная плотность), либо состоять из полостей, но в любом случае мы предположим, что его плотность не зависит от размера (рис. 13.2). Следовательно, при увеличении радиуса объекта от R до $2R$, его масса увеличивается в R^2 раз, если объект круговой, или в R^3 раз, если он сферический. Эту связь массы и длины можно записать в виде

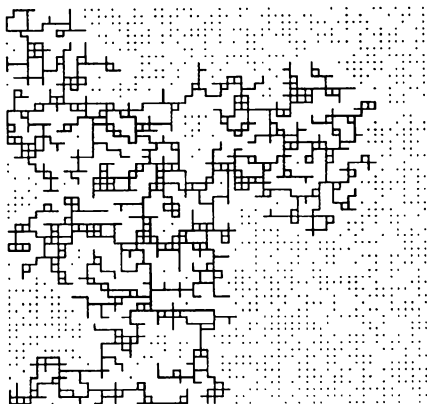


Рис. 13.1. Пример перколяционного кластера, полученного для значения $p = 0.5927$ на квадратной решетке размером $L = 60$. Занятые узлы, не входящие в соединяющий кластер, показаны точками; незанятые узлы не изображаются.

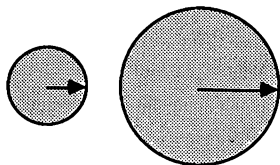


Рис. 13.2. Число «точек» на единицу площади в каждом круге одинаково. Как зависит полное число точек (масса) от радиуса круга?

$$M(R) \sim R^d, \quad (13.1)$$

где d — размерность пространства. Объект, у которого масса и размер связаны соотношением (13.1), называется «компактным». Данное соотношение означает, что если линейный размер компактного объекта увеличивается в R раз при неизменной форме, то его масса увеличивается в R^d раз. Это масштабное соотношение масса — размер тесно связано с интуитивным представлением о размерности, а также является полезным обобщением на размерности больше трех. Заметим, что если масса M и размер R удовлетворяют соотношению (13.1), то масштабное соотношение для плотности $\rho = M/R^d$ имеет вид

$$\rho = R^0. \quad (13.2)$$

Связь между массой объекта и его характерным размером R можно определить в более общем виде, чем в формуле (13.1). Одна из формулировок определения фрактальной размерности основана на соотношении

$$M(R) \sim R^{d_f}. \quad (13.3)$$

Мы называем объект «фрактальным», если он удовлетворяет соотношению (13.3) со значением d_f меньшим, чем пространственная размерность d . Заметим, что если для объекта выполняется соотношение (13.3), то его плотность не будет одинаковой для всех значений R , а масштабируется следующим образом:

$$\rho(R) \sim M/R^d \sim R^{d_f-d}. \quad (13.4)$$

Поскольку $d_f < d$, то плотность фрактального объекта уменьшается с увеличением размера. Данная масштабная зависимость плотности служит

количественной мерой представления о фракталах, как о разреженных или ветвистых объектах. Другой метод описания фрактального объекта основан на предположении, что он содержит пустоты всех размеров.

Перколяционный кластер, показанный на рис. 13.1, является примером *случайного*, или статистического фрактала, поскольку для него соотношение масса—размер (13.1) выполняется только «в среднем», например при усреднении соотношения $M(R)$ по многим кластерам и различным начальным точкам в кластере. Во всех реальных физических системах соотношение (13.3) выполняется не для любых масштабов длины, а ограничивается верхним и нижним пределами. Например, нижний предел длины обуславливается тем или иным микроскопическим расстоянием, таким как период решетки или среднее расстояние между составными частями объекта. В численном моделировании верхний предел длины обычно определяется конечным размером системы. Наличие указанных пределов усложняет вычисление фрактальной размерности.

В задаче 13.1 вычисляется фрактальная размерность перколяционных кластеров с помощью простых методов Монте-Карло. Вычисление d_f методом ренорм-группы рассматривается в задаче 13.2. Учтите, что для получения убедительного доказательства существования степенной зависимости между M и R и для вычисления с приемлемой точностью значения показателя степени требуется повысить точность данных на несколько десятичных знаков. Таким образом, выводы, основанные на таком конечном моделировании, излагаемые в этих задачах, должны интерпретироваться очень осторожно.

ЗАДАЧА 13.1. Фрактальная размерность перколяционных кластеров

а. Сгенерируйте перколяционную конфигурацию для $p = 0.5927$ —наилучшей оценки порога перколяции на квадратной решетке. Рассмотрите свойства соединяющего кластера на решетке размером 61×61 . Почему может потребоваться генерирование нескольких конфигураций, прежде чем получится соединяющий кластер? Сначала прочувствуйте разреженную структуру соединяющего кластера, изображая координаты занятых узлов в соединяющем кластере, как показано на рис. 13.1. Много ли у перколяционного кластера висящих обрывков?

б. Выберите точку в этом кластере и подсчитайте число точек в соединяющем кластере $M(b)$ внутри квадрата площадью b^2 с центром в этой точке. Затем удвойте b и подсчитайте число точек, попавших внутрь такого квадрата. Повторяйте эту процедуру до тех пор,

пока не сумеете оценить зависимость числа точек от b . Можно ли продолжать эту процедуру бесконечно? Используя зависимость $M(b)$, оцените величину d_f в соответствии с определением (13.3). Выберите другую точку в соединяющем кластере и повторите эту процедуру. Получите ли вы аналогичные результаты? Лучшую оценку d_f можно получить, проведя усреднение по нескольким начальным точкам в каждом кластере и по многим соединяющим кластерам.

в. Если вы еще не решили задачу 12.7в, то вычислите d_f , определяя средний размер (массу) M соединяющего кластера при $p = p_c$, как функцию размера решетки L . Рассмотрите случаи $L = 10, 20, 40$ и 60 и оцените d_f , используя график зависимости M от L , построенный в двойном логарифмическом масштабе.

*г. Сгенерируйте перколяционную конфигурацию для $p = 0.8$ на решетке размером 61×61 . Будет ли соединяющий кластер фракталом?

ЗАДАЧА 13.2. Ренорм-групповое вычисление фрактальной размерности

Вычислите $\langle s^2 \rangle$, среднеквадратичное число занятых узлов в соединяющем кластере при $p = p_c$, и величину $\langle s'^2 \rangle$, среднеквадратичное число занятых узлов в соединяющем кластере на ренормированной решетке с $L' = L/b$. Поскольку $\langle s^2 \rangle \sim R^{2d_f}$ и $\langle s'^2 \rangle \sim (R/b)^{2d_f}$, можно получить d_f из соотношения $b^{d_f} = \langle s^2 \rangle / \langle s'^2 \rangle$. Положите масштабный множитель равным $b = 2$ и позаимствуйте тот же метод блокировки, который применялся в разд. 12.5. Для получения качественных результатов для $L = 16$ и $p = 0.5927$ достаточно проводить усреднение по десяти соединяющим кластерам.

В задачах 13.1 и 13.2 мы рассмотрели свойства только соединяющих кластеров, хотя алгоритм предназначен для генерирования перколяционных конфигураций, содержащих кластеры всех размеров. Существует более эффективный способ генерирования *обособленных* перколяционных кластеров, предложенный независимо Хаммерсли, Лисом и Александровичем. Их алгоритм «роста» состоит из следующих шагов (рис. 13.3).

1. Занимается одна затравочная ячейка решетки. Четыре соседних (на квадратной решетке) ячейки образуют *периметр* затравочной ячейки.
2. Выбирается случайным образом ячейка периметра и генерируется случайное число r . Если $r \leq p$, то ячейка занимается, в против-

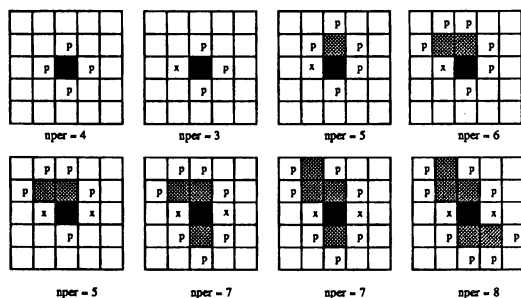


Рис. 13.3. Пример «роста» перколяционного кластера. Ячейки занимаются с вероятностью $p = 0.6$ и остаются пустыми с вероятностью $1 - p$. Затравочная ячейка закрашена черным цветом, периметр обозначен буквой p , а проверенные пустые ячейки — буквой x . Для каждого шага роста приведено число ячеек $nstep$, расположенных по периметру.

ном случае она остается свободной. Чтобы ячейки были свободными с вероятностью $1 - p$, данная ячейка больше не проверяется.

3. Если ячейка занята, то определяется, нет ли новых ячеек периметра, т.е. непроверенных соседних ячеек. Эта процедура продолжается до тех пор, пока не получится («не вырастет») кластер требуемого размера, или пока не останется непроверенных ячеек.

В программе `single_cluster` реализован алгоритм роста и вычисляется число занятых ячеек в радиусе r от затравочной частицы. Затравочная ячейка находится в начале координат, при этом решетка занимает положение от $-L$ до L . В двух одномерных массивах хранятся координаты x и y ячеек периметра.

PROGRAM single_cluster

```
! кластер генерируется по алгоритму Хаммерсли-Лиса-Александровица
DIM num(1000)
CALL parameter(L,p)
CALL grow(L,num,p)
CALL mass_plot(num,L)
END
```

SUB parameter(L,p)

```
INPUT prompt "значение L = ": L      ! решетка (2L+1) x (2L+1)
INPUT prompt "вероятность занятия ячейки = ": p
END SUB
```

```

SUB grow (L,num(),p)  ! генерирование одного перколяционного кластера
  DIM perx(5000),pery(5000),site(-50 to 50,-50 to 50)
  DIM nx(4),ny(4)  ! координаты ближайших соседей вновь занятой ячейки
  DATA 1,0,-1,0,0,1,0,-1
  LET site(0,0) = 1      ! затравочная ячейка в начале координат
  FOR i = 1 to 4
    READ perx(i),pery(i)  ! координаты периметра затравочной ячейки
    LET nx(i) = perx(i)    ! направление роста для ячеек периметра
    LET ny(i) = pery(i)
  NEXT i
  LET nper = 4            ! начальное число ячеек периметра
  DO
    LET iper = int(rnd*nper + 1)  ! случайный выбор ячейки периметра
    LET x = perx(iper)            ! координаты ячейки периметра
    LET y = pery(iper)
    IF rnd < p then                ! ячейка занимается
      LET site(x,y) = 1
      LET r = sqr(x*x + y*y)      ! расстояние от начала координат
      LET num(r) = num(r) + 1    ! число ячеек на расстоянии r
      LET n = n + 1              ! число занятых ячеек
      LET perx(iper) = perx(nper) ! вновь занятая ячейка периметра
      LET pery(iper) = pery(nper) !  заменяется последней из списка
      LET nper = nper - 1
      FOR iz = 1 to 4              ! нахождение новых ячеек периметра
        LET xnew = x + nx(iz)
        LET ynew = y + ny(iz)
        IF site(xnew,ynew) = 0 and abs(xnew) <= L and abs(ynew) <= L then
          LET nper = nper + 1
          LET perx(nper) = xnew
          LET pery(nper) = ynew
        END IF
      NEXT iz
    ELSE                            ! ячейка проверяется, но не занимается
      LET site(x,y) = -1
      LET perx(iper) = perx(nper)
      LET pery(iper) = pery(nper)
      LET nper = nper - 1
    END IF
  LOOP until nper < 1  ! до проверки всех ячеек периметра на решетке
END SUB

```

```

SUB mass_plot(num(),L)                                ! график функции  $\ln M$  от  $\ln R$ 
  LET ymax = 1 + int(log(4*L*L))                      ! задание логарифмических осей
  LET xmax = 1 + int(log(L))
  SET WINDOW -1,xmax + 1,-1,ymax + 1
  PLOT LINES: 0,ymax; 0,0;xmax,0
  PLOT TEXT, at 1,ymax: "График функции  $\log M$  от  $\log R$ "
  FOR x = 1 to xmax                                  ! нанесение делений
    PLOT LINES: x,0; x,0.02*log(L)
  NEXT x
  FOR y = 1 to ymax
    PLOT LINES: 0,y; 0.01*log(L),y
  NEXT y
  LET numtot = 0
  FOR r = 1 to L
    IF num(r) = 0 then EXIT SUB
    LET numtot = numtot + num(r)
    LET lnR = log(r)
    LET lnM = log(numtot)
    BOX AREA lnR - 0.02,lnR + 0.02,lnM - 0.04,lnM + 0.04
  NEXT r
END SUB

```

В задаче 13.3 мы пользуемся алгоритмом роста для генерирования обособленных перколяционных кластеров. Фрактальная размерность определяется путем подсчета в кластере числа ячеек M , находящихся в радиусе r от затравочной частицы. Типичный график зависимости $\ln M$

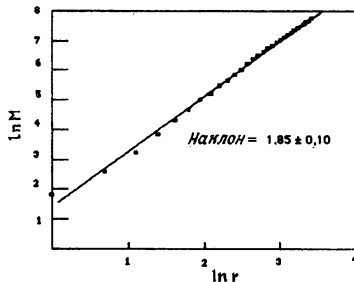


Рис. 13.4. График зависимости $\ln M$ от $\ln r$ для перколяционного кластера, сгенерированного при $p = 0.5927$ на решетке размером 61×61 . Прямая проведена по данным вручную. Наклон этой прямой является оценкой фрактальной размерности. График получен с помощью программы `single cluster`. Точное значение фрактальной размерности для перколяционного кластера составляет $d_f = 91/48 \approx 1.896$.

от $\ln r$ показан на рис. 13.4. Обратите внимание на кривизну в области больших и малых r . Поскольку правильность соотношения (13.3) при малых r и $r \sim L$ не предполагается, то следует придать промежуточным значениям r больший вес. Для получения грубого значения d_f достаточно оценки на глаз.

ЗАДАЧА 13.3. Рост обособленного кластера и фрактальная размерность

а. Воспользуйтесь программой `single_cluster`, в которой реализован алгоритм роста обособленных кластеров на решетке размером $(2L + 1) \times (2L + 1)$. Рассмотрите «соединяющий» кластер, который является одним из тех, масса которых больше, чем заданная минимальная, или принадлежит множеству кластеров, соединяющих между собой верхний и нижний ряды решетки. Можно ли «вырастить» соединяющий кластер для значения $p = 0.4$ или процесс роста обычно заканчивается после занятия нескольких ячеек?

б. Положите $p = 0.5927$ и $L = 30$, растите кластер до тех пор, пока не останется непроверенных ячеек, и постройте несколько картинок соединяющих кластеров. Заметим, что некоторые ваши попытки закончатся неудачей и вы не получите соединяющего кластера. Определите число занятых ячеек $M(r)$ в радиусе r от затравочной ячейки. (Точнее говоря, расстояние r должно измеряться от центра масс кластера.) Получите M для нескольких значений r и усредните $M(r)$ по крайней мере по десяти соединяющим кластерам. Оцените d_f из графика зависимости M от r , построенного в двойном логарифмическом масштабе (рис. 13.4). Если позволит время, то сгенерируйте перколяционные кластеры на решетках большего размера.

в. Сгенерируйте кластеры для значения $p = 0.65$, которое несколько превышает p_c . Постройте график функции $M(r)$ в двойном логарифмическом масштабе. Равен ли приблизительно наклон этого графика значению d_f , полученному в п. «б»? Увеличивается или уменьшается наклон графика с ростом r ? Повторите расчет для значения $p = 0.80$. Является ли соединяющий кластер фракталом при $p > p_c$?

г. Фрактальная размерность перколяционных кластеров не является независимым показателем, а удовлетворяет масштабному закону

$$d_f = d - \beta/\nu \quad (13.5)$$

(значения β и ν были приведены в табл. 12.1). Соотношение (13.5) можно вывести, привлекая аппарат конечномерного масштабирования. Изложим кратко доказательство. Число ячеек в соединяющем кластере на решетке размером L определяется выражением

$$M(L) \sim P_{\omega}(L)L^d, \quad (13.6)$$

где P_{ω} — вероятность того, что занятая ячейка принадлежит соединяющему кластеру, и величина L^d пропорциональна полному числу ячеек решетки. Нам известно, что в предельном случае бесконечной решетки и значений p в окрестности p_c величины $P_{\omega}(p) \sim (p - p_c)^{\beta}$ и $\xi(p) \sim (p - p_c)^{-\nu}$ независимо от значения L . Для $L \sim \xi$ можно воспользоваться конечномерным масштабированием и найти, что $P_{\omega}(L) \sim L^{-\beta\nu}$ [см. (12.12)], так что в результате имеем

$$M(L) \sim L^{-\beta\nu}L^d \sim L^{d_f}. \quad (13.7)$$

Отсюда следует соотношение (13.5). Используя точные значения β и ν из табл. 12.1 и найдите точное значение d_f для $d = 2$. Согласуется ли ваша оценка d с этим значением?

*д. Оцените фрактальную размерность перколяционных кластеров на простой кубической решетке. (Положите $p_c = 0.3117$.)

Фракталы применяются для описания нерегулярных форм в таких различных системах, как турбулентность, береговые линии, горные цепи и облака. Обсуждение фрактальной геометрии, содержащее множество прекрасных иллюстраций фракталов, полученных на компьютере, можно найти в книге Мандельброта (см. список литературы).

13.2. РЕГУЛЯРНЫЕ ФРАКТАЛЫ И САМОПОДОБИЕ

Смысл выражения (13.3) состоит в том, что фрактальные объекты *самоподобны*, т.е. они выглядят одинаково в любом пространственном масштабе. Как будет выглядеть часть перколяционного кластера, изображенного на рис. 13.1, если смотреть на него через лупу? Для уяснения сущности самоподобия рассмотрим пример *регулярного* фрактала, или объекта, который самоподобен при всех масштабах длины. Начнем с

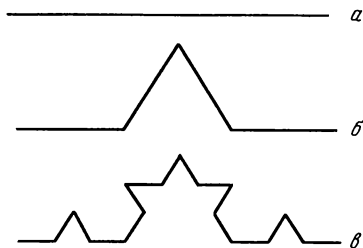


Рис. 13.5. Три стадии (а)–(в) формирования самоподобной кривой Коха. На каждой стадии замена средней трети каждого сегмента производится в направлении, которое увеличивает площадь под кривой. Эти кривые сформированы с помощью программы Koch.

отрезка единичной длины (рис. 13.5,а). Предположим, что удаляется средний кусок этого отрезка длиной $\frac{1}{3}$ и заменяется на два отрезка длиной $\frac{1}{3}$ каждый так, что на кривой появляется треугольный горб и общая длина кривой становится равной $\frac{4}{3}$ (рис. 13.5,б). На следующем шаге каждый сегмент длиной $\frac{1}{3}$ делится на три части длиной $\frac{1}{9}$ и процедура повторяется (рис. 13.5,в). Какова длина кривой, изображенной на рис. 13.5,в?

Представим себе, что показанные на рис. 13.5 три стадии построения могут быть повторены бесконечное число раз, и получается кривая бесконечной длины, состоящей из бесконечного числа бесконечно малых сегментов. Такая кривая называется треугольной кривой Коха. Ниже приводится программа на языке True Basic, в которой используется рекурсивная процедура рисования такой кривой.

PROGRAM Koch

! генерирование треугольной кривой Коха с помощью рекурсии

CALL initial (x1,y1,x2,y2,n)

DO ! рисование кривой Коха для разного числа итераций

LET k = 0

CALL draw (x1,y1,x2,y2,n)

DO

! пауза до нажатия любой клавиши

GET KEY k

LOOP UNTIL k <> 0

LET n = n + 1

! номер шага построения

CLEAR

LOOP

END

```

SUB initial (x1,y1,x2,y2,n)
  LET n = 0
  LET x1 = 0                      ! координаты левого конца отрезка
  LET y1 = 0
  LET x2 = 10                     ! координаты правого конца отрезка
  LET y2 = 0
  SET window x1-1,x2 + 1,-1,7    ! произвольные единицы
END SUB

SUB draw (x1,y1,x2,y2,n)
  IF n > 0 then
    LET dx = (x2 - x1)/3
    LET dy = (y2 - y1)/3
    LET x1n = x1 + dx
    LET y1n = y1 + dy
    LET x2n = x1 + 2*dx
    LET y2n = y1 + 2*dy
    ! поворот сегмента (dx,dy) на 60 градусов и присоединение к (x1n,y1n)
    LET xmid = dx*.5 - dy*.866 + x1n
    LET ymid = dy*.5 + dx*.866 + y1n
    CALL draw(x1,y1,x1n,y1n,n-1)
    CALL draw(x1n,y1n,xmid,ymid,n-1)
    CALL draw(xmid,ymid,x2n,y2n,n-1)
    CALL draw(x2n,y2n,x2,y2,n-1)
  ELSE
    PLOT LINES: x1,y1;x2,y2
  END IF
END SUB

```

Обратите внимание на то, что подпрограмма **draw** вызывает сама себя. С помощью программы **Koch** получите кривые, показанные на рис. 13.5.

Полезно обсудить фрактальную размерность на примере регулярных фракталов. Рассмотрим сначала отрезок единичной длины, который разбит на N равных кусков длиной ℓ , так что $N = 1/\ell$ (рис. 13.6). По мере уменьшения ℓ значение N растет линейно, что и следовало ожидать для одномерной кривой. Аналогично, если мы разделим квадрат единичной площади на N равных квадратиков со стороной ℓ , то получим $N = 1/\ell^2$ —ожидаемый для двумерного объекта результат (рис. 13.6). Можно утверждать, что в общем случае $N = 1/\ell^d$, где d —размерность объекта. Следовательно, логарифмируя обе части этого равенства,

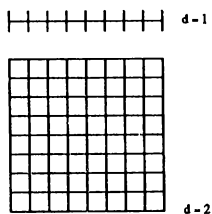


Рис. 13.6. Примеры одномерного и двумерного объектов.

можно выразить размерность в виде

$$d = \log N / \log (1/\ell). \quad (13.8)$$

Теперь применим эти соображения к кривой Коха. Мы нашли, что при каждом уменьшении длины ℓ нашей единицы измерения в три раза число сегментов увеличивается в четыре раза. Таким образом, имеем $N = 4$ и $\ell = \frac{1}{3}$ и фрактальная размерность треугольной кривой Коха равна

$$d_f = \log 4 / \log 3 \approx 1.2619. \quad (13.9)$$

Следовательно, о кривой Коха можно сказать, что она уже не одномерна, но еще и не двумерна. Соответствует ли это утверждение зрительному впечатлению о характере заполнения пространства треугольной кривой Коха?

ЗАДАЧА 13.4. Генерирование регулярных фракталов и их фрактальная размерность.

а. Концепция рекурсивного программирования, проиллюстрированная в программе **Koch**, является, вероятно, одной из самых трудных, с которыми вам доведется столкнуться. Опишите структуру программы **Koch** и объясните характер рекурсии.

б. Регулярные фракталы можно получать из шаблона, который удается использовать самоповторяющимся образом. Напишите программу построения квадратной кривой Коха, изображенной на рис. 13.7,а. Чему равна фрактальная размерность этого объекта?

в. Чему равна фрактальная размерность прокладки Серпинского, показанной на рис. 13.7,б? Напишите программу, которая сформирует

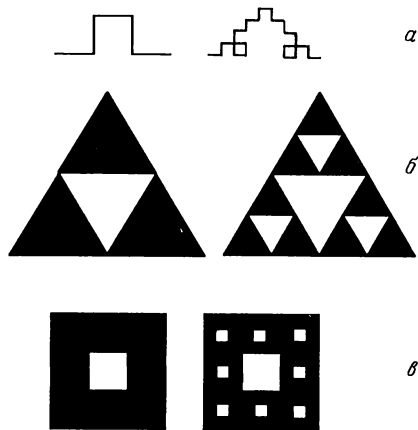


Рис. 13.7. *а*—Несколько первых итераций квадратной кривой Коха; *б*—несколько первых итераций прокладки Серпинского; *в*—несколько первых итераций ковра Серпинского.

несколько следующих итераций для такого объекта.

г. Чему равна фрактальная размерность ковра Серпинского, показанного на рис. 13.7, *в*? Как соотносятся фрактальные размерности ковра Серпинского и перколяционного кластера? Похожи ли внешне эти два фрактальных объекта?

13.3. ПРОЦЕССЫ РОСТА ФРАКТАЛОВ

Мы уже упомянули, что многие системы, встречающиеся в природе, имеют фрактальную структуру. Почему фрактальные структуры встречаются так часто? Как образуются фрактальные структуры? Ниже мы рассмотрим некоторые простые модели, обнаруживающие структуры, которые можно описать на языке фрактальной геометрии и которые проявляют удивительное сходство с объектами, встречающимися в природе.

Модель Эдена. Простую модель роста предложил Эден в 1961 г. для моделирования развития клеточных колоний. Хотя мы установим, что результирующий кластер является компактным, описание алгоритма роста Эдена иллюстрирует суть моделей роста фракталов, которые мы будем обсуждать в дальнейшем.

		Р					
	Р	16	Р				
	Р	14	11	Р			
Р	17	10	2	Р	18	Р	
	Р	6	1	3	13	Р	
	Р	9	5	4	7	12	Р
	Р	15	Р	8	Р	Р	
		Р					

Рис. 13.8. Пример роста кластера на квадратной решетке в соответствии с моделью Эдена. Числа в ячейках обозначают порядок, в котором эти ячейки занимались, а ячейки периметра обозначены буквой p .

Поместим начальную («затравочную») ячейку в центр решетки. Ближайшие к занятым ячейкам свободные соседи образуют их *периметр*. В простейшей версии модели случайным образом выбирается ячейка периметра и занимает. Вновь занятая ячейка удаляется из списка ячеек периметра и в список добавляются новые ячейки периметра. Данный процесс роста повторяется много раз до тех пор, пока не образуется большой кластер, состоящий из занятых ячеек (рис. 13.8). Кластеры Эдена исследуются в задаче 13.5.

ЗАДАЧА 13.5. Моделирование методом Монте-Карло кластеров Эдена

Модифицируйте программу `single_cluster` так, чтобы кластеры генерировались на квадратной решетке в соответствии с моделью Эдена. Простейшая модификация заключается в занятии ячеек периметра с вероятностью $p = 1.0$ до тех пор, пока кластер не достигнет края решетки. Что произойдет, если мы продолжим занятие ячеек периметра до бесконечности? Воспользуйтесь процедурой из задачи 13.3 и вычислите число занятых ячеек $M(r)$, находящихся в радиусе r от начальной ячейки. Положите $M(r) \sim r^{d_f}$ и оцените значение d_f из наклона графика зависимости M от r , построенного в двойном лог-

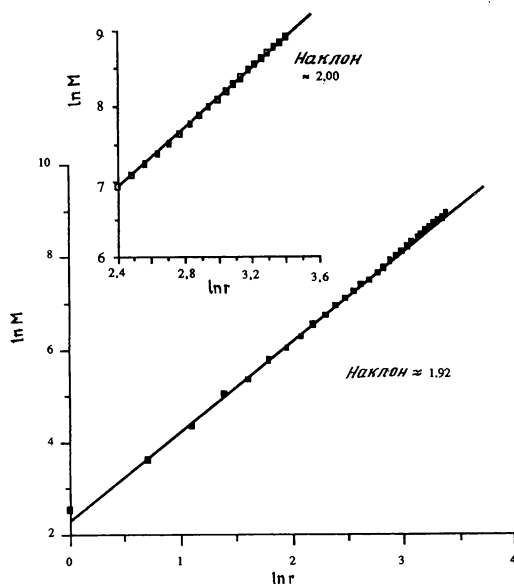


Рис. 13.9. Зависимость $\ln M$ от $\ln r$ для кластера Эдена, полученного на решетке размером 61×61 . Величина наклона прямой, построенной методом наименьших квадратов по данным для всех r , составляет приблизительно 1.92. Удаление данных для малых r уменьшает кривизну и приводит в данном случае к наклону, близкому к 2.0.

рифмическом масштабе. Типичный график такой зависимости в двойном логарифмическом масштабе показан на рис. 13.9. Соответствующий наклон равен 1.92. Для исключения возможной кривизны при малых значениях r мы приводим результаты, полученные методом наименьших квадратов по тем же данным в области $12 \leq r \leq 30$. Найдено, что наклон увеличивается примерно до значения 2.0. Чтобы прийти к выводу о том, что размерность кластеров Эдена равна размерности решетки, необходимо провести тщательный анализ данных, усредненных по многим испытаниям и по крайней мере по двум десяткам значений r . Можете ли вы утверждать, исходя из своих данных, что кластеры Эдена являются компактными?

Оккупирующая перколяция. Динамический процесс, называемый *оккупирующей перколяцией*, применяется для моделирования структуры поверхности раздела масло—вода при нагнетании воды в пористые среды. В данном процессе кластер растет по пути наименьшего сопротивления. Рассмотрим решетку размером $L \times 2L$, в которой вода («оккупант») в начальный момент времени занимает левый край (рис. 13.10). Спро-

тивление оккупации задается случайными числами от 0 до 1, которые приписываются каждой ячейке решетки и не меняются в течение всего процесса. Ячейки, граничащие с ячейками-оккупантами, составляют периметр. На каждом временном шаге оккупант занимает ячейку периметра с наименьшим случайным числом и вытесняет масло («защитник»).

Процесс оккупирующего перколяционного роста реализован в программе *invasion*. Изначально в массиве $r(i,j)$ записаны приписанные ячейкам случайные числа. Если ячейка с координатами (i,j) занята, то элемент $r(i,j)$ увеличивается на 1. Если ячейка (i,j) принадлежит периметру, то $r(i,j)$ увеличивается на 2. Для упорядочения ячеек периметра по величине приписанных им случайных чисел применяются два метода сортировки. Рост оккупирующего кластера продолжается до тех пор, пока не образуется путь, связывающий левый и правый края ре-

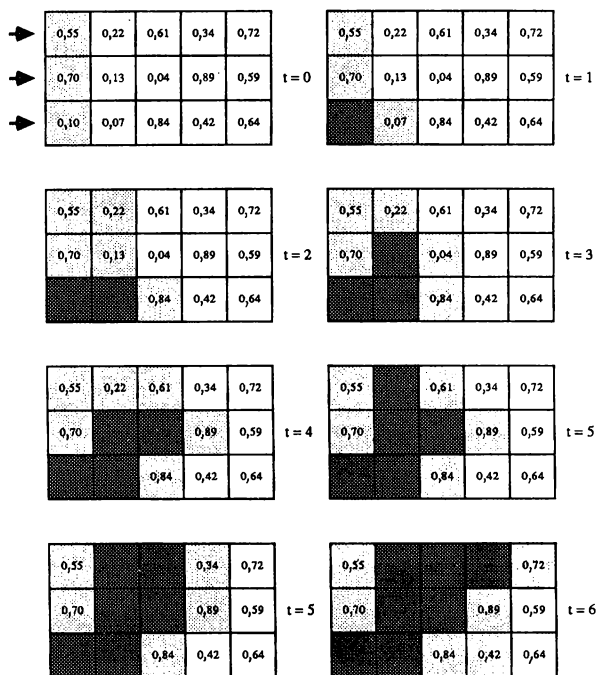


Рис. 13.10. Пример образования кластера при оккупирующей перколяции. Состояние решетки при $t = 0$ показано случайными числами, приписанными ячейкам. Ячейки с темной штриховкой заняты оккупантом, который на следующем шаге занимает ячейку периметра (светлая штриховка) с наименьшим случайным числом.

шетки. Для уменьшения граничных эффектов на верхней и нижней границах берутся периодические краевые условия, а все физические величины измеряются в центральной области размером $L \times L$. Программа рисует занятые («мокрые») ячейки и вычисляет их относительное количество. Основной рассматриваемой величиной является $P(r)dr$ — вероятность занятия ячейки, содержащей случайное число в интервале $[r, r + dr]$. Свойства модели оккупирующей перколяции изучаются в задаче 13.6.

```
PROGRAM invasion  ! расчет оккупирующего перколяционного кластера
DIM r(100,50),perx(1000),pery(1000)
RANDOMIZE
CALL initial(Lx,Ly)
CALL assign(r,perx,pery,Lx,Ly,s)
CALL invade(r,perx,pery,Lx,Ly,s)
CALL average(r,Lx,Ly)
END
```

```
SUB initial(Lx,Ly)
  INPUT prompt "размер решетки в направлении y = ": Ly
  LET Lx = 2*Ly
  SET window 0,Lx+1,0,Ly+2
END SUB
```

```
SUB assign(r(),perx(),pery(),Lx,Ly,s)
  ! присваивание случайных чисел ячейкам и занятие первого столбца
  ! ячейки во втором столбце образуют начальный периметр
  LET s = 0.4      ! полуширина квадрата, изображающего занятый узел
  FOR row = 1 to Ly
    LET r(1,row) = 1      ! занимает первый столбец
    BOX AREA 1 - s,1 + s,row - s,row + s
    FOR col = 2 to Lx      ! присваивание случайных чисел строке
      LET r(col,row) = rnd
    NEXT col
    ! если ячейка периметра, то r(col,row) больше 2
    LET r(2,row) = 2 + r(2,row)
    LET nper = row      ! число ячеек периметра равно номеру строки
    ! сортировка ячеек периметра во втором столбце
    CALL sort_shell(r,perx,pery,nper,2,row) ! или вызов sort_insert
  NEXT row
END SUB
```

```
SUB sort_shell (r(.),perx(.),pery(.),nper,x0,y0)
! список делится пополам и определяется, куда попало случайное число
! половина списка опять делится пополам и определяется, где число
! этот процесс продолжается до определения точного положения числа
IF nper = 1 then                                ! только одна ячейка периметра
    LET perx(nper) = x0
    LET pery(nper) = y0
    EXIT SUB
END IF
! не меньше ли случайное число в новой ячейке всех предыдущих чисел
LET ix = perx(nper-1)
LET iy = pery(nper-1)
IF r(x0,y0) < r(ix,iy) then
    LET perx(nper) = x0
    LET pery(nper) = y0
    EXIT SUB
END IF
LET k1 = 1                                     ! начало списка
LET k2 = nper-1                               ! конец списка
LET kmid = int((k1 + k2)/2)                   ! середина списка
DO                                             ! начало сортировки Шелла
    LET ix = perx(kmid)
    LET iy = pery(kmid)
    ! определение в какой половине списка находится новое число
    IF r(x0,y0) > r(ix,iy) then
        LET k2 = kmid                        ! замена верхней границы
    ELSE
        LET k1 = kmid                        ! замена нижней границы
    END IF
    LET kmid = int((k1 + k2)/2)
    IF (k1 = kmid) or (kmid = k2) then        ! точное положение равно k2
        FOR ilist = nper to k2 + 1 step -1  ! ячейки выше k2 сдвигаются
            LET perx(ilist) = perx(ilist-1)
            LET pery(ilist) = pery(ilist-1)
        NEXT ilist
        LET perx(k2) = x0                    ! внесение в список новой ячейки
        LET pery(k2) = y0
        EXIT SUB
    END IF
LOOP
```

END SUB

```
SUB sort_insert (r(,),perx(),pery(),nper,x0,y0)  ! метод сортировки
  FOR iper = 1 to nper - 1
    LET ix = perx(iper)
    LET iy = pery(iper)
    IF r(x0,y0) > r(ix,iy) then                ! вносим новую ячейку
      FOR ilist = nper to iper + 1 step -1
        LET perx(ilist) = perx(ilist-1)
        LET pery(ilist) = pery(ilist-1)
      NEXT ilist
      LET perx(iper) = x0
      LET pery(iper) = y0
    EXIT SUB
  END IF
NEXT iper
LET perx(iper) = x0 ! новая ячейка периметра меньше всех предыдущих
LET pery(iper) = y0
END SUB
```

```
SUB invade (r(,),perx(),pery(),Lx,Ly,s)
  DIM nx(4),ny(4) ! nx и ny указывают положение ближайших соседей
  DATA 1,0,-1,0,0,1,0,-1
  FOR iz = 1 to 4
    READ nx(iz),ny(iz)
  NEXT iz
  LET nper = Ly
  DO
    LET col = perx(nper)
    LET row = pery(nper)
    LET nper = nper - 1
    LET r(col,row) = r(col,row) - 1 ! занятая ячейка между 1 и 2
    BOX AREA col - s col + s,row - s,row + s
    FOR nn = 1 to 4 ! получение нового периметра
      LET xnew = col + nx(nn)
      LET ynew = row + ny(nn)
      IF ynew > Ly then ! периодические краевые условия по y
        LET ynew = 1
      ELSE IF ynew < 1 then
        LET ynew = Ly
      
```

```

END IF
IF r(xnew,ynew) < 1 then          ! новая ячейка периметра
  LET r(xnew,ynew) = r(xnew,ynew) + 2
  LET nper = nper + 1
  CALL sort_shell(r,perx,pery,nper,xnew,ynew) ! или вызов sort_insert
END IF
NEXT nn
LOOP until col >= Lx ! выход по достижении кластером правой границы
END SUB

SUB average (r(,),Lx,Ly)
! вычисление плотности вероятности P(r)
DIM P(0 to 20),nr(0 to 20)
LET Lmin = Lx/3
LET Lmax = 2*Lmin
LET n = (Lmax - Lmin + 1)*Ly ! число ячеек в средней половине решетки
LET dr = 0.05
LET nbin = 1/dr
FOR col = Lmin to Lmax
  FOR row = 1 to Ly
    LET ibin = nbin*(mod(r(col,row),1))
    LET nr(ibin) = nr(ibin) + 1
    IF (r(col,row) >= 1) and (r(col,row) < 2) then
      LET occupied = occupied + 1      ! полное число занятых ячеек
      LET P(ibin) = P(ibin) + 1
    END IF
  NEXT row
NEXT col
DO          ! для печати результатов ждем нажатия клавиши
LOOP until key input
CLEAR
LET fraction = occupied/n
PRINT "доля оккупированных ячеек = ", fraction
PRINT " r", "P(r)"
PRINT
FOR ibin = 0 to nbin
  LET rnum = dr*ibin
  IF nr(ibin) > 0 then PRINT rnum, p(ibin)/nr(ibin)
NEXT ibin
END SUB

```


ЗАДАЧА 13.6. Оккупирующая перколяция

а. Воспользуйтесь программой `invasion` для генерирования оккупирующего кластера на квадратной решетке. Какие можно сделать качественные утверждения о свойствах такого кластера? Разберите обе подпрограммы сортировки, реализованные в программе. Какой из методов оказывается более быстрым на решетке размером 20×40 ?

б. Модифицируйте программу `invasion` так, чтобы доля ячеек S_r занятых оккупантом в момент первого достижения правой границы, усреднялась по крайней мере по 20 испытаниям. Предполагая, что $S_i \sim L^{-\alpha}$, оцените значение α из графика зависимости $\ln S_i$ от $\ln L$. Как связана величина α с фрактальной размерностью соединяющего кластера? Сравните полученную оценку фрактальной размерности со случаем обычной перколяции. (Опубликованные для S_i результаты Уилкинсона и Уилемсена получены для 2000 реализаций для каждого значения L в интервале от 20 до 100.)

в. Вычислите $P(r)dr$ — вероятность того, что ячейка, содержащая случайное число в интервале $[r, r + dr]$, занята. Достаточно рассмотреть 20 интервалов случайных чисел, т.е. выбрать $dr = 0.05$. Постройте график функции $P(r)$ для $L = 10$, а также для больших значений L вплоть до $L = 50$. В состоянии ли вы определить «критическое значение» r , в окрестности которого $P(r)$ изменяется очень сильно? Как соотносится это критическое значение r со значением p_c для обычной перколяции на квадратной решетке?

*г. Модель оккупирующей перколяции, которую мы обсудили, годится для описания вытеснения бесконечно сжимаемой жидкости несжимаемой жидкостью. Чтобы вытесняемую жидкость (масло) трактовать как несжимаемую, мы вводим механизм поглощения, состоящий в том, что как только масляный кластер становится изолированным, его больше нельзя оккупировать. К сожалению, это правило поглощения сильно замедляет моделирование, поскольку необходимо после каждого шага проверять, не произошло ли поглощение. Учтите механизм поглощения и определите, меняются ли качественно величины α и $P(r)$.

Диффузия в неупорядоченных средах. Предположим, мы хотим изучить процесс диффузии атома в неупорядоченном твердом теле или проводимость случайной резисторной цепи, например проводимость металлической сетки со случайно удаленными узлами. Одна из простейших моделей

таких явлений предложена де Женом и известна как «муравей в лабиринте». Рассмотрим пешехода (муравья), который движется случайным образом только по занятым ячейкам перколяционного кластера. На каждом временном шаге муравей подбрасывает монету с четырьмя возможными исходами (на квадратной решетке). Если результат соответствует переходу в занятую ячейку, то муравей переползает в нее, в противном случае он остается в своей ячейке. В обоих случаях время увеличивается на единицу. Предположим, что занятые ячейки перколяционного кластера встречаются с вероятностью p . В момент времени $t = 0$ мы помещаем муравья случайным образом в кластер и в момент времени t вычисляем квадрат расстояния между его начальным и конечным положениями. Затем повторяем моделирование много раз и получаем среднеквадратичное смещение муравья. Как зависит среднеквадратичное смещение муравья R от p и t ? Как меняются законы диффузии на фрактальной решетке (перколяционный кластер при $p = p_c$)? Эти вопросы рассматриваются в задаче 13.7.

На первый взгляд может показаться удивительным, что случайное блуждание муравья и проводимость случайной резисторной цепи — родственные задачи. Впервые связь между проводимостью и диффузией установил Эйнштейн. Рассмотрим систему частиц, например жидкость. Если следить за движением отдельных частиц в отсутствие внешней силы, то можно определить их среднеквадратичное смещение и, следовательно, коэффициент самодиффузии D . Далее, если приложить «малую» силу, то можно измерить среднюю скорость в направлении действия силы и найти подвижность μ — отношение средней скорости к приложенной силе. Эйнштейну удалось показать, что величины D и μ пропорциональны (Рейф).

Что касается системы заряженных частиц, то можно обобщить приведенные выше рассуждения и показать, что подвижность пропорциональна электрической проводимости. Приложенной силе можно поставить в соответствие напряженность электрического поля и, значит, напряжение, а средней скорости — силу тока. Отсюда, подвижность пропорциональна проводимости (величина, обратная удельному сопротивлению) частиц. Так как подвижность пропорциональна также D , делаем вывод, что D и проводимость пропорциональны. Поэтому, с помощью этой связи можно определить зависимость проводимости перколяционного кластера от p .

*ЗАДАЧА 13.7. Муравей в лабиринте

- а. При $p = 1$ муравей диффундирует на решетке без дефектов и

$R \sim t^{1/2}$. Предположим, что $R \sim D(p)t^{1/2}$ при $p > p_c$. Сгенерируйте перколяционный кластер для значения $p > p_c$, применяя алгоритм роста из задачи 13.3. Положите, что начальное положение муравья совпадает с затравочной ячейкой, и рассмотрите несколько случайных блужданий, вычисляя среднеквадратичное смещение муравья в этом кластере. Времена положите достаточно большими, чтобы R^2 было приблизительно пропорционально t , но достаточно малыми, чтобы выполнялось условие $R < L$. Рассмотрите значения $p = 0.8, 0.7, 0.65$ и 0.62 для $L = 60$ и оцените $D(p)$. Постройте график отношения $D(p)/D(p=1)$ и обсудите его качественное поведение.

б. При значениях $p < p_c$ кластеры конечны, функция $R(t)$ ограничена и поэтому диффузия невозможна. Поскольку $D(p) = 0$ при $p < p_c$ и $D(p) \neq 0$ при $p > p_c$, мы предполагаем, что $D(p) \sim (p - p_c)^z$ для p в окрестности p_c . Повторите вычисления п. «а» для больших значений L и большего интервала значений p и оцените динамический показатель степени z . Так как проводимость пропорциональна D , показатель степени z показывает также, как убывает проводимость вблизи p_c . Выполненные Уотсоном и Лисом измерения проводимости проволоочной сетки дают оценку $z \approx 1.38 \pm 0.12$. Согласуются ли с этим ваши результаты?

в. Можно ожидать, что при $p = p_c$ наблюдается другая зависимость $R(t)$, например $R(t) \sim t^k$ при больших t . Как вы считаете, будет ли значение k больше или меньше $1/2$? Рассчитайте методом Монте-Карло функцию $R(t)$ при $p = p_c$ и оцените показатель степени k .

г. Как говорилось в задаче 11.16, для рассмотрения случайного блуждания на решетке лучше использовать комбинаторный метод, а не обычное моделирование методами Монте-Карло. Суть комбинаторного метода (полного перебора) состоит в том, что $W_{t+1}(i)$ — вероятность того, что муравей находится в i -й ячейке в момент времени $t + 1$, определяется исключительно вероятностями появления муравья в соседних с i -й ячейках в момент времени t . Запомните положение занятых ячеек в каком-нибудь массиве и введите два массива для хранения $W_{t+1}(i)$ и $W_t(i)$ для всех ячеек кластера. Вычисляйте $W_{t+1}(i)$, используя вероятности $W_t(i)$ (рис. 13.11). Зная функцию распределения вероятностей для различных моментов времени, можно рассчитать пространственные средние, такие как среднеквадратичное смещение. Подробности этого метода и полученные результаты обсуждаются в статье Маджиди и др. (1984). Эти исследо-

ватели рассматривали 5000-шаговые блуждания в кластерах из $\sim 10^3$ ячеек, и усредняли результаты по 1000 различных кластеров.

Агрегация с ограничением диффузии. Рост многих объектов, встречающихся в природе, происходит путем случайного присоединения частей. Примеры таких явлений: снежные хлопья, зигзаги молнии и образование трещин вследствие геологического сдвига. Трудно себе представить, что все эти явления объединяет какое-то общее свойство. Однако в последние годы обнаружено множество фактов, указывающих на единство этих процессов. Одна из простейших моделей, помогающих глубже понять такие процессы, называется *агрегацией с ограничением диффузии*, или АОД. На примере этой модели можно увидеть, как случайное движение приводит к образованию красивых самоподобных кластеров. Первый шаг состоит в занятии ячейки с затравочной частицей. Затем из периметра большой окружности с центром в затравочной ячейке выпускается частица. Частица движется случайным образом (диффундирует) до тех пор, пока либо не уйдет за пределы окружности, либо не достигнет периметра затравочной ячейки и не стыкуется с ним. Затем выпускается другая частица и блуждает до тех пор, пока не достигнет периметра одной из частиц и не стыкуется с ним. Эта процедура повторяется много (обычно несколько тысяч) раз до тех пор, пока не образуется

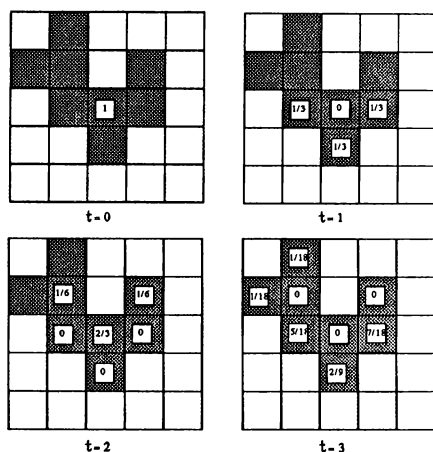


Рис. 13.11. Временная эволюция функции распределения вероятности $W_i(i)$ для трех последовательных моментов времени.

большой кластер. Типичный АОД-кластер показан на рис. 13.12. Напоминает ли вам этот кластер какой-нибудь природный объект? Некоторые свойства АОД-кластеров изучаются в задаче 13.8.

ЗАДАЧА 13.8. Агрегация с ограничением диффузии

а. Напишите программу генерирования кластеров методом Монте-Карло на квадратной решетке в соответствии с алгоритмом диффузии, ограниченной агрегацией. Предположите, что каждый пешеход начинает движение из случайной ячейки на окружности радиусом $2R_{\max}$, где R_{\max} — максимальный радиус кластера, сгенерированного до сих пор. Для уменьшения машинного времени сделайте так, что если пешеход удаляется от начала на расстояние $3R_{\max}$, то он «исчезает», а на начальной окружности случайным образом помещается новый пешеход. Начните с решетки размером $L \sim 40$, а затем рассмотрите решетку максимального размера, определяемого возможностями имеющихся технических средств и вашим терпением. Ограничьте количество пешеходов так, чтобы внешний радиус кластера не рос слишком близко к краям решетки. Как выглядят АОД-кластеры? Если они похожи на фракталы, то оцените на глаз фрактальную размерность. (Специалисты умеют оценивать фрактальную размерность на глаз с точностью до нескольких процентов!)

*б. Возможно, что ваша программа генерирует АОД-кластеры неэффективно, поскольку большую часть времени центральный процессор тратит на моделирование блуждания пешехода вдали от периметра кластера. Существует несколько методов преодоления этой труднос-

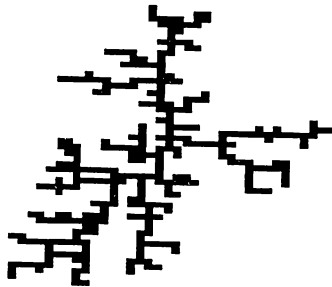


Рис. 13.12. Пример АОД-кластера из примерно 400 частиц на квадратной решетке.

ти. Один из них состоит в том, что чем дальше от кластера находится пешеход, тем большие шаги ему разрешено делать. Например, если пешеход находится на расстоянии $R > R_{\max}$, то разрешается шаг длиной больше или равной $R - R_{\max} - 2$, если это расстояние превышает период решетки. Если пешеход находится очень близко к кластеру, то шаг полагается равным периоду решетки. Другие возможные способы обсуждаются в статье Микина (см. список литературы). Модифицируйте свою программу и оцените фрактальную размерность двумерных кластеров, получаемых при ограничении диффузии.

АОД представляет собой только одну из многих моделей, приводящих к самоподобным кластерам. Мы предлагаем вам привлечь свое воображение и разработать собственную модель процесса роста. Вам поможет также чтение научной литературы по моделям роста, поскольку большая часть ее доступна для вашего понимания.

*ЗАДАЧА 13.9. Образование узоров

а. Вам, вероятно, известен сложный и хаотичный характер картины электрического пробоя, наблюдаемого в атмосферной молнии. Хотя это явление, называемое *пробоем диэлектрика*, сложное, мы увидим, что простая, но нетривиальная модель порождает узоры электрического пробоя, похожие на экспериментально наблюдаемые. Поскольку молния происходит в неоднородной среде из-за различий в плотности, влажности и проводимости воздуха, нам нужно построить модель электрического пробоя в неоднородном диэлектрике. Известно, что когда происходит электрический пробой, потенциал ϕ удовлетворяет уравнению Лапласа $\nabla^2 \phi = 0$. Эту модель (Фемели и др., Нимейер и др.) можно задать следующими правилами (на квадратной решетке):

- i. Рассмотрим большую окружность радиусом R и поместим в ее центр источник заряда. Положим потенциал в центре $\phi = 0$ и на окружности $\phi = 1$ (рис. 13.13). Радиус R необходимо выбрать большим, чем размеры растущего узора.
- ii. Каждому узлу, лежащему внутри окружности, припишем случайное число r ($0 \leq r \leq 1$). Случайное число r_i представляет коэффициент пробоя и случайную структуру диэлектрика.
- iii. Используя метод релаксации (гл. 9), вычисляем значения потенциала ϕ_i в узлах, которые расположены внутри окружности.

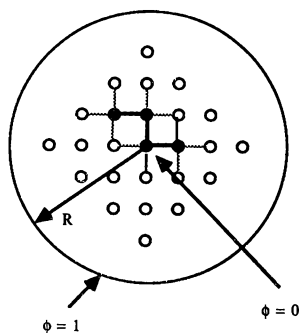


Рис. 13.13. Иллюстрация модели пробоя диэлектрика (Фемили и др.). Центральный узел изображает источник заряда, а «земля» представлена кружком с потенциалом $\phi = 0$. Узор пробоя обозначается темными кружками, потенциал которых везде равен $\phi = 0$. Волнистые линии соединяют занятые узлы с узлами периметра.

iv. Выбираем в качестве периметра ближайшие к узору разряда узлы (черные кружки на рис. 13.13). Обозначим через u_i градиент потенциала в i -м узле и составим произведения ru^a для каждого узла периметра, где a — регулируемый параметр.

v. Узел периметра $imax$, в котором произведение ru^a максимально, пробивается, т.е. полагается $\phi_{imax} = 0$.

vi. Используя метод релаксации, пересчитываем потенциалы в оставшихся «незанятыми» узлах и повторяем шаги (v) и (vi).

Положите $a = \frac{1}{4}$ и проанализируйте структуру узора разряда. Похож ли этот узор на молнию? Имеет ли узор фрактальную структуру? Вычислите фрактальную размерность, подсчитывая $M(b)$ — среднее число узлов в квадрате размером $b \times b$, принадлежащих узору разряда. Рассмотрите другие значения параметра a , например $a = \frac{1}{6}$ и $a = \frac{1}{3}$, и покажите, что узоры обладают фрактальной структурой с подстраивающейся фрактальной размерностью. В опубликованных результатах (Фемили и др.) для формирования узоров делалось 800 шагов роста. Как вы думаете, зависит ли структура узоров от структуры решетки? Какого типа узоры образуются в трехмерном случае?

б. Рассмотрим модель детерминированного роста, в которой не нужно приписывать каждому узлу случайное число. В этой модели все

узлы периметра на каждом шаге роста проверяются на занятость. Рассматриваем те же краевые условия и геометрию, что и в п. «а», и решаем методом релаксации уравнение Лапласа для ϕ . Затем находим узел периметра с наибольшим значением градиента потенциала u и полагаем это значение равным P_{\max} . Далее проверяются все узлы периметра и только те, для которых отношение u/P_{\max} больше, чем значение параметра p , зачерняются. После каждого шага роста определяются новые узлы периметра и с помощью метода релаксации пересчитываются значения ϕ в каждом незанятом узле. Положите $p = 0.35$ и определите структуру узора регулярного фрактала. Чему равна фрактальная размерность? Рассмотрите другие значения p и определите соответствующую фрактальную размерность. Данные узоры называют *фрактальными коврами Лапласа* (Фемили и др.).

13.4. КЛЕТОЧНЫЕ АВТОМАТЫ

В этом разделе мы познакомимся с другим классом геометрических моделей, которые называются *клеточными автоматами*. Отличительной особенностью клеточных автоматов являются их полная дискретность и, следовательно, приспособленность к точному моделированию на цифровом компьютере. Клеточные автоматы были впервые рассмотрены фон Нейманом и Уламом в 1948 г. в качестве возможной идеализации биологического самовоспроизводства. Наш интерес к ним обусловлен, главным образом, тем, что многие модели клеточных автоматов являются примерами простых динамических систем, которые дают упорядоченные узоры, возникающие из случайных начальных условий. Вы можете убедиться в том, что пространственные узоры многих клеточных автоматов напоминают узоры, наблюдаемые в природных явлениях, таких как коагуляция и рост кристаллов. Однако сейчас мы будем рассматривать клеточные автоматы как интересные компьютерные модели, которые сохраняют привлекательные возможности для описания физических систем.

Клеточные автоматы представляют собой математические идеализации физических систем, в которых пространство и время дискретны, а физические величины принимают конечное множество дискретных значений. Представьте регулярную решетку (или «таблицу») ячеек («клеток»), каждая из которых может находиться в конечном числе возможных состояний, например 0 или 1. Состояние системы полностью определяется значениями переменных в каждой клетке. Важными особенностями клеточных автоматов являются следующие:

1. Состояние каждой ячейки обновляется за последовательность *дискретных* шагов по времени.
2. Переменные в каждой ячейке изменяются *одновременно* («синхронно»), исходя из значений переменных на предыдущем шаге.
3. Правило определения нового состояния ячейки зависит только от *локальных* значений в соседних ячейках.

Сначала рассмотрим одномерный автомат с двумя допустимыми значениями переменных в каждой клетке, при этом окружением данной клетки считается она сама и ближайшие клетки справа и слева. Для такого «элементарного» автомата существует 256 возможных правил. На рис. 13.14 иллюстрируется один конкретный набор локальных правил. Свойства всех 256 элементарных одномерных клеточных автоматов каталогизированы (Вольфрам). В задаче 13.10 изучаются некоторые свойства одномерных клеточных автоматов.

ЗАДАЧА 13.10. Одномерные клеточные автоматы

а. Напишите программу, представляющую элементарный одномерный клеточный автомат. Для одновременного изменения состояния клеток проще всего использовать разные массивы для состояний клеток на шагах t и $t + 1$. Примите правило «сложения по модулю два», показанное на рис. 13.14, т.е. значение клетки на шаге $t + 1$ является суммой по модулю 2 ее соседей на шаге t . В качестве начальной выберите конфигурацию с одной ненулевой клеткой (затравка). Достаточно рассмотреть временную эволюцию в течение приблизительно 20 шагов. Получается ли узор из ненулевых клеток самоподобным? Можете ли вы охарактеризовать его фрактальной размерностью?

б. Рассмотрите свойства аналогичного правила, в котором значение клетки на шаге $t + 1$ является суммой по модулю 2 значений ее со-

t	111	110	101	100	011	010	001	000
$t + 1$:	0	1	0	1	1	0	1	0

Рис. 13.14. Пример локального правила временной эволюции одномерного клеточного автомата. Переменные в каждой клетке равны 0 или 1. В верхнем ряду показаны все $2^3 = 8$ возможных комбинаций трех клеток. В нижнем ряду иллюстрируется применение правила временной эволюции и дается значение в центральной клетке на следующем шаге по времени. Данное правило называется 01011010 в двоичной записи (см. второй ряд), правилом сложения по модулю два или правилом 90 в обозначениях Вольфрама. (Заметим, что число 90 является десятичным эквивалентом двоичного числа 01011010, т.е. $90 = 2^1 + 2^3 + 2^4 + 2^6$.)

седей *плюс* ее самой на шаге t . Это правило называется также правилом 10010110 или $150 = 2^1 + 2^2 + 2^4 + 2^7$.

в. Задайте «хаотическую» начальную конфигурацию, в которой каждая клетка находится в состоянии 1 с равной вероятностью 50%. Рассмотрите временную эволюцию правила 00010010 (правило $18 = 2^1 + 2^4$), правила 01001001 (правило $73 = 2^0 + 2^3 + 2^6$), правила 136 (10001000), правила 90 и правила 150. Достаточно рассмотреть 80 клеток на протяжении 60 временных шагов. Чувствительны ли образующиеся узоры к изменению начальных условий? Зависит ли структура узоров от того, используются ли периодические краевые условия или другие? В зависимости от качественного поведения временной эволюции распределите эти автоматы по четырем классам:

1. Стационарное однородное состояние.
2. Узор, состоящий из отдельных периодических областей.
3. Хаотический аперiodический узор.
4. Сложные локализованные структуры.

Результаты задачи 13.10 наводят на мысль, что важным свойством клеточных автоматов является их способность к «самоорганизации». В следующей задаче мы рассмотрим хорошо известный пример двумерного клеточного автомата, который обнаруживает самоорганизацию.

*ЗАДАЧА 13.11. Игра «Жизнь»

Самым известным двумерным клеточным автоматом является игра «Жизнь», придуманная в 1970 г. Джоном Конвеем. Эта игра прославилась своими многочисленными очаровательными узорами и способностью заставлять людей тратить уйму машинного времени на поиски более интересных узоров. Правила игры просты. Для каждой клетки на квадратной решетке определим сумму значений в восьми окружающих ее клетках (рис. 13.15, а). «Живая» клетка (значение 1) «вы-

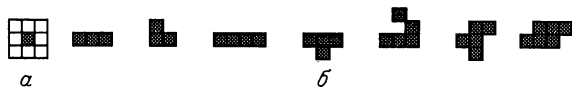


Рис. 13.15. а — «Локальная» окрестность клетки определяется суммой ее восьми соседей; б — примеры начальных конфигураций игры «Жизнь», некоторые из них дают интересные узоры. «Живые» клетки закрашены.

живает» только в том случае, когда эта сумма равна 2 или 3. Если сумма больше 3, то ячейка «умирает» (значение становится равным 0) из-за перенаселения. Если сумма меньше 2, то клетка умирает из-за обособленности. «Мертвая» клетка «оживает» на следующем шаге по времени только если сумма равна 3. Напишите программу, реализующую алгоритм игры «Жизнь». Начните с ряда различных начальных конфигураций (случайных или упорядоченных) и изучите возможные виды узоров, которые могут появиться. Некоторые предлагаемые начальные конфигурации показаны на рис. 13.15,б.

ЗАДАЧА 13.12. Примеры моделей двумерных клеточных автоматов

а. Рассмотрим простой *булев* автомат, каждая клетка которого может быть в состояниях 1 («включено») и 0 («выключено»). Примем «правило голосования», согласно которому состояние клетки в момент времени $t + 1$ определяется голосами ее четырех (на квадратной решетке) ближайших соседей в момент времени t (см. работу Вичняка). Правило состоит в том, что клетка «включается», если включены 2, 3 или 4 ее соседа. Рассмотрите начальные конфигурации, для которых клетка в состоянии 1 появляется с вероятностью p , а клетка в состоянии 0 — с вероятностью $1 - p$. Поскольку правило голосования способствует росту клеток в состоянии 1, интересно начать с конфигурации, в которой таких клеток меньшинство. Положите $p = 0.1$. Что произойдет с изолированными клетками в состоянии 1? Как они растут на первом этапе? Для какой формы (выпуклой или вогнутой) рост кластера из клеток в состоянии 1 прекращается? Что произойдет с кластерами из клеток в состоянии 1, такими, как показано на рис. 13.16? (Если необходимо, создайте такую конфигурацию.) Покажите, что для значения $p = 0.1$ сис-

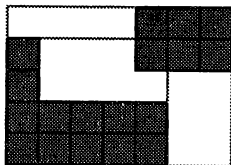


Рис. 13.16. Два прямоугольных перекрывающихся кластера, состоящих из клеток в состоянии 1. Как эволюционируют эти кластеры при правиле «два из четырех» (см. задачу 13.12а)?

тема в конце концов замораживается в виде узора, состоящего из прямоугольных островов клеток в состоянии 1, окруженных морем клеток в состоянии 0. Что произойдет при $p = 0.14$? Можете ли вы определить «критическую плотность» p_c , при которой поведение системы изменяется. Рассмотрите решетки с $L = 128$ и $L = 256$.

б. Рассмотрите клеточный автомат, у которого значение центральной ячейки на следующем временном шаге определяется суммой ее восьми соседей и ее самой (см. работу Вичняка). В частности, положите, что центральная клетка находится в состоянии 1 на следующем шаге по времени, если эта сумма равна или больше 4. Заметим, что это правило стимулирует рост клеток в состоянии 1. Данное правило приводит к явлению, похожему на обнаруженное в п. «а». Рассмотрите начальную конфигурацию, в которой клетки в состоянии 1 появляются с вероятностью p , а клетки в состоянии 0 — с вероятностью $1 - p$. Выберите решетку размером 128×128 и покажите, что для $p = 0.2$ система в конце концов замораживается. Какой формы получаются кластеры из клеток в состоянии 1? Покажите, что если на поверхности наибольшего кластера из клеток в состоянии 1 перевести одну клетку из состояния 0 в состояние 1, то этот кластер будет расти. Каково конечное состояние системы? Как ведет себя система при $p = 0.3$? Можно ли определить «критическую плотность» p_c такую, что для $p \geq p_c$ рост кластеров из клеток в состоянии 1 продолжается до тех пор, пока вся решетка не перейдет в состояние 1? Рассмотрите решетки большего размера и покажите, что значение p_c нечувствительно к размеру решетки. Чему равно вычисленное вами значение p_c в пределе бесконечной решетки?

в. Существует одна проблема, заключающаяся в том, что выводы, которые вы сделали в пп. «а» и «б», являются неточными. Значение p_c равно нулю в пределе бесконечной решетки, и результаты, полученные для конечных решеток, сбивают с толку. Вероятность *любой* конфигурации, состоящей из клеток в состоянии 1, на бесконечной решетке равна единице. Следовательно, где-нибудь на решетке обязательно встретится «критический кластер», который будет расти бесконечно до тех пор, пока вся решетка не перейдет в состояние 1. Мораль этой истории заключается в том, что «нельзя доверять моделированию без знания теории» (изложение высказывания, обычно приписываемого Эддингтону).

13.5. ЗАКЛЮЧЕНИЕ

Мы увидели, что большинство моделей, рассмотренных в этой главе, представлено в виде алгоритма для компьютера, а не в виде дифференциального уравнения. Эти модели служат примером развития «компьютерной культуры» (Вичняк) и отражают влияние техники на способ нашего мышления. Можете ли вы представить большинство рассмотренных моделей без компьютера? Можете ли вы представить себе уяснение сути этих моделей без использования машинной графики?

Геометрическая природа моделей, рассмотренных в гл. 11—13, делает их относительно доступными для тех, чьи знания по физике ограничены. С другой стороны, существует многое, что невозможно понять на этом уровне. Например, после эмпирического «измерения» показателя степени мы могли бы задаться вопросами: «Почему значение показателя именно такое? Как узнать точное значение критического показателя для двумерной перколяции? Можно ли найти более универсальные свойства для большинства моделей кинетического роста, существующих в настоящее время? Могут ли помочь геометрические модели в понимании других физических явлений? Как связаны эти модели с физическими экспериментами? Какие еще величины кроме фрактальной размерности необходимы для описания геометрии объекта?».

Геометрическими моделями помимо физиков и других специалистов интересуются разработчики вычислительной техники. В самом деле, основными доводами в пользу изучения клеточных автоматов является их связь с теориями вычислений и разработкой новых компьютерных архитектур (см. работу Хиллиса).

ЛИТЕРАТУРА

F. Family, D. P. Landau, eds., *Kinetics of Aggregation and Gelation*, North-Holland, 1984. Сборник статей, в котором содержится обилие информации, рисунков и библиографии по целому ряду моделей кинетического роста.

J. M. Hammersley, D. C. Handscomb, *Monte Carlo Methods*, Methuen, 1964. В главе, посвященной процессам протекания, обсуждается алгоритм «роста» для перколяции.

B. Hayes, *Computer recreations*, *Sci. Amer.* **250**, 12 (March 1984). Простое введение в теорию клеточных автоматов.

H. J. Herrmann, Geometrical Cluster Growth Models and Kinetic Gelation, Phys. Repts. **136**, 154 (1986).

W. D. Hillis, The Connection Mashine, MIT Press, 1985. Обсуждение новой «полностью параллельной» архитектуры компьютера, на которую частично повлияли физические модели, обсуждаемые в этой главе.

B. B. Mandelbrot, The Fractal Geometry of Nature, W. H. Freeman, 1983. Важная и прекрасно иллюстрированная книга по фракталам.

I. Majid, *D. Ben-Avraham*, *S. Havlin*, *H. E. Stanley*, Exact-enumeration approach to random walks on percolation clusters in two dimensions, Phys. Rev. **B30**, 1626 (1984).

L. Niemeyer, *L. Pietronero*, *H. J. Wiesmann*, Fractal dimension of dielectric breakdown, Phys. Rev. Letts. **52**, 1033 (1984). См. также: *F. Family*, *Y. C. Zhang*, *T. Vicsek*, Invasion percolation in an external field: dielectric breakdown in random media, J. Phys. A. **19**, L733 (1986). Близкая модель предложена для роста снежных хлопьев путем образования льда в переохлажденном паре. См. *F. Family*, *D. E. Platt*, *T. Vicsek*, Deterministic growth model of pattern formation in dendritic solidification, Emory University preprint (1987). В этом препринте авторы обсуждают также структуру фрактальных ковров Лапласа.

H. O. Peitgen, *P. H. Richter*, The Beauty of Fractals, Springer-Verlag, 1986.

L. Pietronero, *E. Tosatti*, eds., Fractals in Physics, North-Holland, 1986. Сборник статей, большинство из которых доступно целенаправленному читателю.

W. Poundstone, The Recursive Universe, Contemporary Books, 1985. Книга, основанная на игре «Жизнь», в которой делается попытка провести аналогии между узорами игры и понятиями теории информации и космологии. Другая литература об игре «Жизнь»: *E. R. Berlekamp*, *J. H. Conway*, *R. K. Guy*, Winning Ways for Your Mathematical Plays, Vol. 2, Academic Press, 1984 и *M. Gardner*, Wheels, Life and other Mathematical Amusements, Freeman Press, 1983.

M. Przyborowski, *M. van Woerkom*, Diffusion of many interacting random walkers on three-dimensionel lattice with a personel computer, Eur. J. Phys. **6**, 242 (1985). Эта работа была выполнена авторами в то время, когда они были студентами в Западной Германии.

F. Reif, Fundamentals of Statistical and Thermal Physics, McGraw-Hill, 1965. В гл. 15 обсуждается соотношение Эйнштейна между диффузией и подвижностью.

L. M. Sander, Fractal growth, *Sci. Amer.* **256**, 94 (January 1987).
Доступное введение в модели фрактального роста и их приложения, принадлежащее одному из авторов модели агрегации с ограничением диффузии (см. задачу 13.8).

L. Schulman, *P. Seiden*, Statistical Mechanics of dynamical system based on Conway's game of Life, *J. Stat. Phys.* **19**, 293 (1978).

H. E. Stanley, *N. Ostrowsky*, eds., On Growth and Form, Martinus Nijhoff Publishers, Netherlands, 1986. Сборник статей примерно того же уровня, что и сборник под ред. Фемели и Ландау. На статью *P. Meakin*, посвященную АОД, мы ссылались в тексте.

D. D. Thornburg, Discovering Logo, Addison-Wesley, 1983. Эта книга более точно характеризуется подзаголовком An Invitation to the Art and Pattern of Nature. В книге на множестве простых примеров рассматриваются свойства рекурсивных процедур и фракталов.

G. Y. Vichniac, Cellular automata models of disorder and organization, in Disordered Systems and Biological Organization, *E. Bienenstock*, *F. F. Soulie*, *G. Weisbuch*, eds., Springer-Verlag, 1986. См. также: *G. Y. Vichniac*, Taking the computer seriously in teaching science (an introduction to cellular automata), in Microscience, Proceedings of the UNESCO Workshop on Microcomputers in Science Education, *G. Marx*, *P. Szucs*, eds., Balaton, Hungary, 1985.

S. Wolfram, ed., Theory and Applications of Cellular Automata, World Scientific, Singapore, 1986. Сборник статей по клеточным автоматам разного уровня сложности. Приведена обширная библиография.

D. Wilkinson, *J. F. Willemsen*, Invasion percolation: a new form of percolation Theory, *J. Phys.* **A16**, 3365 (1983). Одной из первых работ по оккупирующей перколяции была статья *R. Lenormand*, *S. Bories*, Description of a bond percolation mechanism used for the simulation of drainage with trapping in porous media (на фр. языке), *C. R. Acad. Sci. Paris* **B291**, 279 (1980).

ДОПОЛНИТЕЛЬНАЯ ЛИТЕРАТУРА

Эфрос А. Л., Физика и геометрия беспорядка. — М.: Наука, 1982.

Шкловский Б. И., *Эфрос А. Л.*, Электронные свойства легированных полупроводников. — М.: Наука, 1979. Гл. 5 посвящена перколяции.

Шустер Г., Детерминированный хаос: введение. — М.: Мир, 1988. В начальных главах рассматриваются фракталы и фрактальная размерность.

ПРИБЛИЖЕНИЕ К РАВНОВЕСИЮ

14

Мы рассматриваем простую модель макроскопической системы и вводим понятия равновесия и энтропии.

14.1 ВВЕДЕНИЕ

В этой главе мы воспользуемся далее нашими знаниями законов микро-скопической физики, чтобы понять качественные свойства макроскопических объектов. Типичные макроскопические системы охватывают столь несхожие объекты, как газы, жидкости, твердые тела, полимеры, гели и биологические организмы, и содержат порядка 10^{20} – 10^{25} взаимодействующих частиц. Поведение макроскопических систем отличается большим разнообразием. Это и турбулентное движение быстро текущей речейки, и плавление льда, и размножение живого организма. Колоссальную сложность, присущую макроскопическим системам, нельзя понять путем прямого применения компьютеров—просчитать движение 10^{20} взаимодействующих частиц нельзя даже на самых супер из суперкомпьютеров. Более того, детальная информация о поведении отдельных частиц не много бы дала для понимания основных свойств макроскопической системы.

Одним из наиболее важных свойств макроскопических систем является их стремление к беспорядку. А именно, если вначале частицы упорядочены и система изолирована от внешних воздействий, то после удаления внутренних связей частицы будут стремиться прийти в беспорядок. Пример этой тенденции можно наблюдать, добавив чернил в стакан воды. Допустим, что чернила имеют одинаковую с водой плотность и их осторожно льют на поверхность воды. Стакан стоит неподвижно, и внешние условия не меняются. Мы знаем, что по прошествии некоторого времени чернила и вода полностью перемешаются. Предположим, что мы сняли фильм о системе вода—чернила и случайно запустили его задом наперед. Мы увидели бы, что случайное движение молекул чернил вынесло все чернила на поверхность. Наше интуитивное представление о действии законов природы подсказало бы, что здесь что-то не то—и правда, ведь фильм прокручивается задом наперед. Исходя из своего опыта, мы можем сказать, что природная тенденция макроскопических систем к беспорядку определяет направление или «вектор» времени.

Смесь чернил с водой может быть описана путем задания ее *микро-скопического состояния*, или *микросостояния*. Такое описание отвечает наиболее полной, совместимой с законами механики характеристике всех молекул системы. Согласно классической механике, такое описание соответствует заданию координат и скоростей каждой молекулы. В крупном масштабе *макроскопическое состояние*, или *макросостояние*, смеси чернил с водой может быть описано путем задания средней концентрации молекул в любой малой, но конечной области стакана воды.

Микроскопическое состояние системы меняется самым невообразимым образом. В противоположность этому мы знаем, что макроскопическое состояние через какое-то время перестает меняться со временем, если не говорить о малых случайных *флуктуациях*. Говорят, что такое состояние является *равновесным*.

14.2. ПРОСТАЯ МОДЕЛЬ

Понятия беспорядка и порядка, направления времени, а также микроскопического и макроскопического описаний системы можно проиллюстрировать на простой модели, которую мы будем рассматривать в этой главе в разнообразных контекстах. Рассмотрим идеальный газ N тождественных частиц, взаимодействием между которыми можно пренебречь. Такой газ можно приготовить в лабораторных условиях, если сделать его достаточно разреженным. В этом случае вероятность взаимодействия частиц газа друг с другом мала. Предположим, что газ заключен в сосуд, который изолирован (не подвержен влиянию внешней системы или силы) и пребывал в покое в течение долгого времени. Давайте внимательно посмотрим на распределение частиц в пространстве и представим себе, что сосуд разделен перегородкой. Перегородка содержит небольшое отверстие, закрываемое подвижной заслонкой (рис. 14.1). Вначале в левой половине сосуда находятся n частиц и n' — в правой половине, причем $n + n' = N$. Затем отверстие открывается. Поскольку движение каждой частицы происходит независимо от всех остальных частиц, вероятность прохождения траектории частицы через отверстие

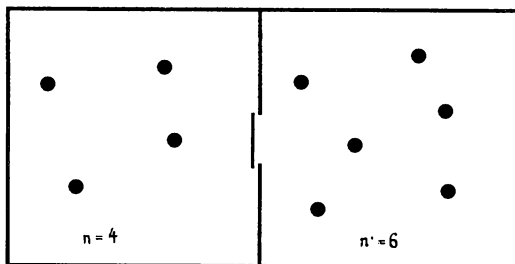


Рис. 14.1. Ящик поделен на две равные половины перегородкой, содержащей небольшое отверстие с подвижной заслонкой. После открывания отверстия через него проходит за каждый шаг по времени одна частица.

для всех частиц одинакова. Будем предполагать, что в единицу времени через отверстие проходит одна частица.

Для определения микросостояний и макросостояний нашей модели будем считать, что частицы в чем-то различны и пронумерованы от 1 до N . Микросостояние можно описывать номерами частиц в левой половине сосуда, ибо коль скоро номера слева известны, тем самым определены и номера справа. Макросостояние характеризуется числом частиц n с левой стороны.

Какие вопросы, возможно, было бы уместно задать о поведении данной простой модели «частиц в ящике»? Один вопрос очевиден: «Если известны n и n' при $t = 0$, то чему равны n и n' в момент времени t (после t перемещений)?» Ясно, что ответ на этот вопрос носит статистический характер. Мы могли бы тот же вопрос задать иначе и спросить: «Чему равно среднее число частиц в левой половине сосуда в момент времени t ?». Дальнейшая детализация, возможно, потребовала бы знания $P_N(n, t)$ — вероятности нахождения в момент времени t слева n частиц из N .

14.3. ТОЧНЫЙ ПЕРЕБОР

Теперь, когда мы представляем себе характер модели и постановку некоторых вопросов, какие методы решения нам подойдут? Один из методов — использовать *точный перебор* и определять все комбинации на каждом шаге по времени. Например, предположим, что при $t = 0$ $n = 10$ и $n' = 0$. При $t = 1$ возможна единственная комбинация: $n = 9$ и $n' = 1$, и отсюда $P(n = 9, t = 1) = 1$. При $t = 2$ возможны комбинации, когда одна из девяти частиц слева перемещается направо или же частица справа возвращается в левую половину. Поскольку первая комбинация может реализовываться девятью разными способами, мы имеем ненулевые вероятности

$$P(8, 2) = \frac{9}{10}, \quad P(10, 2) = \frac{1}{10}.$$

Следовательно, при $t = 2$ среднее число частиц в левой половине ящика составляет

$$\langle n \rangle = 8P(8, 2) + 10P(10, 2) = 8.2.$$

На следующем шаге по времени мы имеем $P(7, 3) = \frac{8}{10} P(8, 2) = \frac{72}{100}$, что

отвечает перемещению одной из восьми частиц направо. Те же соображения дают $P(9,3) = \frac{10}{10} P(10,2) + \frac{2}{10} P(8,2) = \frac{28}{100}$. Отсюда при $t = 3$ получаем

$$\langle n \rangle = 7P(7,3) + 9P(9,3) = 7.56.$$

Поскольку полное число частиц мало, такой перебор удастся проделать еще для нескольких временных шагов. Однако, поскольку количество комбинаций увеличивается с t , выполнить расчет для больших t уже отнюдь не легко. Для больших N можно легко подсчитать еще меньше шагов по времени. Чтобы справиться с этой трудностью, можно обратиться к компьютеру для вычисления всех различных комбинаций. Но при достаточно больших N или t число комбинаций становится столь огромным, что никакой компьютер не сумеет их рассчитать.

14.4. МЕТОД МОНТЕ-КАРЛО

В противоположность методу точного перебора метод Монте-Карло годится для больших систем и больших времен. В методе Монте-Карло мы генерируем выборку случайных перемещений и считаем, что наша выборка является характерным представителем множества всех возможных перемещений (репрезентативная выборка). Понятно, что чем больше выборки мы имеем, тем ближе мы подойдем к точному результату.

Для реализации метода Монте-Карло нам необходимо знать вероятность перехода частицы из левой половины ящика в правую. Поскольку у каждой частицы имеется одинаковый шанс оказаться очередным кандидатом на проход через отверстие, вероятность перемещения слева направо в единицу времени равна числу частиц в левой половине в данный момент времени, поделенному на полное число частиц. Таким образом, вероятность одного перехода слева направо равна n/N . Теперь мы можем моделировать временную эволюцию модели в соответствии с указанной вероятностью. Алгоритм состоит из следующих шагов:

1. Из набора равномерно распределенных в интервале от 0 до 1 случайных чисел формируем случайное число r .
2. Сравниваем r с текущим значением доли частиц n/N в левой половине ящика.
3. Перемещаем частицу слева направо, если $r \leq n/N$; в противном случае перемещаем частицу справа налево.

В программе **box** реализован этот алгоритм и строится график эволюции n .

```

PROGRAM box
RANDOMIZE
CALL initial(N,tmax)           ! ввод данных
CALL move(N,tmax)              ! перемещение частицы сквозь отверстие
END

SUB initial(N,tmax)
  INPUT prompt "число частиц = ": N           ! попробуйте N = 1000
  LET tmax = 10*N
  SET window -0.1*tmax,1.1*tmax,-0.1*N,1.1*N
  BOX LINES 0,tmax,0,N
  PLOT 0,N;
END SUB

SUB move(N,tmax)
  LET nl = N                           ! сначала все частицы с левой стороны
  FOR itime = 1 to tmax
    LET prob = nl/N
    ! генерируем случайное число и перемещаем частицу
    IF rnd <= prob then
      LET nl = nl - 1
    ELSE
      LET nl = nl + 1
    END IF
    PLOT itime,nl;
  NEXT itime
END SUB

```

ЗАДАЧА 14.1. Модель «частиц в ящике»

а. Опишите временную эволюцию числа частиц в левой половине ящика. Рассмотрите $N = 10, 20, 40$ и 80 , при этом задавайте все N частиц в начальный момент в левой половине.

б. Как качественно характеризуется равновесие? Эволюционирует ли система к равновесию? Можете ли вы определить направление времени? Какое приблизительно время требуется системе, чтобы достигнуть равновесия?

в. Модифицируйте программу **box**, чтобы число частиц с левой стороны усреднялось по нескольким испытаниям. Какое определение вероятности используется при получении среднего числа $\langle n(t) \rangle$ на каждом шаге по времени t ? Совпадает ли это определение с использованным при рассмотрении точного перебора в § 14.3? Если оба определения неодинаковы, то равносильны ли они?

г. Одним из способов убедиться в том, что модифицированная вами программа **box** работает правильно, является сравнение результатов для $\langle n(t) \rangle$, полученных методом Монте-Карло и точным перебором. Возьмите какое-нибудь небольшое значение N , например $N = 4$, и применяя точный перебор, вычислите $\langle n(t) \rangle$ точно для нескольких шагов по времени. Затем возьмите достаточно большое число испытаний, чтобы получить правдоподобную оценку для $\langle n(t) \rangle$, и убедитесь в том, что ваши результаты совпадают.

д. Пропустите модифицированную вами программу **box** с $N = 10$, причем все частицы в начальный момент времени поместите в левую половину ящика. Нарисуйте график зависимости n — числа частиц с левой стороны — от времени для пяти разных испытаний. Затем нарисуйте n как функцию времени, усреднив по тем же пяти испытаниям. Сравните временную зависимость n для одного испытания с зависимостью n , усредненной по нескольким испытаниям.

е. Мерой равновесных флуктуаций служит дисперсия σ^2 , определяемая равенством

$$\begin{aligned}\sigma^2 &= \langle (n - \langle n \rangle)^2 \rangle = \\ &= \langle n^2 \rangle - \langle n \rangle^2.\end{aligned}\tag{14.1}$$

Угловые скобки показывают усреднение по времени, проведенное после того, как система пришла в равновесное состояние. Относительная величина флуктуаций составляет $\sigma/\langle n \rangle$. Модифицируйте программу **box** так, чтобы после достижения системой равновесия вычислялась относительная флуктуация в σ .

ж. Определите время установления равновесного состояния, $\langle n \rangle$ и $\sigma/\langle n \rangle$ для $N = 20, 40$ и 80 . Как эти различные физические величины зависят от N ?

14.5. ЭНТРОПИЯ

Вычислительные эксперименты, проведенные в задаче 14.1, иллюстрируют фундаментальное свойство систем многих частиц: любая изолированная система, будучи приготовлена в неслучайном или упорядоченном состоянии, будет изменяться со временем таким образом, чтобы прийти в состояние максимального беспорядка. В этом состоянии максимального беспорядка все макроскопические величины не зависят от времени, не считая малых флуктуаций, и говорят, что такая система находится в термодинамическом равновесии.

Благодаря простоте нашей модели мы можем подсчитать сколько микросостояний существует у каждого макросостояния. Нам известно, что каждая частица может пребывать в одном из двух состояний—быть слева или справа. Кроме того, местонахождение каждой частицы не зависит от всех остальных. Эти два условия означают, что если всего имеется N частиц, причем n слева и n' справа, то количество микросостояний, отвечающих каждому макросостоянию, равно $N!/n!n'!$, т.е. определяется биномиальным распределением. В табл. 14.1 мы приведем число микросостояний Ω_n для $N = 10$ и различных значений n . Обратите внимание на то, что максимальное число микросостояний приходится на $n = 5$.

ТАБЛИЦА 14.1. Число микросостояний Ω_n и энтропия $S_n = \ln \Omega_n$ для модели «частиц в ящике» с $N = 10$. Макросостояние характеризуется числом частиц с левой стороны и задается величиной n . Полное число микросостояний для $N = 10$ составляет $2^{10} = 1024$

n	Ω_n	$\ln \Omega_n$
0	1	0.00
1	10	2.30
2	45	3.81
3	120	4.79
4	210	5.35
5	252	5.53
6	210	5.35
7	120	4.79
8	45	3.81
9	10	2.30
10	1	0.00

Порядку и беспорядку можно придать еще одну количественную меру. В задаче 14.1 мы нашли, что для $N = 10$ равновесному макросостоянию отвечает $n = 5$, т.е. макросостояние с наибольшим числом микросостояний. Мы говорим, что это макросостояние отвечает состоянию «максимального беспорядка». По сравнению с этим макросостояние с $n = 0$ отвечает состоянию нулевого беспорядка, поскольку систему можно обнаружить только в одном микросостоянии. В качестве характеристики степени беспорядка удобно ввести *энтропию* S , определяя ее так, чтобы при существовании только одного микросостояния энтропия равнялась нулю и возрастала с увеличением числа микросостояний. Согласно Больцману, энтропия S системы, пребывающей в определенном макросостоянии, есть по определению (см. учебник Райфа)

$$S_n/k_B = \ln \Omega_n, \quad (14.2)$$

где Ω_n суть полное число всевозможных микросостояний, соответствующих n . Включение в (14.2) постоянной k_B обеспечивает согласование этого определения с термодинамическим определением S . Поскольку в данном случае присутствие этой постоянной ни на что не влияет, мы будем измерять S в единицах k_B , т.е. полагаем k_B равной единице.

Согласно (14.2), энтропия системы логарифмически зависит от числа микросостояний, отвечающих данному макросостоянию. В табл. 14.1 приведена энтропия, отвечающая каждому макросостоянию. Видно, что S_n максимальна для $n = 5$. Говорят, что такое состояние имеет максимальный беспорядок, поскольку число микросостояний, соответствующих данному макросостоянию, максимально.

Определение энтропии (14.2) требует перебора всех возможных микросостояний, что в общем случае является неразрешимой задачей. Желательно иметь такой метод измерения энтропии, который опирается на наблюдаемые характеристики системы, а не на идеализированное описание, требующее перебора всех возможных микросостояний. Ма предложил определение энтропии, пригодное для непосредственного измерения в численном эксперименте (см. список литературы). В этом определении энтропии использован тот факт, что всякая система в процессе своей эволюции во времени в конце концов обязательно повторит любое микросостояние (или близкое микросостояние). Чем больше проходит времени до совпадения двух некоррелированных микросостояний, тем меньше имеется микросостояний и, следовательно, меньше энтропия системы. Величина, которую мы можем непосредственно измерить—это частота совпадения R_n , определяемая как отношение числа совпавших пар

ТАБЛИЦА 14.2. Последовательность двадцати микросостояний для макросостояния $n = 1$. Каждое из двадцати микросостояний характеризуется номером частицы с левой стороны

8	7	5	10	7	2	4	6	2	10	3	4	3	9	6	5	2	9	2	4
---	---	---	----	---	---	---	---	---	----	---	---	---	---	---	---	---	---	---	---

микросостояний к полному числу сравнений двух макросостояний. Энтропия определяется через R_n в виде

$$S_n/k_B = \ln \frac{1}{R_n}. \quad (14.3)$$

Соотношение (14.3) служит методической основой метода совпадений. Измерение R_n следует производить на протяжении относительно долгого времени. По мере увеличения времени результаты должны сходиться к точному значению R_n .

Чтобы понять реализацию определения S в виде (14.3), применим ее к модели «частиц в ящике». Поскольку (14.3) применимо только к системе, находящейся в определенном макросостоянии, мы должны обеспечить постоянство числа частиц с каждой стороны. Простейшая модификация нашего алгоритма, удовлетворяющая данному условию, основана на предположении, что система эволюционирует путем *обмена* одной частицы слева с одной частицей справа. По сравнению с этим наше прежнее моделирование допускало, чтобы n флуктуировало за счет случайных *перемещений* частиц с одной стороны на другую.

В табл. 14.2 показана последовательность двадцати обменов для $n = 1$; номера частиц соответствуют частицам с левой стороны. Так как общее число обменов равно двадцати, существует $20(20-1)/2 = 190$ возможных вариантов сравнений макросостояний. Число совпадений для каждого микросостояния составляет $m(m-1)/2$, где m — число появлений данного микросостояния. Из табл. 14.2 видно, что для микросостояния 2 величина $m = 4$, и отсюда число возможных совпадений этого микросостояния равно $(4)(3)/2 = 6$. Количество совпадений для микросостояний приведено в табл. 14.3. Поскольку полное число совпадений равно пятнадцати, частота совпадений получается равной $R = 15/190$ и, следовательно, $S \sim \ln 190/15 \approx 2.5$. Этот приближенный результат для S согласуется с точным результатом $S = \ln 10 = 2.3$, приведенным в табл. 14.1.

Отметим, что для $n = 2$ число возможных микросостояний приблизительно в четыре раза больше, чем для $n = 1$. Поэтому нам нужен ка-

ТАБЛИЦА 14.3. Совпадения десяти возможных микросостояний, отвечающих макросостоянию $n = 1$ (одна частица с левой стороны ящика), для последовательности из двадцати обменов. Найдено, что полное число совпадений для этой последовательности равно пятнадцати

Микросостояние	Число событий	Число совпадений
1	0	0
2	4	6
3	2	1
4	3	3
5	2	1
6	2	1
7	2	1
8	1	0
9	2	1
10	2	1

ТАБЛИЦА 14.4. Последовательность двадцати микросостояний для макросостояния $n = 2$. Каждое микросостояние однозначно определяется величиной $micro = 2^{L_1} + 2^{L_2}$, где L_1 и L_2 — номера обеих частиц с левой стороны

Испытание	L_1	L_2	$micro$	Испытание	L_1	L_2	$micro$
1	6	9	576	11	3	10	1032
2	4	9	528	12	3	5	40
3	4	10	1040	13	5	7	160
4	3	10	1032	14	5	6	96
5	2	10	1028	15	5	10	1056
6	2	3	12	16	5	7	160
7	3	10	1032	17	5	6	96
8	5	10	1056	18	5	8	288
9	1	5	34	19	2	5	36
10	1	10	1026	20	2	8	260

кой-нибудь простой метод нумерации и сравнения микросостояний. Один метод заключается в описании каждого микросостояния величиной $micro$, определяемой равенством

$$micro = \sum 2^L, \quad (14.4)$$

где L — номер каждой частицы и сумма берется по всем частицам с левой стороны. Заметим, что сумму (14.4) можно рассматривать как двоичное число. (В данном случае переменная $micro$ принимает значение

1032 три раза и по два раза 1056, 96 и 160.) Отсюда $R = 6/190$ и $S \sim \ln 190/6 \approx 3.45$. Эта оценка для S согласуется с точностью 10% с точным значением $S = \ln 45 \approx 3.81$ (см. табл. 14.1).

Следующая программа вычисляет энтропию для произвольного n , используя описанную выше процедуру сравнения микросостояний.

```

PROGRAM entropy
DIM left(10),right(10),micro(0 to 2000)
RANDOMIZE
! ввод параметров и выбор начальной конфигурации частиц
CALL initial(nl,nr, left, right, micro, nexch)
CALL exchange(nl,nr, nexch, left, right, micro)           ! обмен частиц
CALL output(nexch, micro)           ! расчет частоты совпадений и энтропии
END

SUB initial (nl,nr, left(),right(), micro(), nexch)
! задание макросостояния
INPUT prompt "полное число частиц = ": N
INPUT prompt "число частиц слева = ": nl
LET nr = N - nl           ! число частиц с правой стороны
LET micro(0) = 0
FOR il = 1 to nl
    LET left(il) = il           ! список номеров частиц с левой стороны
    LET micro(0) = micro(0) + 2^il           ! начальное микросостояние
NEXT il
FOR ir = 1 to nr
    LET right(ir) = ir + nl ! список номеров частиц с правой стороны
NEXT ir
INPUT prompt "число обменов = ": nexch
END SUB

```

```

SUB exchange (nl,nr,nexch, left(),right(),micro())
  ! обмен частицы слева с номером lindex с частицей справа
  !   с номером rindex
  FOR iexch = 1 to nexch
    ! индексы массивов выбираем случайным образом
    LET lindex = int(rnd*nl + 1)      ! индекс массива частиц слева
    LET rindex = int(rnd*nr + 1)      ! индекс массива частиц справа
    LET left_particle = left(lindex)
    LET right_particle = right(rindex)
    LET left(lindex) = right_particle      ! новый номер частицы слева
    LET right(rindex) = left_particle      ! новый номер частицы справа
    LET micro(iexch) = micro(iexch - 1) + 2^right_particle
    LET micro(iexch) = micro(iexch) - 2^left_particle ! новое микросостояние
  NEXT iexch
END SUB

SUB output (nexch,micro())
  ! вычисление частоты совпадений и энтропии
  LET ncomparisons = nexch*(nexch - 1)/2      ! полное число сравнений
  ! сравниваем микросостояния
  FOR iexch = 1 to nexch - 1
    FOR jexch = iexch + 1 to nexch
      IF micro(iexch) = micro(jexch) then
        LET ncoincidences = ncoincidences + 1      ! число совпадений
      END IF
    NEXT jexch
  NEXT iexch
  LET rate = ncoincidences/ncomparisons      ! частота совпадений
  IF rate > 0 then LET S = log(1/rate)
  PRINT "оценка энтропии = ",S
END SUB

```

ЗАДАЧА 14.2. Динамическое определение энтропии

- а. Возьмите $N = 10$ и с помощью программы **entropy** вычислите частоту совпадений R_n и энтропию S_n для каждого макросостояния модели частиц в ящике.
- б. Сравните полученные приближенные результаты для S_n с точными, приведенными в табл. 14.1.

в. Оцените погрешность своих вычислений S_n , пропустив программу с более длинными последовательностями.

г. Если возможно, повторите все указанные выше вычисления для ббльших значений N . Какое наибольшее значение N можно рассматривать, используя величину *micro*, как она определена в (14.4) и в программе **entropy**? Существует ли практическое ограничение сверху на значения N , которое можно рассматривать?

В программе **entropy** формирование конфигураций осуществляется гораздо быстрее, чем подсчет совпадений. Первое пропорционально числу обменов, а второе пропорционально квадрату числа обменов. Поэтому, вероятно, было бы удобно запомнить значения *micro*, порождаемые подпрограммой **exchange**, в отдельном файле. Таким образом мы можем отделить анализ данных от их формирования и при необходимости, использовать различные методы анализа. В языке True BASIC файл открывается с помощью инструкции:

```
OPEN #1: name "data1", access output, create new
```

Запись значений *micro* в файл #1 производится инструкцией:

```
PRINT #1: micro(iexch)
```

Файл #1 закрывается инструкцией:

```
CLOSE #1
```

Чтобы прочесть данные, используем инструкции:

```
OPEN #1: name "data1", access input  
INPUT #1: micro(iexch)  
CLOSE #1
```

14.6. ВЛИЯНИЕ КОРРЕЛЯЦИИ

Вероятно, вы обратили внимание на то, что в табл. 14.4 частица 5 находится с левой стороны в течение относительно долгого времени. Что касается данного конкретного примера, то эта кратковременная

корреляция обусловлена корреляциями в датчике случайных чисел (не тот датчик в True BASIC). Такой вид кратковременных корреляций встречается также в реальных системах. Однако полезность определения энтропии (14.3) зависит от некоррелированности микросостояний. По-настоящему надо бы сравнивать только микросостояния, которые отстоят по времени дальше, чем на *время корреляции*, т.е. среднее время между двумя такими микросостояниями, чтобы они были некоррелированы. В нашем примере время корреляции есть количество обменов, необходимое, чтобы система перешла из одного микросостояния в другое, где последнее не имеет почти никаких общих частиц с первым. Одной из оценок времени корреляции служит время, которое требуется системе для достижения термодинамического равновесия. В следующей задаче мы посмотрим, насколько важны корреляции при вычислении энтропии в нашей простой модели.

*ЗАДАЧА 14.3. Энтропия и корреляции

а. Модифицируйте программу `entropy` таким образом, чтобы не производить никаких сравнений состояний, отстоящих по времени ближе, чем на время установления равновесия (см. задачу 14.1). Вычислите S_n для $N = 10$ с различными n , учитывая влияние корреляций.

б. Что говорят вам результаты, полученные в п.«а», о времени корреляции, если энтропия, найденная в п.«а», совпадает с вычисленной в задаче 14.2?

14.7. РАВНОВЕСНАЯ ЭНТРОПИЯ

Определение энтропии (14.3) применимо к системе, о которой известно, что она находится в определенном макросостоянии. Однако в нашей первоначальной формулировке модели частиц в ящике мы предусматривали случайное перемещение частиц из одной половины в другую и, следовательно, допускали флуктуации n . Для вычисления энтропии системы, находящейся в равновесном состоянии, т.е. в условиях нефиксированного макросостояния, мы обобщим определение (14.3) и запишем

$$S = \sum_n P_n S_n, \quad (14.5)$$

где P_n — равновесная вероятность появления макросостояния n ; суммирование производится по всем макросостояниям.

ЗАДАЧА 14.4. Равновесная энтропия

а. Для модели частиц в ящике вероятность P_n есть вероятность появления макросостояния с n частицами слева. P_n может быть вычислена точно путем подсчета числа микросостояний, отвечающих макросостоянию n , и деления на полное число возможных микросостояний. Воспользуйтесь результатами табл. 14.1 и вычислите P_n для $N = 10$.

б. Простая процедура вычисления P_n по последовательности состояний, заключается в том, что подсчитывается количество появлений каждого макросостояния и эта сумма делится на полное число событий. Модифицируйте программу **box**, чтобы вычислялась P_n . Возьмите $N = 10$ и сравните свои результаты для P_n с точными, найденными в п.«а».

в. Вычислите равновесную энтропию, используя соотношение (14.5) и вероятности P_n и S_n , подсчитанные в п.«б» и задаче 14.2 соответственно.

г. Как соотносятся ваши результаты для S , найденные в п.«в», с энтропией S_n для $n = 5$? Как бы соотносилась равновесная энтропия с S_n для $n = N/2$, если $N \gg 1$?

*14.8. ЭНТРОПИЯ И ХАОС

В гл. 7 мы рассматривали простую модель удвоения периода, поведение которой напоминало наступление хаоса в турбулентных системах. Эта модель описывается разностным уравнением

$$x_{n+1} = 4rx_n(1 - x_n). \quad (14.6)$$

Мы нашли, что отображение (14.6) проявляет периодическое либо хаотическое поведение в зависимости от величины управляющего параметра r . Возможно, применение метода Ма вычисления энтропии позволит

глубже понять природу области хаоса. Мы разбиваем отрезок $[0,1]$ на N равных частей и описываем каждое микросостояние модели величиной x ; макроскопическое состояние модели характеризуется параметром r . Энтропия вычисляется путем подсчета числа совпадений. Для $r < r_c$ и достаточно больших N мы предполагаем, что энтропия будет достигать некоторого предельного значения, не зависящего от N . Как зависит от N энтропия при $r > r_c$? Если микроскопическое состояние никогда не повторяется, мы предполагаем, что энтропия будет возрастающей функцией от N и не достигнет предельного значения.

ЗАДАЧА 14.5. Энтропия системы с периодическим удвоением

а. Составьте подпрограмму, которая вычисляет частоту совпадений для квадратичного отображения (14.6). Выберите такое значение r , чтобы x_n было периодическим, например $r = 0.85$, и удостоверьтесь, что метод совпадений Ма правильно определяет число возможных значений x_n . Покажите, что для больших N энтропия не зависит от N . Влияет ли на ваши результаты тот факт, что операции с числами компьютер производит с конечным числом знаков?

б. Рассмотрите макросостояние $r = 1$, относительно которого известно, что между $x = 0$ и 1 нет никаких областей, где не побывала бы x_n . Такой выбор r соответствует области «максимального» хаоса. Определите энтропию как функцию от N .

в. Для определенных значений $r > r_c$ (например, $r = 0.91$) траектория x кажется хаотичной, но, возможно, имеются области x , которые могут быть непосещаемы. Определите энтропию в этой области как функцию от N . Влияет ли на ваши результаты в этой области то, что компьютер оперирует с числами с конечным числом знаков? Чтобы отличить «истинно» хаотическую область от «почти» хаотической, потребовалось бы детальное и кропотливое исследование.

ЛИТЕРАТУРА

P.W. Atkins, The Second Law, W.H.Freeman, 1984. Прекрасно иллюстрированная книга, в которой рассматриваются понятия энтропии и беспорядка с самых различных сторон.

R. Baierlein, Atoms and Information Theory, W.H.Freeman, 1971.

Современный учебник для младшекурсников, в котором статистическая механика излагается с позиций теории информации.

R. M. Eisberg, *Applied Mathematical Physics with Programmable Pocket Calculators*, McGraw-Hill, 1976. В гл. 7 обсуждается энтропия и направление времени.

R. M. Eisberg, *L. S. Lerner*, *Physics*, Vol. II, McGraw-Hill, 1981. В гл. 18 см. обсуждение беспорядка и энтропии.

Shang-keng Ma, *Calculation of entropy from data of motion*, *J. Stat. Phys.* **26**, 221 (1981). См. также гл. 25 учебника для средних курсов того же автора: *Statistical Mechanics*, World Scientific, 1985. Заметим, что частота совпадений связана с временем возврата, представляющим собой время, за которое конечная система вернется в сколь угодно малую окрестность почти любого заданного начального состояния.

F. Reif, *Statistical and Thermal Physics*, *Berkeley Physics*, Vol. 5, McGraw-Hill, 1965. [Имеется перевод: *Райф Ф.*, *Статистическая физика*. — М.: Наука, 1986.]

C. R. Smith, *W. T. Grandt, Jr.*, eds., *Maximum-Entropy and Bayesian Methods in Inverse Problems*, D. Reidel, 1985. Сб. статей по применению принципа максимальной энтропии. Одна из наиболее доступных статей касается восстановления изображений.

ДОПОЛНИТЕЛЬНАЯ ЛИТЕРАТУРА

Матвеев А. Н., *Молекулярная физика*. — М.: Высшая школа, 1981. В этой книге рассматриваются многие вопросы, затронутые в данной главе.

Климонтович Ю. Л., *Статистическая физика*. — М.: Наука, 1982. В гл. 3 и 4 обсуждаются различные аспекты равновесного состояния.

Шустер Г., *Детерминированный хаос: Введение*. — М.: Мир, 1988. В этой книге рассматриваются проблемы хаоса и энтропии.

Методы Монте-Карло в статистической физике/Под ред. *К. Биндера*. — М.: Мир, 1982. Обсуждается метод Монте-Карло и его приложения к различным задачам термодинамики и статистической физики.

МИКРОКАНОНИЧЕСКИЙ АНСАМБЛЬ

В этой главе мы моделируем микроканонический ансамбль и «открываем» распределение Больцмана для систем, находящихся в тепловом контакте с тепловым резервуаром.

15.1. ВВЕДЕНИЕ

Моделирование молекулярной динамики в гл. 6 и задача «частиц в ящике» из гл. 14 дали нам представление о некоторых качественных свойствах макроскопических систем, например необратимости приближения к равновесию и существовании равновесных флуктуаций макроскопических величин. В настоящей главе мы используем методы Монте-Карло для моделирования равновесных свойств систем с многими степенями свободы. Этот круг задач позволит нам изучить методы статистической механики и ввести понятие температуры.

В какой-то степени благодаря численному моделированию статистическая механика добилась в последние годы многих впечатляющих результатов. Область приложений статистической механики простирается от традиционных задач плотных газов и жидкостей до исследования фазовых переходов, физики частиц и теорий ранней Вселенной. Фактически, используемый в этой главе алгоритм был разработан физиком, который заинтересовался возможностями численного моделирования для прогнозирования экспериментально проверяемых величин из «теорий калибровки на решетках», т.е. теорий, применяемых для описания фундаментальных взаимодействий элементарных частиц.

15.2. МИКРОКАНОНИЧЕСКИЙ АНСАМБЛЬ

Рассмотрим сначала *замкнутую* систему, у которой число частиц N , объем V и полная энергия E фиксированы. Предположим, кроме того, что система *изолированная*, т.е. влиянием внешних параметров, таких, как гравитационные и магнитные поля, можно пренебречь. Мы знаем, что в общем случае замкнутая макроскопическая система стремится перейти в стационарное равновесное состояние с максимальным беспорядком, или энтропией. *Макросостояние* системы характеризуется величинами N , V и E . На микроскопическом уровне существует в общем случае громадное число различных способов, иначе говоря *конфигураций*, в которых может реализовываться данное макросостояние (N, V, E) . Каждая конкретная конфигурация, или *микросостояние*, является *достижимой*, если его характеристики соответствуют данному микросостоянию.

Все, что нам известно о достижимых микросостояниях, — это что их свойства соответствуют известным физическим характеристикам системы. Раз у нас нет никаких причин предпочесть одно микросостояние другому, разумно постулировать, что в любой данный момент времени

система с *равной* вероятностью может оказаться в любом из своих достижимых микросостояний. Чтобы точнее выразить этот постулат *равных априорных вероятностей*, представим себе изолированную систему с Ω достижимыми состояниями. Вероятность найти систему в микросостоянии s равна

$$P_s = \begin{cases} 1/\Omega, & \text{если } s \text{ — достижимо,} \\ 0 & \text{в противном случае.} \end{cases} \quad (15.1)$$

Сумма P_s по всем Ω равна единице.

Средние от физических величин можно определять двумя способами. В обычном лабораторном эксперименте измерение физических величин производят в течение достаточно большого промежутка времени, чтобы в системе успело реализоваться большое число ее достижимых микросостояний. Такие усреднения по времени мы уже выполняли в гл. 6, где методом молекулярной динамики рассчитывались средние по времени значения таких величин, как температура и давление. С точки зрения такого усреднения по времени смысл вероятностей в формуле (15.1) заключается в том, что величина P_s дает долю времени, которую одна система за время последовательности наблюдений находится в данном микроскопическом состоянии s .

Несмотря на простой смысл средних по времени, удобно сформулировать статистические средние в данный момент времени. Вместо проведения измерений на одной системе представим себе совокупность, или *ансамбль*, систем, которая составлена из идентичных воображаемых копий, характеризующихся одним и тем же макросостоянием. Количество систем в ансамбле равно числу возможных микросостояний. В этой интерпретации вероятности в (15.1) описывают ансамбль тождественных систем. Ансамбль систем, характеризуемый величинами E , T , V и описываемый распределением вероятностей вида (15.1), называется *микроканоническим ансамблем*.

Предположим, что некоторая физическая величина A имеет значение A_s , когда система находится в состоянии s . Тогда среднее от A по ансамблю дается выражением

$$\langle A \rangle = \sum_s A_s P_s, \quad (15.2)$$

где P_s определено в (15.1).

Поскольку проведенные рассуждения о временных средних и средних по ансамблю остаются все же формальными, рассмотрим простой пример.

ТАБЛИЦА 15.1. Шестнадцать возможных микросостояний для одномерной системы из $N = 4$ невзаимодействующих частиц. Скорости частиц могут равняться v_0 или $-v_0$. Буква R отвечает частице, движущейся вправо, а буква L — влево. Масса частиц принята равной единице, и полная (кинетическая) энергия системы составляет $E = 4(v_0^2/2)$

L L L L	L L L R	L L R R	L R R R	R R R R
	L L R L	L R L R	R L R R	
	L R L L	L R R L	R R L R	
	R L L L	R L L R	R R R L	
		R L R L		
		R R L L		

Пусть у нас есть монета и мы хотим знать, с какой вероятностью P_h при ее бросании выпадет «орел». Мы можем найти P_h , бросая одну монету N раз и вычисляя отношение

$$P_h = \frac{n_h}{N}, \quad (15.3)$$

где n_h есть полное число случаев выпадения «орла». Можно ли ожидать, что всегда при бросании монеты N раз мы получим одинаковое значение P_h ? В то же время мы имеем право рассматривать ансамбль из 2 одинаковых монет, одна из которых находится в состоянии «орел» и другая — в состоянии «решка». Поскольку полное число состояний монеты Ω равно двум, из (15.1) имеем, что $P_h = 1/2$. Можно предположить, что оба метода вычисления P_h дают одинаковый результат.

С целью проиллюстрировать эти идеи на примере, имеющем более непосредственное отношение к физике, рассмотрим модель, в которой частицы различимы, не взаимодействуют и могут иметь только две скорости v_0 и $-v_0$. Поскольку частицы невзаимодействующие, размер системы и положения частиц не имеют значения. В табл. 15.1 изображен ансамбль систем, соответствующий $N = 4$ и $E = 2v_0^2$. Масса частиц считается равной единице.

Перебор всех шестнадцати систем ансамбля позволяет нам вычислить для физических величин системы средние по ансамблю. Например, из табл. 15.1 видно, что P_n — вероятность того, что число частиц, движущихся вправо, равно n , — составляет $1/16$, $4/16$, $6/16$, $4/16$, $1/16$ для $n = 0, 1, 2, 3$ и 4 соответственно. Отсюда среднее число частиц, движущихся вправо, равно

$$\langle n \rangle = \sum n P_n = (0 \cdot 1 + 1 \cdot 4 + 2 \cdot 6 + 3 \cdot 4 + 4 \cdot 1)/16 = 2.$$

15.3. МОДЕЛИРОВАНИЕ МЕТОДОМ МОНТЕ-КАРЛО

В гл. 6 мы выяснили, что систему многих частиц с заданными N , V и E можно промоделировать, интегрируя уравнения движения каждой частицы и вычисляя средние по времени значения рассматриваемых физических величин. Как можно выполнить усреднение по ансамблю при заданных N , V и E ? В качестве одного из способов можно было бы перебрать все микросостояния и вычислить средние по ансамблю от требуемых физических величин, как мы это делали в наших примерах. Однако такой подход, как правило, непрактичен, поскольку даже для небольшой системы число микросостояний очень уж велико. Следуя духу метода Монте-Карло, желательно разработать практический метод получения репрезентативной выборки из полного числа микросостояний. Очевидная процедура—это зафиксировать N и V , изменить случайным образом координаты и скорости отдельных частиц и принять конфигурацию, если она имеет требуемую полную энергию. Данная процедура, однако, весьма неэффективна, поскольку большинство конфигураций вообще не будут иметь требуемую полную энергию и должны быть отвергнуты.

Эффективную процедуру Монте-Карло предложил Мишель Кройц с сотрудниками. Представим себе макроскопическую систему, которая состоит из двух «подсистем», а именно исходной рассматриваемой системы, называемой далее *системой*, и *подсистемы*, состоящей из одного элемента. В силу исторических причин эта дополнительная степень свободы называется «демон». Демон путешествует по системе и переносит энергию, когда пытается изменить динамические переменные системы. Если демон хранит в своей мантии достаточно энергии, он отдает энергию любому элементу системы, которому требуется энергия, чтобы произвести требуемое изменение. Наоборот, если требуемое изменение уменьшает энергию системы, то избыточная энергия отдается демону. Единственное ограничение состоит в том, что энергия демона не может быть отрицательной. В итоге алгоритм принимает следующий вид:

- а. Выбираем случайным образом частицу и производим пробное изменение ее координат.
- б. Вычисляем изменение энергии системы, обусловленное изменением координат.
- в. Если пробное изменение уменьшает энергию системы, то система отдает энергию демону и новая конфигурация принимается.

- г. Если пробное изменение увеличивает энергию системы, то новая конфигурация принимается в том случае, если демон имеет достаточную энергию, чтобы передать ее системе. В противном случае новая конфигурация отбрасывается и частица сохраняет свои старые координаты.
- д. Если пробное изменение не меняет энергию системы, новая конфигурация принимается.

Указанная процедура повторяется до тех пор, пока не будет получена репрезентативная выборка состояний. Через определенный промежуток времени, необходимый для установления равновесия, демон и система достигнут компромисса и «договорятся» на какую-то среднюю для каждого энергию. Полная энергия остается постоянной, и, поскольку демон представляет только одну степень свободы по сравнению с многими степенями свободы системы, можно предположить, что флуктуации энергии системы будут малы.

Откуда следует, что результаты моделирования микроканонического ансамбля методами Монте-Карло и усреднения по времени в молекулярной динамике одинаковы? Предположение, что оба усреднения дают одинаковые результаты, называют *эргодической* гипотезой (или, точнее, квазиэргодической гипотезой). Хотя в общем случае нельзя показать, что оба средних идентичны, найдено, что во всех практически интересных случаях они дают одинаковые результаты.

15.4. ОДНОМЕРНЫЙ КЛАССИЧЕСКИЙ ИДЕАЛЬНЫЙ ГАЗ

Применим сначала алгоритм демона к классическому идеальному газу. В этом случае скорости частиц непрерывны и неограничены. Энергия любой конфигурации не зависит от положений частиц, и полная энергия есть сумма кинетических энергий отдельных частиц. Следовательно, для идеального газа из всех фазовых координат надо рассматривать только скорости. Чтобы изменить конфигурацию, мы выбираем случайным образом частицу и изменяем ее скорость на случайную величину. Для простоты рассматривается только одномерное движение частиц.

Разумеется, совершенно ни к чему пользоваться алгоритмом демона для идеального газа, поскольку в этом случае уменьшение энергии одной частицы можно легко компенсировать соответствующим увеличением энергии другой частицы. Тем не менее обычно неплохо рассмотреть сначала простой пример.

Программа `ideal_demon`, листинг которой приведен ниже, является примером микроканонического моделирования методом Монте-Карло одномерного идеального классического газа. Считается, что частицы имеют одинаковые начальные скорости. Переменные *vinitial* и *dvmax*, задаваемые в подпрограмме `initial`, определяют масштаб скорости и максимальное изменение скорости за одно испытание. Параметр *nmcs*—число шагов Монте-Карло на частицу—играет в моделировании методом Монте-Карло важную роль. Смысл этой величины состоит в том, что в среднем в одном расчете демон пытается изменить скорость каждой частицы *nmcs* раз.

```
PROGRAM ideal_demon
! алгоритм демона для одномерного идеального классического газа
DIM vel(100)
RANDOMIZE
CALL initial(N,nmcs,esystem,edemon,vel,vtot,dvmax)
FOR imcs = 1 to nmcs
  CALL changes(N,esystem,edemon,vel,vtot,escum,vcum,edcum,dvmax,accept)
NEXT imcs
CALL averages(N,nmcs,escum,vcum,edcum,accept)
END

SUB initial (N,nmcs,esystem,edemon,vel()),vtot,dvmax)
  INPUT prompt "число частиц = ": N
  INPUT prompt "число шагов Монте-Карло на частицу = ": nmcs
  INPUT prompt "начальная энергия системы = ": esystem
  LET edemon = 0 ! начальная энергия демона
  INPUT prompt "максимальное изменение скорости = ": dvmax
  LET vinitial = sqr(2*esystem/N) ! делим энергию поровну между частицами
  ! у всех частиц одинаковые начальные скорости
  FOR ipart = 1 to N
    LET vel(ipart) = vinitial
    LET vtot = vtot + vinitial ! полная скорость системы
  NEXT ipart
END SUB
```



```

SUB changes (N, esystem, edemon, vel(), vtot, escum, vcum, edcum, dvmax, accept)
  FOR i = 1 to N
    LET dv = (2*rnd - 1)*dvmax      ! пробное изменение скорости
    LET ipart = int(rnd*N + 1)      ! берем случайную частицу
    LET vtrial = vel(ipart) + dv    ! пробная скорость
    LET de = .5*(vtrial*vtrial - vel(ipart)*vel(ipart)) ! изменение энергии
    IF (de <= edemon) then
      LET vel(ipart) = vtrial
      LET vtot = vtot + dv          ! полная скорость системы
      LET accept = accept + 1
      LET edemon = edemon - de
      LET esystem = esystem + de
    END IF
    ! накапливаем данные после каждого испытания
    CALL data(esystem, edemon, vtot, escum, vcum, edcum)
  NEXT i
END SUB

SUB data (esystem, edemon, vtot, escum, vcum, edcum)
  LET edcum = edcum + edemon
  LET escum = escum + esystem
  LET vcum = vcum + vtot
END SUB

SUB averages (N, nmcs, escum, vcum, edcum, accept)
  LET norm = 1/(nmcs*N)
  LET edave = edcum*norm           ! средняя энергия демона
  LET accept = accept*norm         ! коэффициент принятия
  ! средние системы на частицу
  LET norm = norm/N
  LET esysave = escum*norm         ! средняя энергия на частицу системы
  LET vave = vcum*norm             ! средняя скорость на частицу системы
  PRINT "средняя энергия демона = "; edave
  PRINT "средняя энергия системы на частицу = "; esysave
  PRINT "средняя скорость на частицу = "; vave
  PRINT "коэффициент принятия = "; accept
END SUB

```

ЗАДАЧА 15.1. Моделирование идеального газа методом Монте-Карло

а. Выполните программу `ideal_demon`, взяв в качестве начального условия скорости всех частиц одинаковыми. Масса частиц считается равной единице. Возьмите для своих первых запусков следующие численные значения параметров: $N = 20$, начальная полная энергия $esystem = 10$, $dvmax = 2$ и $nmcs = 50$. Увеличивайте число шагов Монте-Карло на частицу $nmcs$ до тех пор, пока требуемые средние не станут постоянными с точностью в несколько процентов.

б. Чему равна начальная средняя скорость на частицу? Чему равно равновесное значение средней скорости на частицу?

в. Вычислите среднюю энергию демона и среднюю энергию системы на частицу. Зависят ли результаты, полученные в п.п. «а» и «б», от того, выбираются ли частицы случайно или последовательно?

г. Возьмите $esystem = 20$ и найдите значение $dvmax$, при котором коэффициент принятия получается равным около 50%. После того как установится равновесие, вычислите среднюю энергию демона и среднюю энергию системы на частицу. Затем рассмотрите $esystem = 40$ и получите приближенное соотношение между средней энергией демона и средней энергией системы на частицу.

д. Моделирование микроканонического ансамбля методом Монте-Карло проводится при фиксированной полной энергии независимо от температуры. Определим температуру соотношением $\frac{1}{2} m \langle v^2 \rangle = \frac{1}{2} k_B T_{kin}$, где $\frac{1}{2} m \langle v^2 \rangle$ есть средняя кинетическая энергия на частицу. Положим постоянную Больцмана k_B равной единице. Используя это соотношение, получите T_{kin} . Связана ли T_{kin} со средней энергией демона?

е. Какой смысл имеет «коэффициент принятия», определяемый в программе `ideal_demon`? Каково эмпирическое соотношение между коэффициентом принятия и $dvmax/v_{initial}$?

15.5. ТЕМПЕРАТУРА И КАНОНИЧЕСКИЙ АНСАМБЛЬ

Несмотря на простоту идеи микроканонического ансамбля, он не отражает ситуацию, наблюдаемую в лаборатории. Дело в том, что лабораторные системы не являются замкнутыми, а находятся в тепловом кон-

такте с окружающей средой. За счет этого теплового контакта осуществляется обмен энергией между лабораторной системой и окружающей средой в виде тепла. Обычно лабораторная система мала по сравнению с окружающей средой. Большая система, имеющая намного больше степеней свободы, называется *тепловым резервуаром* или *тепловой баней*.

Рассмотрим теперь более реалистичный случай, когда условие постоянства полной энергии будет относиться к *составной* системе, состоящей из лабораторной системы и теплового резервуара, а энергия лабораторной системы может меняться. Представим себе большое число воображаемых копий лабораторной системы и теплового резервуара. Рассматриваемые как единое целое, лабораторная система вместе с тепловым резервуаром являются изолированными и могут быть описаны микроканоническим ансамблем. Однако, поскольку нас интересуют равновесные значения физических величин, описывающих лабораторную систему, желательно знать вероятность P_s , с которой лабораторная система обнаруживается в состоянии s с энергией E_s . Ансамбль, который описывает распределение вероятностей состояний лабораторной системы, находящейся в тепловом равновесии с тепловым резервуаром, называется *каноническим*.

Вообще, в качестве лабораторной системы может выступать любая макроскопическая система, которая гораздо меньше теплового резервуара. Например, это могут быть затонувшие в океане сокровища, причем океан действует как тепловой резервуар. Лабораторной системой может быть столь малый объект, как отдельная частица, если ее можно надежно выявить среди частиц резервуара. Примером такой лабораторной системы служит демон. Следовательно, мы вправе считать, что демон является лабораторной системой, микроканоническое состояние которой определяется только ее энергией.

Наш метод отыскания формулы распределения вероятностей в каноническом ансамбле заключается в том, чтобы выполнить численное моделирование демона, который обменивается энергией с идеальным газом из N частиц. Сам идеальный газ будет играть роль тепловой бани, и мы будем определять вероятность $P(E_d)$ того, что демон имеет энергию E_d . В задаче 15.2 мы найдем, что формула $P(E_d)$ имеет вид:

$$P(E_d) = \frac{1}{Z} \exp(-E_d/k_B T), \quad (15.4)$$

где Z — нормировочный множитель, выбираемый из условия равенства единице суммы по всем состояниям демона. Параметр T в (15.4) назы-

вается *абсолютной температурой*. Если T измерять в градусах Кельвина, то постоянная Больцмана равна $k_B = 1.38 \cdot 10^{-16}$ эрг·град⁻¹. Распределение вероятностей (15.4) называется *распределением Больцмана* или *каноническим распределением*.

Отметим, что распределение Больцмана характеризуется температурой. Если предположить, что формула (15.4) применима для любой лабораторной системы, находящейся в тепловом равновесии с тепловой баней, то мы увидим, что в каноническом ансамбле всякое макросостояние определяется параметрами T , N и V . По сравнению с этим в микроканоническом ансамбле макросостояние характеризуется величинами E , N и V .

Формула (15.4) распределения Больцмана обеспечивает простой способ вычисления T по средней энергии $\langle E_d \rangle$ демона. Поскольку $\langle E_d \rangle$ равняется

$$\langle E_d \rangle = \int E \exp(-E/k_B T) dE / \int \exp(-E/k_B T) dE = k_B T, \quad (15.5)$$

то понятно, что T есть просто средняя энергия демона, поделенная на k_B . Заметим, что результат $\langle E_d \rangle = k_B T$ в (15.5) справедлив, только если энергия демона может принимать непрерывные значения.

ЗАДАЧА 15.2. Распределение Больцмана

а. Добавьте к программе `ideal_demon` подпрограмму, которая вычисляет распределение вероятностей $P(E_d)$ демона и его среднюю энергию. Нарисуйте график логарифма $P(E_d)$ как функцию от E_d и проверьте формулу (15.4) распределения Больцмана. Чему равен угловой коэффициент этого графика? Выберите такую систему единиц, в которой k_B равняется единице, и оцените соответствующую величину T . Параметры возьмите те же, что использовались в задаче 15.1.

б. Определите величину T из выражения (15.5). Совпадают ли ваши две оценки T ?

в. Сравните значение T , полученное в п.п. «а» или «б», со значением T , найденным в задаче 15.1 с помощью кинетического определения температуры. Находится ли демон в тепловом равновесии со своей тепловой баней?

15.6. МОДЕЛЬ ИЗИНГА

Простейшей и самой распространенной в статистической физике моделью системы фазовых взаимодействий является модель *Изинга*. С этой моделью связана богатая история, которую расписал Браш (см. список литературы). Модель была предложена Ленцем и исследована его дипломником Изингом с целью изучения фазового перехода из парамагнитного состояния в ферромагнитное. Изинг рассчитал термодинамические свойства модели в одномерной постановке и нашел, что в ней фазовый переход отсутствует. Однако в двумерном и трехмерном случаях модель Изинга действительно обнаруживает переход. Природа фазового перехода в двумерном случае и приложения модели Изинга к столь несхожим системам, как ферромагнетики и антиферромагнетики, бинарные сплавы, жидкости и немагнитные материалы, рассматриваются в гл. 16.

Чтобы познакомиться с моделью Изинга, рассмотрим решетку, содержащую N узлов, и предположим, что с каждым узлом решетки i связано число s_i , где $s_i = +1$, если спин ориентирован «вверх», и $s_i = -1$, если он ориентирован «вниз». Любая конкретная конфигурация, т.е. микросостояние решетки, задается набором переменных $\{s_1, s_2, \dots, s_N\}$ для всех узлов решетки.

Мы знаем, что макроскопические свойства системы определяются свойствами ее достижимых микросостояний. Следовательно, необходимо знать зависимость энергии E от конфигурации спинов. Полная энергия при наличии магнитного поля h в модели Изинга равняется

$$E = -J \sum_{\langle i, j \rangle} s_i s_j - h \sum_{i=1}^N s_i, \quad (15.6)$$

где первая сумма в (15.6) берется по всем ближайшим соседним парам спинов, а вторая сумма — по всем спинам решетки. Константа обменного взаимодействия J является мерой силы взаимодействия между ближайшими соседними спинами (рис. 15.1). Если $J > 0$, то состояния $\uparrow\uparrow$ и $\downarrow\downarrow$, которые характеризуются одинаковой ориентацией спинов ближайших со-



Рис. 15.1. Энергия взаимодействия между ближайшими соседними спинами в отсутствие внешнего магнитного поля.

седей, энергетически выгоднее по сравнению с состояниями $\uparrow\downarrow$ и $\downarrow\uparrow$, у которых соседние спины ориентированы в противоположные стороны. Следовательно, можно ожидать, что для $J > 0$ состояние с наименьшей полной энергией является *ферромагнитным*, т.е. в среднем суммарное число спинов, ориентированных в одном направлении, не равно нулю. Если $J < 0$, предпочтительнее состояния $\uparrow\downarrow$ и $\downarrow\uparrow$, для которых соседние спины антипараллельны, и можно ожидать, что состояние с наименьшей энергией является *антиферромагнитным*, т.е. спины упорядочены через один. Если наложить внешнее «магнитное поле» h , направленное вверх, то спины \uparrow и \downarrow приобретают дополнительную внутреннюю энергию, равную $-h$ и $+h$ соответственно. Отметим, что h измеряется в таких единицах, что магнитный момент на спин равен 1.

Важным достоинством модели Изинга является ее простота. В ряду упрощающих предположений, положенных в основу модели, отметим такие: в ней пренебрегается кинетической энергией атомов, связанных с узлами решетки; в энергии взаимодействия учитывается вклад только ближайших соседей и предусматривается только два дискретных состояния для спинов. Несмотря на простоту модели, мы увидим, что она проявляет интересные свойства.

Для хорошо знакомого случая классических частиц, координаты и скорости которых могут принимать континуум значений, динамика определяется законами Ньютона. В модели Изинга зависимости (15.6) энергии от спиновой конфигурации недостаточно, чтобы определить временные свойства системы. Иными словами, соотношение (15.6) не говорит нам, как меняется система при изменении спиновой конфигурации, и нам приходится вводить динамику отдельно. Наибольшее распространение для спиновых систем Изинга получила динамика «опрокидывания спина». В этой динамике спин выбирается случайным образом и пробное изменение (испытание Монте-Карло) соответствует опрокидыванию спина из состояния \uparrow в \downarrow или из \downarrow в \uparrow . Другая динамика для модели Изинга обсуждается в гл. 16.

Применим теперь алгоритм демона к моделированию модели Изинга в микроканоническом ансамбле. В одномерной модели Изинга демон должен выбирать спины случайно, чтобы не попадать на периодически повторяющиеся конфигурации. Поскольку нас интересуют свойства бесконечной системы, нужно учесть краевые условия. В качестве простейшего краевого условия выбирается «свободная граница», означающая, что спины в узлах 1 и N взаимодействуют только с одним ближайшим соседом. Вообще говоря, лучше выбирать периодические (тороидальные) краевые

условия. В этом варианте решетка превращается в кольцо и спины в узлах 1 и N взаимодействуют друг с другом., а значит, имеют такое же число взаимодействий, что и остальные спины.

Чему равны некоторые средние физические величины, которые желательно вычислить? Очевидной физической величиной является суммарный магнитный момент, или *намагниченность* M , определяемый формулой

$$M = \sum_{i=1}^N s_i . \quad (15.7)$$

(Напомним, что мы положили магнитный момент спина равным единице.) Обычно интерес представляют средние значения величины $\langle M \rangle$ и флуктуации $\langle M^2 \rangle - \langle M \rangle^2$ как функции температуры системы и наложенного магнитного поля. Зависимость температуры от энергии можно определять двумя способами. Один состоит в измерении вероятности того, что демон имеет энергию E_d . Поскольку мы знаем, что эта вероятность пропорциональна $\exp(-E_d/k_B T)$, то можно определить температуру из графика логарифма вероятности как функции от E_d . Более легкий способ определения температуры — измерять среднюю энергию демона. Однако поскольку в модели Изинга значения энергии демона не непрерывны, то температура не пропорциональна средней энергии демона, как это имеет место для идеального газа. В приложении 15А мы показываем, что в пределе бесконечной системы температура для $h = 0$ связана с E_d соотношением

$$k_B T/J = \frac{4}{\ln(1 + 4J/E_d)} . \quad (15.8)$$

Формула (15.8) получается в результате замены интегралов в выражении (15.5) суммами по всем возможным значениям энергии демона. Заметим, что в пределе $|J/E_d| \ll 1$ формула (15.8) переходит в равенство $k_B T = E_d$, что и следовало ожидать.

В программе `Ising_demon` реализовано микроканоническое моделирование одномерной модели Изинга с использованием периодических краевых условий и динамикой опрокидывания спина. Когда начальная конфигурация выбрана, алгоритм демона аналогичен алгоритму, описанному в разд. 15.3. Однако по сравнению с идеальным газом в одномерной модели Изинга спины нужно выбирать случайным образом.

```
PROGRAM Ising_demon
! алгоритм демона для одномерной модели Изинга
DIM s(1000)
RANDOMIZE
CALL initial(N,nmcs,esystem,edemon,s,J,h,mag)
FOR imcs = 1 to nmcs
  FOR i = 1 to N
    CALL changes(N,esystem,edemon,s,J,h,mag,accept)
    ! накопление данных после каждого пробного опрокидывания
    CALL data(esystem,edemon,mag,escum,magcum,mag2cum,edcum)
  NEXT i
NEXT imcs
CALL averages(N,nmcs,J,escum,magcum,mag2cum,edcum,accept)
END

SUB initial (N,nmcs,esystem,edemon,s(),J,h,mag)
  INPUT prompt "число спинов = ": N
  INPUT prompt "число шагов Монте-Карло на спин = ": nmcs
  LET h = 0 ! внешнее магнитное поле
  INPUT prompt "константа обменного взаимодействия = ": J
  INPUT prompt "искомая полная энергия = ": esi
  ! начальная конфигурация спинов в состоянии с минимальной энергией
  FOR ispin = 1 to N
    LET s(ispin) = 1
  NEXT ispin
  LET mag = N ! суммарная намагниченность
  ! вычисляем начальную энергию системы
  LET esystem = -(J + h)*N
  LET edemon = 4*J*int((esi - esystem)/(4*J))
  PRINT "полная энергия = "; esystem + edemon
END SUB
```



```

SUB changes (N, esystem, edemon, s(), J, h, mag, accept)
  LET ispin = int(rnd*N + 1)           ! случайный спин
  ! значения соседних спинов находим из периодических краевых условий
  IF ispin = 1 then
    LET left = s(N)
  ELSE
    LET left = s(ispin - 1)
  END IF
  IF ispin = N then
    LET right = s(1)
  ELSE
    LET right = s(ispin + 1)
  END IF
  ! spin flip dynamics
  LET de = 2*s(ispin)*(-h + J*(left + right)) ! пробное изменение энергии
  IF de <= edemon then
    LET s(ispin) = -s(ispin)
    LET mag = mag + 2*s(ispin)
    LET accept = accept + 1           ! число принятых изменений
    LET edemon = edemon - de
    LET esystem = esystem + de
  END IF
END SUB

SUB data (esystem, edemon, mag, escum, magcum, mag2cum, edcum)
  ! накопление данных
  LET edcum = edcum + edemon
  LET escum = escum + esystem
  LET magcum = magcum + mag
  LET mag2cum = mag2cum + mag*mag
END SUB

```

```

SUB averages (N,nmcs,J,escum,magcum,mag2cum,edcum,accept)
  LET norm = 1/(N*nmcs)      ! данные собраны по каждому испытанию
  LET edave = edcum*norm
  LET accept = accept*norm    ! коэффициент принятия
  ! средние на спин
  LET norm = norm/N
  LET es_ave = escum*norm
  LET magave = magcum*norm
  LET mag2ave = mag2cum*norm
  PRINT "средняя энергия демона = ";edave
  PRINT "средняя энергия системы на спин = "; es_ave
  PRINT "средняя намагниченность на спин = "; magave
  PRINT "средний квадрат намагниченности на спин = "; mag2ave
  PRINT "коэффициент принятия = ";accept
  LET temperature = 4*J/log(1 + 4*J/edave)
  PRINT "температура = "; temperature
END SUB

```

Отметим, что для $h = 0$ обусловленное опрокидыванием спина изменение энергии равно либо 0, либо $\pm 4J$. Отсюда начальная энергия системы и демона должна быть кратна $4J$. Поскольку спины взаимодействуют, то трудно выбрать начальную конфигурацию спинов, имеющую точно требуемую энергию. В процедуре, заложенной в подпрограмму *initial*, все спины в начальной конфигурации выбираются ориентированными «вверх», т.е. формируется конфигурация с минимальной энергией. Преимущество данной конфигурации в том, что можно легко вычислить полную энергию. После этого энергия демона выбирается так, чтобы полная энергия системы и демона равнялась искомому кратному $4J$.

ЗАДАЧА 15.3. Одномерная модель Изинга

а. Используйте программу *Ising_demon* с $N = 100$, $J = 1$, $h = 0$ и требуемой полной энергией $esi = -20$. Какая начальная энергия придается демону в подпрограмме *initial*? По мере распределения энергии демона по N спином все физические величины медленно меняются. Вычислите скользящее среднее от энергии демона и M как функции числа шагов Монте-Карло на спин («время»). Обратите внимание, что данные берутся после каждого испытания, а не после каждого шага Монте-Карло на спин. Чему приближенно равно время,

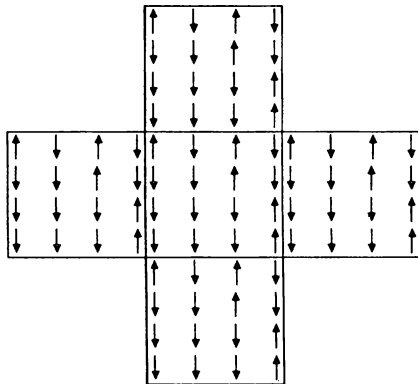


Рис. 15.2. Одна из 2^N возможных конфигураций системы $N = 16$ спинов в модели Изинга на квадратной решетке. Показаны также четыре периодические копии центральной ячейки. Состояние спина «вверх» обозначено значком \uparrow , а состояние «вниз» — значком \downarrow . Обратите внимание на то, что в квадратной решетке число ближайших соседей равно четырем. При периодических краевых условиях энергия изображенной конфигурации равна $E = -8J + 4h$.

необходимое для выхода этих величин на равновесные значения? Модифицируйте эту программу таким образом, чтобы в вычислении средних от физических величин не учитывались неравновесные конфигурации. Чему равны средние равновесные значения $\langle E_d \rangle$, $\langle M \rangle$ и $\langle M^2 \rangle$? Для тестирования программы вполне можно взять $nmcs = 100$, при этом точность результатов составит приблизительно 20%. Чтобы получить точность не хуже 5%, $nmcs$ должно быть порядка 1000.

б. Для параметров системы, рассмотренных в п. «а», определите, используя соотношение (15.8), равновесную температуру T . Энергию E измеряйте в единицах J . Чему равна соответствующая энергия системы?

в. Вычислите E и T для трех случаев $N = 100$, $J = 1$ с $esi = -40$, -60 и -80 . Сравните полученные результаты с точным ответом для бесконечной одномерной решетки, который равен $E/N = -\ln(J/k_B T)$. Как вычисленные вами результаты для E/N зависят от числа спинов N и количества шагов Монте-Карло на спин?

г. Используя результаты тех же вариантов, вычислите $\langle M^2 \rangle$ как

функцию от T . Как ведет себя $\langle M^2 \rangle$ с ростом T ? Возрастает или убывает?

д. Модифицируйте программу **Ising_demon** и проверьте, подчиняется ли энергия демона распределению Больцмана (15.4).

*е. Введите ненулевое магнитное поле h и вычислите $\langle E_d \rangle$, $\langle M \rangle$ и $\langle M^2 \rangle$ как функции от h для заданной полной энергии E . Являются ли возможные значения E произвольными? Прочитайте, что говорится по этому поводу в приложении 15А, и определите связь $\langle E_d \rangle$ с температурой для $h \neq 0$. Больше или меньше равновесная температура при той же полной энергии?

*ЗАДАЧА 15.4. Двумерная модель Изинга

а. Обобщите программу **Ising_demon** на случай двумерной модели Изинга на квадратной решетке с динамикой опрокидывания спинов. Возьмите решетку со стороной, равной L . Полное число спинов N равняется $N = L^2$. Используйте периодические краевые условия, как показано на рис. 15.2, так чтобы спины в левом столбце взаимодействовали со спинами в правом столбце и т. д.

б. В отличие от одномерного случая в многомерных решетках спины можно выбирать как последовательно, так и случайно. Вычислите среднюю энергию демона и $\langle M^2 \rangle$ как функции от E . Начальные параметры удобно выбрать равными $L = 5$ и $h = 0$. С помощью формулы (15.8) определите зависимость температуры от энергии системы.

в. Модифицируйте свою программу, чтобы сделать несколько «снимков» конфигураций спинов. Опишите качественно вид этих конфигураций при различных энергиях и температурах. Упорядочены они или нет? Имеются ли домены спинов, ориентированных только вверх или вниз?

*15.7. ПОТОК ТЕПЛА

Во всех рассмотренных задачах энергия одного демона распределялась поровну между всеми спинами. В результате все спины приобретали одинаковую среднюю энергию взаимодействия. Много интересных вопро-

сов возникает в случае, когда система не является пространственно однородной и пребывает в неравновесном, но не зависящем от времени (стационарном) состоянии.

Рассмотрим задачу о потоке тепла в однородной модели Изинга. Предположим, что вместо распределения энергии одного демона по всем узлам каждый узел имеет своего демона. Поток тепла через такую систему можно изучать, требуя, чтобы демоны в узлах 1 и N удовлетворяли условиям, отличным от демонов в остальных узлах. Демон в узле 1 добавляет энергию в систему, опрокидывая спин 1 так, чтобы он всегда был в состоянии с наибольшей энергией, т.е. ориентирован противоположно спину 2. Демон в узле N забирает энергию у системы, опрокидывая спин N так, чтобы спин N был в состоянии с наименьшей энергией, т.е. параллелен спину $N - 1$. В результате этих допущений энергия передается от узла 1 к узлу N посредством демонов, связанных с другими узлами. Чтобы энергия не скапливалась на «горячем» конце цепочки Изинга, мы требуем, чтобы спин 1 мог добавить энергию в систему, только если одновременно с этим спин N отбирает энергию у системы. Поскольку демоны на обоих концах решетки удовлетворяют условиям, отличным от условий для остальных демонов, необходимо убрать периодические краевые условия.

Температура системы определяется, по сути дела, обобщением соотношения (15.8), т.е. температура в узле i связана со средней энергией демона в узле i . Градиент температуры можно регулировать, изменяя концевые спины со скоростью, отличной от других спинов. Максимальный градиент температуры получается, если мы изменяем концевые спины после каждого изменения внутреннего спина. Меньший градиент температуры получается, если мы изменяем концевые спины реже. Градиент температуры между любыми двумя спинами можно определить из «температурного профиля», т.е. пространственной зависимости температуры. Поток энергии можно определить, вычисляя величину энергии, входящей в решетку через узел 1 в единицу времени.

Для реализации указанной процедуры мы модифицируем программу **Ising_demon**, превратив переменные *edemon* и *edcut* в массивы. Обычную процедуру пересчета спинов мы осуществляем для спинов со 2-го по $(N - 1)$ -й, а к спинам 1 и N обращаемся с постоянной периодичностью, определяемой значением переменной *nheat*. Поскольку о каждом спине почти никакой новой информации не образуется до тех пор, пока все спины не испытают одно опрокидывание, мы снимаем данные в программе **conduct** только на каждом шаге Монте-Карло на спин, а не после каждого пробного опрокидывания.

```
PROGRAM conduct  ! алгоритм многих демонов для 1-d модели Изинга
! тепло подводится у спина 1 и отводится у спина N
DIM s(1000), edemon(1000), edcum(1000), magcum(1000)
RANDOMIZE
CALL initial(N, nmcs, s, J, nheat)
FOR imcs = 1 to nmcs
  FOR ispin = 1 to N - 2
    LET istep = (imcs - 1)*(N - 2) + ispin ! число пробных опрокидываний
    IF (mod(istep, nheat) = 0) then CALL heat(N, J, s, edcum)
    CALL change(N, edemon, s, J, accept)
  NEXT ispin
  CALL data(N, s, edemon, magcum, edcum)
NEXT imcs
CALL average(N, nmcs, J, magcum, edcum, accept)
END

SUB initial (N, nmcs, s(), J, nheat)
  INPUT prompt "число спинов = ": N
  INPUT prompt "число шагов Монте-Карло на спин = ": nmcs
  INPUT prompt "константа обменного взаимодействия = ": J
  INPUT prompt "период изменения концевых спинов (шаги) = ": nheat
  FOR i = 1 to N
    ! начальная случайная конфигурация
    IF rnd > 0.5 then
      LET s(i) = 1
    ELSE
      LET s(i) = -1
    END IF
  NEXT i
END SUB

SUB change (N, edemon(), s(), J, accept)
! динамика опрокидывания спина
! выбираем случайный спин с номером от 2 до N - 1
LET ispin = int(rnd*(N - 2) + 2)
LET de = 2*s(ispin)*J*(s(ispin - 1) + s(ispin + 1)) ! пробное изменение энергии
IF de <= edemon(ispin) then
  LET s(ispin) = - s(ispin)
  LET accept = accept + 1
  LET edemon(ispin) = edemon(ispin) - de
END IF
END SUB
```

```

SUB data (N,s(),edemon(),magcum(),edcum())
  FOR i = 2 to N - 1
    LET edcum(i) = edcum(i) + edemon(i)
    LET magcum(i) = magcum(i) + s(i)
  NEXT i
END SUB

SUB average (N,nmcs,J,magcum(),edcum(),accept)
  INPUT prompt "имя выходного файла = ": out$
  OPEN #1: name out$, access output, create new
  LET norm = 1/nmcs
  LET accept = accept*norm/(N - 2)
  PRINT "коэффициент принятия = "; accept
  PRINT #1: "коэффициент принятия = "; accept
  PRINT "edave(1) = приток тепла, edave(N) = выход энергии"
  PRINT tab(2);"i";tab(16);"edave";tab(35);"T";tab(45);"намагниченность"
  PRINT
  PRINT #1: "edave(1) = приток тепла, edave(N) = выход энергии"
  PRINT #1: tab(2);"i";tab(16);"edave";tab(35);"T";tab(45);"намагниченность"
  PRINT #1
  FOR i = 1 to N
    LET edave = edcum(i)*norm
    LET temperature = 0
    IF edave <> 0 then
      IF (1 + 4*J/edave) > 0 then
        LET temperature = 4*J/log(1 + 4*J/edave)
      END IF
    END IF
    LET mag = magcum(i)*norm
    PRINT i,edave,temperature,mag
    PRINT #1: i,edave,temperature,mag
  NEXT i
  CLOSE #1
END SUB

```

```

SUB heat(N,J,s(),edcum())
  ! попытка добавить тепло у спина 1 и отвести у спина N принимается,
  ! только если спины 1 и 2 параллельны, а N и N-1 антипараллельны
  IF (s(1)*s(2) = 1) and (s(N)*s(N-1) = -1) then
    LET edcum(1) = edcum(1) + 2*J
    LET edcum(N) = edcum(N) - 2*J
    LET s(1) = -s(1)
    LET s(N) = -s(N)
  END IF
END SUB

```

*ЗАДАЧА 15.5. Одномерный поток тепла

а. Модифицируйте программу **conduct** так, чтобы все демоны были эквивалентны, т.е. наложите периодические краевые условия и не используйте подпрограмму **heat**. Вычислите среднюю энергию демона в каждом узле и, используя формулу (15.8), определите локальную температуру узлов. В качестве параметров возьмите $N = 22$, $nmcs = 1000$ и $J = 1$. Однородна ли локальная температура? Как соотносятся ваши результаты со случаем одного демона?

б. В программе **conduct** энергия вводится в систему в узле 1 и отводится в узле N . Возьмите параметры $N = 22$, $J = 1$, $nmcs = 1000$ и $nheat = 100$. Для каждого узла определите среднюю энергию демона и получите соответствующую локальную температуру. Начертите профиль температуры, изображая температуру как функцию номера узла. Градиент температуры есть разность температур в узлах от $(N-1)$ -го до 2-го, деленная на расстояние между ними. (Напомним, что расстояние между соседними узлами принято равным единице.) Вследствие флуктуаций локальной температуры и концевых эффектов градиент температуры следует оценивать, аппроксимируя профиль температуры в середине решетки прямой линией. Определите также среднюю температуру системы.

в. Плотность теплового потока Q есть поток энергии на единицу длины в единицу времени. Поток энергии—это величина энергии, которую демон 1 добавляет в систему в узле 1. Удобно измерять «время» в единицах шагов Монте-Карло на спин. Определите Q для параметров, использованных в п. «б».

г. Если градиент температуры $\partial T/\partial x$ не слишком велик, то плотность теплового потока Q пропорциональна $\partial T/\partial x$. Можно определить удельную теплопроводность κ соотношением

$$Q = -\kappa \frac{\partial T}{\partial x}. \quad (15.9)$$

Используя полученные результаты для $\partial T/\partial x$ и Q , оцените κ . Из-за ограниченного числа спинов и шагов Монте-Карло ваши результаты получатся с точностью только около 20%. Для получения более точных результатов потребовалась бы система из по крайней мере 50 спинов и 10 000–100 000 шагов Монте-Карло на спин. Однако если не использовать специальные приемы программирования (см. разд. 15.8), то для такого варианта потребовался бы уже большой компьютер.

д. Максимальные плотность теплового потока и градиент температуры могут достигаться при $nheat = 1$. Определите Q , профиль температуры и среднюю температуру при различных значениях $nheat$. Для всех ли $nheat$ профиль температуры линеен? Если профиль температуры линеен, оцените $\partial T/\partial x$ и определите κ . Зависит ли κ от значения средней температуры?

Отметим, что в задаче 15.5 нам удалось рассчитать профиль температуры, прибегнув к алгоритму, который оперирует только с целыми числами. При стандартном подходе надо было бы решать уравнение теплопроводности, сходное по виду с уравнением диффузии (11.34).

*ЗАДАЧА 15.6. Профиль намагниченности

а. Модифицируйте программу `conduct`, удалив подпрограмму `heat` и ограничив спины 1 и N значениями +1 и -1 соответственно. Оцените профиль намагниченности, построив графики среднего значения спина в каждом узле как функции номера узла. Возьмите параметры $N = 22$, $nmcsc = 1000$ и $J = 1$. Как меняются ваши результаты с ростом N ?

б. Вычислите среднюю энергию демона и по ней локальную температуру в каждом узле. Однородна ли температура системы, даже если намагниченность неоднородна? Находится ли система в тепловом равновесии?

в. Влияние этого ограничения легче наблюдать в двумерном и трехмерном случаях, чем в одномерном. Составьте программу для двумерной модели Изинга на квадратной решетке $L \times L$. Ограничьте спины в узле (i, j) значениями $+1$ и -1 для $i = 1$ и $i = L$ соответственно. Используйте периодические краевые условия по y . Как соотносятся ваши результаты с одномерным случаем?

г. Откажитесь от периодических краевых условий по y и зафиксируйте все граничные спины от $i = 1$ до $L/2$ равными $+1$, а все остальные граничные спины равными -1 . Возьмите начальную конфигурацию, где все спины в левой половине системы равны $+1$, а все остальные -1 . Проведите моделирование и изобразите конфигурацию спинов после достижения системой равновесия. Соедините на схеме каждую пару спинов противоположного знака. Опишите кривую, разделяющую спины $+1$ и спины -1 . Опишите вид этой кривой. Начните с $L = 20$ и установите, что будет происходить при увеличении L .

15.8. ЗАМЕЧАНИЯ

Достоинством микроканонического ансамбля является обезоруживающая простота постулата равных априорных вероятностей. Главное достоинство в проведении численного моделирования с использованием микроканонического ансамбля, а не других ансамблей состоит в том, что демон не предъявляет никаких требований к генератору случайных чисел. Кройцу с сотрудниками удалось разработать очень быстрые алгоритмы, используя один машинный бит на спин и много демонов. Разумеется, алгоритм демона обладает также и некоторыми недостатками. Один из них связан с трудностями задания системы с требуемой величиной энергии. Однако для нас самый важный недостаток заключен в самом замысле. Дело в том, что в наших задачах более естественно рассматривать поведение макроскопических физических величин в зависимости от температуры, а не полной энергии. Поэтому дальнейшие свойства модели Изинга мы рассматриваем в гл. 16 в контексте канонического ансамбля.

ЛИТЕРАТУРА

S. G. Brush, History of the Lenz-Ising Model, Rev. Mod. Phys. **39**, 883 (1967).

M. Creutz, Deterministic Ising dynamics, Ann. Phys. **167**, 62 (1986). Автор приводит правило детерминированных клеточных автоматов, которое моделирует модель Изинга и позволяет исследовать неравновесные процессы.

M. Creutz, Microcanonical Monte Carlo Simulation, Phys. Rev. Letts. **50**, 1411 (1983). См. также G. Bhanot, M. Creutz, H. Neuberger, Microcanonical simulation of Ising systems, Nucl. Phys. **235**, 417 (1984).

ДОПОЛНИТЕЛЬНАЯ ЛИТЕРАТУРА

Хуанг К., Статистическая механика. — М.: Мир, 1966. В монографии известного теоретика подробно рассматриваются следующие вопросы: классический идеальный газ, микроканонический ансамбль и одномерная модель Изинга.

Дайсон Ф., Монтролл Э., Кац М., Фишер М., Устойчивость и фазовые переходы. — М.: Мир, 1973. Лекция Э. Монтролла посвящена модели Изинга.

Методы Монте-Карло в статистической физике/Под ред. К. Биндера. — М.: Мир, 1982. Обсуждается метод Монте-Карло и его приложения к различным задачам термодинамики и статистической физики.

ПРИЛОЖЕНИЕ 15А. СВЯЗЬ СРЕДНЕЙ ЭНЕРГИИ ДЕМОНА С ТЕМПЕРАТУРОЙ

Мы знаем, что энергия демона E_d ограничена положительными значениями и определяется формулой $E_d = E - E_s$, где E_s есть энергия системы, а E — полная энергия. Как мы установили в задачах 15.2 и 15.3, величина энергии демона равна E_d с вероятностью, пропорциональной $\exp(-E_d/k_B T)$. Предположим, что такое же распределение вероятностей справедливо и для любой макроскопической системы, находящейся в термодинамическом равновесии. Тогда $\langle E_d \rangle$ равно

$$\langle E_d \rangle = \frac{\sum E_d \exp(-E_d/k_B T)}{\sum \exp(-E_d/k_B T)}, \quad (15.10)$$

где суммы вычисляются по всем возможным значениям E_d . Если в модели Изинга спин опрокидывается в нулевом магнитном поле, то минимальная ненулевая потеря энергии системы составляет $4J$ (рис. 15.3). Следовательно энергия демона может равняться $0, 4J, 8J, 12J, \dots$. Если мы запишем $x = 4J/k_B T$, то в выражении (15.10) можно выполнить суммирование, и получим

$$\langle E_d/k_B T \rangle = \frac{0 + xe^{-x} + 2xe^{-2x} + \dots}{1 + e^{-x} + e^{-2x} + \dots} = \frac{x}{e^x - 1}. \quad (15.11)$$

Формулу (15.8) можно получить, выразив из (15.11) T через E_d . Убедитесь самостоятельно, что для решеток с четным числом ближайших соседей соотношение (15.11) не зависит от размерности.

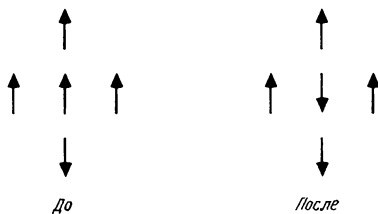


Рис. 15.3. Изменение энергии, обусловленное изменением ориентации центрального спина, равно $4J$. В общем случае изменение энергии за счет опрокидывания спина составляет $+2sJ + 2h$, где s — суммарный спин четырех ближайших соседних спинов.

Если магнитное поле отлично от нуля, то возможными энергиями демона являются $0, 2h, 4J - 2h, 4J + 2h, \dots$. Если J кратно h , то результат получается такой же, как раньше, с заменой $4J$ на $2h$, поскольку возможные значения энергии демона кратны $2h$. В противном случае нам необходимо по отдельности вычислять суммы в числителе и знаменателе (15.10) для всех возможных энергий демона. Проще всего вычислить прямо эти суммы на компьютере и составить таблицу значений E_d в зависимости от T для нескольких значений h . Как правило, бывает достаточно учитывать в каждой сумме порядка 10–20 членов.

МОДЕЛИРОВАНИЕ КАНОНИЧЕСКОГО АНСАМБЛЯ МЕТОДОМ МОНТЕ-КАРЛО

16

Излагаются методы Монте-Карло для моделирования систем, находящихся в равновесии с тепловой баней. Рассматриваются приложения к моделям магнетизма и жидкостей.

16.1. КАНОНИЧЕСКИЙ АНСАМБЛЬ

Большинство физических систем не являются изолированными, а обмениваются энергией с окружающей средой. Поскольку обычно такая система мала по сравнению со своим окружением, будем считать, что любое изменение энергии малой системы не влияет заметным образом на температуру большой системы. Тем самым большая система действует как *тепловой резервуар* или *тепловая баня* с заданной абсолютной температурой T . Если малую, но макроскопическую, систему привести в тепловой контакт с тепловой баней, то система будет стремиться перейти в равновесное состояние путем обмена энергией с тепловой баней, и этот процесс будет продолжаться до тех пор, пока система не достигнет температуры бани.

Представим себе бесконечно большое число воображаемых копий системы и тепловой бани. В гл. 15 мы убедились в том, что P_s , вероятность того, что *система* находится в микросостоянии s с энергией E_s , описывается формулой

$$P_s = \frac{1}{Z} \exp(-E_s/k_B T), \quad (16.1)$$

где Z — нормировка. Ансамбль, определяемый выражением (16.1), называется *каноническим*. Поскольку $\sum P_s = 1$, то Z равно

$$Z = \sum_{s=1}^M \exp(-E_s/k_B T). \quad (16.2)$$

Сумма в (16.2) берется по всем M микросостояниям системы. Величина Z называется *суммой по состояниям* системы.

С помощью формулы (16.1) можно получить среднее по ансамблю от рассматриваемых физических величин. Например, средняя энергия равна

$$\langle E \rangle = \sum_s E_s P_s = \frac{1}{Z} \sum_s E_s \exp(-\beta E_s). \quad (16.3)$$

Заметим, что в каноническом ансамбле возможны флуктуации энергии.

16.2. АЛГОРИТМ МЕТРОПОЛИСА

Как можно промоделировать систему N частиц, находящихся в объеме V при постоянной температуре T ? Поскольку мы умеем генерировать только ограниченное число m из полного числа M конфигураций, то можно надеяться получить оценку среднего значения $\langle A \rangle$ из выражений

$$\langle A \rangle = \frac{\sum_s^M A_s \exp(-\beta E_s)}{\sum_s^M \exp(-\beta E_s)} \approx \quad (16.4)$$

$$\approx \frac{\sum_s^m A_s \exp(-\beta E_s)}{\sum_s^m \exp(-\beta E_s)}, \quad (16.5)$$

где E_s и A_s обозначают полную энергию и значение физической величины A в конфигурации s . Напрашивается простейшая процедура Монте-Карло, состоящая в том, что генерируется случайная конфигурация, вычисляются E_s , A_s и произведение $A_s \exp(-\beta E_s)$ и подсчитывается соответствующий вклад этой конфигурации в суммы (16.5). Однако каждая такая конфигурация была бы, по-видимому, очень маловероятна и поэтому давала бы малый вклад в сумму. Вместо этого, как мы увидели в целом ряде задач, нужно пользоваться методом *существенной выборки* и генерировать конфигурации в соответствии с функцией распределения вероятностей π_s . Поскольку усреднение будет проводиться по m конфигурациям смещенной выборки, мы для исключения этого смещения должны взвешивать каждую конфигурацию с множителем $1/\pi_s$:

$$\langle A \rangle \approx \sum_s^m A_s \frac{1}{\pi_s} \exp(-\beta E_s) / \sum_s^m \frac{1}{\pi_s} \exp(-\beta E_s). \quad (16.6)$$

Внимательный анализ формулы (16.6) убедит вас в том, что в качестве π_s целесообразно выбрать само распределение Больцмана, т.е.

$$\pi_s = \frac{\exp(-\beta E_s)}{\sum_s \exp(-\beta E_s)}. \quad (16.7)$$

В результате такого выбора π_s величину $\langle A \rangle$ можно записать в виде

$$\langle A \rangle \approx \frac{1}{m} \sum_s A_s. \quad (16.8)$$

Выбор π_s в виде (16.7) предложен Метрополисом и др. (см. список литературы).

В задачах статистической механики выражения «метод Монте-Карло» и «метод выборки Метрополиса» — почти синонимы. Хотя в гл. 10 мы говорили о методе выборки Метрополиса, рассматривая численное интегрирование, его можно легко ввести и применительно к настоящим (и исходным) задачам. Ниже мы дадим наиболее общую форму алгоритма Метрополиса на примере системы спинов или частиц.

0. Формируем начальную конфигурацию.
1. Производим случайное пробное изменение в начальной конфигурации. Например, выбираем случайным образом какой-нибудь спин и пробуем его опрокинуть. Или выбираем случайную частицу и пробуем переместить ее на случайное расстояние.
2. Вычисляем ΔE , т.е. изменение энергии системы, обусловленное произведенным пробным изменением конфигурации.
3. Если ΔE меньше или равно нулю, то принимаем новую конфигурацию и переходим к шагу 7.
4. Если ΔE положительно, вычисляем «вероятность перехода» $W = \exp(-\Delta E/k_B T)$.
5. Генерируем случайное число r в интервале $(0, 1)$.
6. Если $r \leq W$, то новую конфигурацию принимаем, в противном случае сохраняем предыдущую конфигурацию.
7. Определяем значения требуемых физических величин.
8. Повторяем шаги 1—7 для получения достаточного числа конфигураций или «испытаний».
9. Вычисляем средние по конфигурациям, которые статистически независимы друг от друга.

Описанные выше шаги можно интерпретировать как случайное блуждание. Будем считать различные конфигурации «точками», снабженными порядковыми номерами $i = 1, 2, 3, \dots$, и рассмотрим случайное блуждание по этим точкам. Шаги 2—6 дают условную вероятность того, что в «момент времени» $t+1$ прохожий будет находиться в точке i при условии, что в момент времени t он был в точке j . Поскольку нужно вычислять только отношение $P(i)/P(j)$, нормировать $P(i)$ на единицу нет необходимости. Заметим, что конфигурации генерируются с вероятностью, пропорциональной требуемой вероятности, и поэтому все средние превращаются в арифметические средние, как в формуле (16.8).

Поскольку, однако, коэффициент пропорциональности неизвестен, то оценить этим способом сумму по состояниям Z невозможно.

Доказательство того, что после достаточного числа шагов алгоритм Метрополиса генерирует состояния с вероятностью, пропорциональной распределению Больцмана, мало дает для физического понимания алгоритма. Вместо этого применим сначала этот алгоритм к идеальному классическому газу и классическому магниту в магнитном поле и убедимся, что по прошествии достаточного «времени» алгоритм Метрополиса действительно дает распределение Больцмана.

Хотя мы выбираем в качестве π_s распределение Больцмана, возможны и другие варианты, например $\pi_s = \omega_s \exp(-\beta E_s)$. Принятие другого распределения, т.е. с не постоянным значением ω_s , в некоторых задачах оказывается полезным (см. список литературы). Кроме того, указанный выше выбор вероятности перехода W не является единственным приводящим к распределению Больцмана в асимптотическом пределе. Можно показать, что единственное требование заключается в том, чтобы W удовлетворяло принципу «детального равновесия»:

$$W(1 \rightarrow 2) \exp(-E_1/k_B T) = W(2 \rightarrow 1) \exp(-E_2/k_B T), \quad (16.9)$$

где $W(1 \rightarrow 2)$ — вероятность перехода в единицу времени системы из конфигурации 1 в конфигурацию 2.

16.3. ПРОВЕРКА РАСПРЕДЕЛЕНИЯ БОЛЬЦМАНА

Рассмотрим сначала идеальный классический газ и покажем, что алгоритм Метрополиса приводит к распределению Больцмана для отдельных микросостояний. Поскольку энергия идеального газа зависит только от скорости частиц, то любое микросостояние этой системы полностью описывается заданием скорости (или импульса) каждой частицы. Однако поскольку скорость есть непрерывная переменная, необходимо описать все возможные микросостояния так, чтобы их число было счетным. Как обычно, диапазон изменения скорости разбиваем на сколь угодно малые дискретные интервалы. Предположим, у нас есть $N = 10$ частиц и возможные значения скорости поделены на двадцать интервалов. Тогда полное число микросостояний будет составлять 20^{10} . Было бы не только трудно перенумеровать эти 20^{10} состояний, но получение сколь угодно точной оценки вероятности любого из них потребовало бы недопустимо много времени.

Чтобы обойти эту трудность, рассмотрим сначала моделирование методом Монте-Карло одномерного движения одиночной классической частицы. В программе **Boltzmann** массив P содержит вероятность $P(E)dE$ того, что энергия системы заключена между E и $E + dE$. Отметим, что при выбранных единицах измерения температуры $k_B = 1$, и поэтому температура измеряется в единицах энергии.

```

PROGRAM Boltzmann
! одномерный алгоритм Метрополиса для одной классической частицы
DIM P(0 to 500)
RANDOMIZE
CALL initial(nmcs, nequil, beta, vel, E, dvmax, nbin, del_E)
FOR imcs = 1 to nmcs + nequil
    CALL Metropolis(beta, vel, E, dvmax, accept)
    ! накапливаем данные после каждого пробного изменения
    IF imcs > nequil then CALL data(E, vel, Ecum, E2cum, vcum, P, nbin, del_E)
NEXT imcs
CALL averages(nmcs, vcum, Ecum, E2cum, P, accept, beta, nbin)
END

SUB initial (nmcs, nequil, beta, vel, E, dvmax, nbin, del_E)
    INPUT prompt "число шагов Монте-Карло = ": nmcs
    INPUT prompt "абсолютная температура = ": T
    INPUT prompt "начальная скорость = ": vel
    INPUT prompt "максимальное изменение скорости = ": dvmax
    LET beta = 1/T
    LET nequil = 0.1*nmcs
    LET E = 0.5*vel*vel
    LET del_E = 0.05          ! интервал разбиения энергии
    LET nbin = 4*T/del_E     ! максимальное число интервалов
END SUB

```

```
SUB Metropolis (beta,vel,E,dvmax,accept)
  LET dv = (2*rnd - 1)*dvmax      ! пробное изменение скорости
  LET vtrial = vel + dv           ! пробная скорость
  LET de = 0.5*(vtrial*vtrial - vel*vel) ! пробное изменение энергии
  IF de > 0 then
    IF exp(-beta*de) < rnd then
      EXIT SUB                    ! шаг не принимается
    END IF
  END IF
  LET vel = vtrial
  LET accept = accept + 1
  LET E = E + de
END SUB

SUB data (E,vel,Ecum,E2cum,vcum,P(),nbin,del_E)
  LET Ecum = Ecum + E
  LET E2cum = E2cum + E*E
  LET vcum = vcum + vel
  CALL probability(E,P,nbin,del_E)
END SUB

SUB probability(E,P(),nbin,del_E)
  LET ibin = E/del_E
  IF ibin > nbin then LET ibin = nbin
  LET P(ibin) = P(ibin) + 1
END SUB
```

```

SUB averages ( nmcs, vcum, Ecum, E2cum, P(), accept, beta, nbin)
  LET norm = 1/nmcs
  LET accept = accept*norm           ! коэффициент принятия
  LET Eave = Ecum*norm               ! средняя энергия
  LET E2ave = E2cum*norm
  LET vave = vcum*norm               ! средняя скорость
  PRINT "средняя энергия ="; Eave
  LET sigma2 = E2ave - Eave*Eave
  PRINT "средняя скорость ="; vave
  PRINT "коэффициент принятия ="; accept
  PRINT "sigma = "; sqrt(sigma2)
  PRINT "bin", "P(E)", "log P(E)"
  FOR ibin = 0 to nbin
    IF P(ibin) > 0.01*norm then ! печатаются только существенные P
      LET prob = p(ibin)*norm
      PRINT ibin, prob, log(prob)
    END IF
  NEXT ibin
END SUB

```

ЗАДАЧА 16.1. Распределение Больцмана

а. Рассмотрим движение только одной частицы; в этом случае ее микросостояния можно характеризовать ее энергией. Строго говоря, каждой энергии отвечают два микросостояния. Почему? С помощью программы **Boltzmann** определите вид распределения вероятностей, порождаемый алгоритмом Метрополиса. Возьмите $T = 1.0$, $dv_{max} = 2.0$, $nmcs = 1000$, $v_{initial} = 0$ и вычислите среднюю энергию, среднюю скорость и плотность вероятности $P(E)$.

б. Является ли $P(E)$ возрастающей или убывающей функцией от E ? График зависимости $\ln P(E)$ от E должен представлять собой прямую линию с угловым коэффициентом, равным $-1/T$. Увеличивайте $nmcs$ до тех пор, пока $P(E)$ хотя бы приближенно не примет экспоненциальную форму.

в. Насколько хорошо результаты для средней энергии и средней скорости, полученные при $nmcs = 1000$, совпадают с соответствующими точными значениями? Надо ли выбирать $nmcs$ таким же большим, как в п. «б», чтобы получить приемлемую точность?

г. Чтобы посмотреть не зависят ли получаемые результаты от начальных условий, задайте $v_{initial} = 2$ и вычислите средние энергию и скорость. Влияет ли на что-нибудь значение величины d_{max} ?

*ЗАДАЧА 16.2. Плоский спин во внешнем магнитном поле

а. Рассмотрим классический плоский магнит с магнитным моментом μ_0 . Этот магнит может ориентироваться в любом направлении в плоскости $x-y$. Энергия взаимодействия между спином и внешним магнитным полем B , направленным по оси y , равна $-\mu_0 B \cos \phi$, где ϕ — угол между спиновым моментом и B . Каковы возможные микросостояния этой системы? Составьте программу по методу Монте-Карло, в которой осуществляется выборка конфигураций такой системы, находящейся в тепловом равновесии с тепловой баней при температуре T . Рассчитайте среднюю энергию как функцию отношения $\mu_0 B / k_B T$.

б. Вычислите плотность вероятности $P(E)$ и исследуйте полученные результаты таким же образом, как это делалось в задаче 16.1.

ЗАДАЧА 16.3. Одномерное моделирование классического идеального газа

а. Модифицируйте программу **Boltzmann** для моделирования одномерного идеального газа, состоящего из N частиц. Положите $N = 20$, $T = 100$ и $nmcs = 200$. Сообщите всем частицам¹ одинаковую начальную скорость $v_{initial}$ и в первом варианте задайте ее равной 10. Определите значение d_{max} , максимальное изменение скоростей, исходя из условия, чтобы коэффициент принятия составил приблизительно 50%. Чему равны средняя кинетическая энергия и средняя скорость частиц?

б. Можно было бы ожидать, что полная энергия идеального газа остается постоянной, поскольку частицы не взаимодействуют друг с другом и, следовательно, не могут непосредственно обмениваться энергией. Чему равно значение начальной энергии системы для $v_{initial} = 10$? Остается ли полная энергия постоянной? Если энергия не остается постоянной, объясните, как она изменяется. Объясните, почему измеренная средняя скорость частиц приблизительно равна нулю даже при ненулевых начальных скоростях частиц.

в. Каков простой критерий «теплового равновесия»? Оцените число шагов Монте-Карло на частицу, необходимых для достижения систе-

мой теплового равновесия. Какими надо выбрать начальные скорости, чтобы система достигала теплового равновесия при температуре T за минимальное время?

г. Вычислите среднюю энергию на частицу для $T = 10, 100$ и 400 . Чтобы средние вычислялись после достижения системой теплового равновесия, обращайтесь к подпрограмме **DATA** только после того, как равновесие уже достигнуто. Увеличивайте число шагов Монте-Карло до тех пор, пока искомые средние не перестанут заметно меняться. Чему приблизительно равно значение nmc s, необходимое для установления теплового равновесия при $N = 10$ и $T = 100$ и при $N = 40$ и $T = 100$? Если в двух случаях получаются разные значения nmc s, то объясните причину этого.

д. Вычислите вероятность $P(E)dE$ того, что величина полной энергии системы N частиц лежит между E и $E + dE$. Как вы считаете, будет ли $P(E)$ пропорциональна $\exp(-E/k_B T)$? Постройте график зависимости $P(E)$ от E и опишите качественно ее поведение. Является ли график зависимости $\ln P(E)$ прямой линией? Опишите качественные особенности кривой $\ln P(E)$ как функции от E .

е. Вычислите среднюю энергию для $T = 10, 20, 30, 90, 100$ и 110 и оцените теплоемкость.

ж. Вычислите средний квадрат флуктуаций энергии $\langle \Delta E^2 \rangle = \langle E^2 \rangle - \langle E \rangle^2$ для $T = 10$ и $T = 40$. Сравните величину отношения $\langle \Delta E^2 \rangle / T^2$ с теплоемкостью, определенной в п. «е».

Возможно, вы удивились, обнаружив в задаче 16.3д, что функция $P(E)$ имеет вид гауссоиды, центрированной относительно средней энергии системы. Иначе говоря, если *микроскопические* состояния системы распределены в соответствии с распределением Больцмана, то функция распределения *макроскопической* величины, такой, как полная энергия, имеет форму острого пика около своего среднего значения.

Теперь рассмотрим моделирование модели Изинга, находящейся в равновесии с тепловой баней. Как говорилось в гл. 15, энергия модели Изинга определяется выражением

$$E = -J \sum_{\langle i, j \rangle} s_i s_j - \mu_0 H \sum_i s_i \quad (16.10)$$

где $s = \pm 1$, J есть мера силы взаимодействия спинов и первая сумма берется по всем парам спинов, являющихся ближайшими соседями. Второй член в (16.10) отвечает энергии взаимодействия магнитного момента с внешним магнитным полем.

В задаче 16.4 для моделирования одномерной модели Изинга мы используем алгоритм Метрополиса с динамикой опрокидывания спинов. В указанной динамике возможным пробным изменением является опрокидывание спина $s_i \rightarrow -s_i$. Обратите внимание на то, что параметры J , k_B и T не появляются порознь, а встречаются только в безразмерной комбинации $J/k_B T$. Если это не оговорено особо, мы будем измерять температуру в таких единицах, чтобы $J/k_B = 1$. Большинство наших моделирований будет проводиться для случая $H = 0$.

ЗАДАЧА 16.4. Одномерная модель Изинга

а. Составьте программу моделирования методом Монте-Карло одномерной модели Изинга, находящейся в равновесии с тепловой баней. Указания: модифицируйте в программе `Ising_demon` (см. гл. 15) подпрограмму `changes` или посмотрите, как сделана программа `Ising` из разд. 16.4, в которой реализован алгоритм Метрополиса для двумерной модели Изинга. Рассмотрите только случай нулевого магнитного поля. Ваша программа должна вычислять среднюю энергию и намагниченность решетки и вычерчивать микроскопическое состояние системы после каждого шага Монте-Карло на спин. Используйте периодические краевые условия.

б. Задайте $N = 20$, $T = 1.0$, $nmcs = 30$ и все спины в начальном состоянии ориентированными «вверх», т.е. $s_i = +1$. Чему равна начальная «температура» системы? Визуально изучите микроскопическое состояние системы после каждого шага Монте-Карло и оцените время, которое требуется системе для достижения равновесия.

в. Измените начальное условие так, чтобы все спины в начальный момент были ориентированы хаотично. Чему равна начальная «температура» системы? Визуально изучите микроскопическое состояние системы после каждого шага Монте-Карло и оцените время, которое требуется системе для достижения равновесия.

г. Задайте $N = 20$ и просчитайте эволюцию системы к равновесию в течение 100 шагов Монте-Карло на спин. Используйте по крайней мере 200 шагов Монте-Карло на спин и определите зависимость

средней энергии $\langle E \rangle$ и намагниченности $\langle M \rangle$ от T в интервале T от 0.5 до 5.0. Начертите кривую $\langle E \rangle$ как функцию от T и рассмотрите ее качественные особенности. Сравните результаты вычислений средней энергии с точным ответом (для нулевого магнитного поля)

$$\langle E \rangle = -Nth \frac{J}{k_B T} . \quad (16.11)$$

Какие результаты получились у вас для $\langle M \rangle$? Зависят ли они от начальной конфигурации?

д. Является ли коэффициент принятия возрастающей функцией от T или убывающей? Повышается или понижается эффективность алгоритма Метрополиса при низких температурах?

е. Вычислите плотность вероятности $P(E)$ для системы из 50 спинов с $T = 1.0$. Примите $nmcsc = 200$. Постройте зависимость $\ln P(E)$ от $(E - \langle E \rangle)^2$ и обсудите ее качественные особенности.

*ЗАДАЧА 16.5. Другой выбор W

Другой вариант вероятности перехода, отвечающий принципу детального равновесия (16.9), определяется выражением

$$W'(1 \rightarrow 2) = \frac{\exp(-E_2/k_B T)}{\exp(-E_1/k_B T) + \exp(-E_2/k_B T)} . \quad (16.12)$$

Для реализации данной симметричной формы вероятности перехода будем вычислять величину

$$W' = \frac{\exp(-\Delta E/k_B T)}{\exp(-\Delta E/k_B T) + 1} . \quad (16.13)$$

Отметим, что если $\Delta E = 0$, то $W' = 1/2$ и пробная конфигурация с равной вероятностью может быть принята или отклонена. Если $J/k_B T \gg 1$, то при каком виде W' получится система, быстрее достигающая равновесия? Справедливо ли то же самое утверждение для $J/k_B T \ll 1$?

16.4. МОДЕЛИРОВАНИЕ ДВУМЕРНОЙ МОДЕЛИ ИЗИНГА

Одним из наиболее интересных наблюдаемых в природе явлений является ферромагнетизм. Возможно, вы знакомы с материалами, такими как железо и никель, которые проявляют спонтанную намагниченность в отсутствие внешнего магнитного поля. Эта ненулевая намагниченность присутствует только при температуре ниже вполне определенной температуры T_c , называемой температурой Кюри или критической температурой. При температурах $T > T_c$ намагниченность пропадает. Тем самым T_c отделяет хаотическую фазу при $T > T_c$ от ферромагнитной фазы при $T < T_c$.

Несмотря на то что механизм ферромагнетизма имеет квантовомеханическую природу и служит объектом интенсивных современных экспериментальных и теоретических исследований, изучение классической модели Изинга в двумерной и трехмерной постановках много дало для понимания свойств магнитных систем в окрестности фазового перехода. Однако в силу своего классического характера и пренебрежения другими спиновыми компонентами модель Изинга не дает полного описания ферромагнетизма, особенно при температурах значительно ниже T_c . В частности, в модели Изинга предполагается, что отдельные моменты локализованы, и поэтому модель Изинга не применима к металлам, таким, как железо и никель.

Для исследования свойств модели Изинга нам необходимо конкретизировать, какие физические свойства представляют интерес, и разработать программу их вычисления. К рассматриваемым обычно равновесным характеристикам относятся средняя энергия $\langle E \rangle$, средняя намагниченность $\langle M \rangle$, теплоемкость C и магнитная восприимчивость χ . Один из методов измерения C при постоянном внешнем магнитном поле вытекает из определения

$$C = \frac{\partial \langle E \rangle}{\partial T}. \quad (16.14a)$$

Другой метод измерения C основан на использовании связи теплоемкости со статистическими флуктуациями полной энергии в каноническом ансамбле:

$$C = \frac{1}{k_B T^2} (\langle E^2 \rangle - \langle E \rangle^2). \quad (16.14b)$$

Магнитная восприимчивость χ является другим примером «функции отклика», поскольку она характеризует способность системы «откли-

каться» или опрокидываться за счет изменения во внешнем магнитном поле. Изотермическая магнитная восприимчивость при нулевом поле определяется посредством термодинамической производной

$$\chi = \lim_{H \rightarrow 0} \frac{\partial \langle M \rangle}{\partial H}. \quad (16.15a)$$

Восприимчивость при нулевом поле можно связать с флуктуациями намагниченности в системе:

$$\chi = \frac{1}{k_B T} (\langle M^2 \rangle - \langle M \rangle^2), \quad (16.15b)$$

где $\langle M \rangle$ и $\langle M^2 \rangle$ отвечают нулевому полю. Соотношения (16.14b) и (16.15b) являются примерами общего соотношения между функциями отклика и равновесными флуктуациями. Для полноты они выводятся в приложении 16A.

Теперь, когда мы конкретизировали некоторые интересующие нас равновесные величины, нам надо реализовать алгоритм Метрополиса. Этот алгоритм был сформулирован в разд. 16.2 как рецепт для генерации состояний с требуемым распределением Больцмана. Однако алгоритм с динамикой «опрокидывания спина» является вполне разумным приближением к реальной динамике анизотропного магнита, спины которого связаны с колебаниями решетки. В этом случае связь приводит к беспорядочному опрокидыванию спинов и можно ожидать, что один шаг Монте-Карло на спин пропорционален среднему времени между опрокидываниями спинов, наблюдаемому в лабораторном эксперименте. Следовательно, мы можем рассматривать динамику опрокидывания спина как настоящий нестационарный процесс и наблюдать релаксацию к равновесию по прошествии достаточно больших времен.

Ниже мы разработаем программу Ising для моделирования методом Монте-Карло двумерной модели Изинга, находящейся в контакте с тепловой баней. С точки зрения вычислительных затрат одним из самых трудоемких элементов алгоритма Метрополиса является вычисление экспоненты $\exp(-\beta \Delta E)$, где $\beta = 1/T$. Однако, как видно из рис. 16.1, для модели Изинга имеется лишь небольшое число возможных значений $\beta \Delta E$. Поэтому с целью экономии машинного времени мы запоминаем несколько различных вероятностей опрокидываний спина в массиве w . Отметим, что если опрокидывание не принимается и оставляется старая конфигурация, то тепловое равновесие не будет правильно описываться, если при вычислении средних не учитывать старую конфигурацию

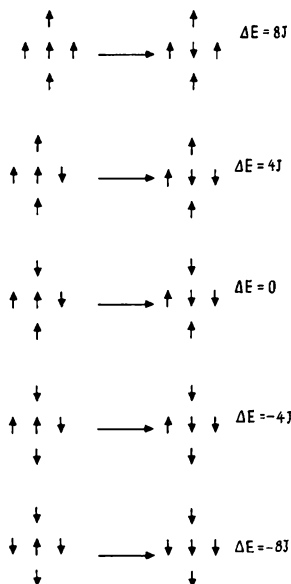


Рис. 16.1. Пять возможных переходов в модели Изинга квадратной решетки с динамикой опрокидывания спина. Изменения энергии равны соответственно $8J$, $4J$, 0 , $-4J$ и $-8J$.

еще раз. Отметим также, что в подпрограмме **data** значения всех физических измеряемых величин регистрируются после каждого шага Монте-Карло; оптимальный промежуток времени для «снятия показаний» физических величин исследуется в задаче 16.8.

```

PROGRAM Ising                                ! алгоритм Метрополиса
! для двумерной модели Изинга на квадратной решетке
DIM spin(32,32),w(-4 to 4)
CALL initial(N,L,T,nmcs,spin,E,M,w)
FOR imcs = 1 to nmcs
    CALL Metropolis(N,L,spin,E,M,w,ratio)
    CALL data(E,M,ecum,e2cum,mcum,m2cum)
NEXT imcs
CALL output(N,nmcs,ecum,e2cum,mcum,m2cum,ratio)
CALL save_config(N,L,T,spin)
END

```

В подпрограмме **initial** выбирается начальная ориентация спинов, вычисляются начальные значения энергии и намагниченности и рассчитываются значения различных вероятностей перехода. Более подробно начальные условия рассмотрены в задаче 16.6. Обратите внимание на то, что при вычислении полной энергии мы учитываем взаимодействие каждого спина только с ближайшими соседними спинами сверху и слева. Таким образом, энергия каждого спин-спинового взаимодействия вычисляется только один раз и тем самым мы избегаем двойного счета.

```

SUB initial (N, L, T, nmcs, spin(, ), E, M, w())
  RANDOMIZE
  INPUT prompt "линейный размер решетки = ": L
  LET N = L*L                      ! число спинов
  INPUT prompt "число шагов Монте-Карло на спин = ": nmcs
  INPUT prompt "приведенная температура = ": T
  FOR y = 1 to L                    ! случайная начальная конфигурация
    FOR x = 1 to L
      IF rnd < 0.5 then
        LET spin(x,y) = 1          ! спин вверх
      ELSE
        LET spin(x,y) = -1
      END IF
      LET M = M + spin(x,y)         ! суммарная намагниченность
    NEXT x
  NEXT y
  FOR y = 1 to L                    ! вычисление начальной энергии E
    IF y = L then
      LET up = 1                    ! периодические краевые условия
    ELSE
      LET up = y + 1
    END IF
    FOR x = 1 to L
      IF x = L then
        LET right = 1
      ELSE
        LET right = x + 1
      END IF
      LET sum = spin(x,up) + spin(right,y)
      LET E = E - spin(x,y)*sum     ! полная энергия
    NEXT x
  NEXT y

```

```

NEXT γ
! вычисление вероятностей перехода в порядке возрастания энергии
! индекс массива w равен сумме значений спинов ближайших соседей
LET e4 = exp(-4/T)
LET e8 = e4*e4
LET w(4) = e8
LET w(-4) = 1/e8
LET w(2) = e4
LET w(-2) = 1/e4
LET w(0) = 1
END SUB

```

Мы вправе реализовать алгоритм Метрополиса за два шага. Сначала проверяется условие $\Delta E \leq 0$ и в случае его выполнения пробное опрокидывание принимается. Если же условие не выполняется, генерируется случайное число и сравнивается с $\exp(-\beta\Delta E)$. Пробное опрокидывание принимается, если это случайное число меньше или равно вероятности перехода.

```

SUB Metropolis (N,L,spin(.),E,M,w(.),ratio)
FOR ispin = 1 to N
  LET x = int(L*rnd + 1)      ! разыгрываем координату по x
  LET y = int(L*rnd + 1)      ! разыгрываем координату по y
  ! находим соседние спины, используя периодические краевые условия
  CALL periodic(x,y,L,spin,sum) ! считаем сумму 4 ближайших соседей
  IF spin(x,y)*sum <= 0 then
    CALL accept(x,y,M,E,sum,spin,ratio)
  ELSEIF rnd < w(sum) then
    CALL accept(x,y,M,E,sum,spin,ratio)
  END IF
NEXT ispin
END SUB

```

Типичная лабораторная система содержит по крайней мере 10^{23} спинов. По сравнению с этим число спинов, изучаемых в вычислительных экспериментах, обычно находится в пределах от 32^2 до 600^3 . Как мы уже указывали по другому поводу, применение периодических краевых условий минимизирует эффекты конечного размера. Однако недостаток периодических краевых условий состоит в том, что при них минимальное расстояние между спинами уменьшается до половины длины системы.

Возможны также и более сложные краевые условия. Например можно представить ситуацию, когда у поверхностных спинов имеются дополнительные соседи, ориентация которых связана со средним спином рассматриваемой конфигурации в данный момент времени. В подпрограмме *periodic* приняты периодические краевые условия, как более простые.

```

SUB periodic (x,y,L,spin(, ),sum)
  IF x = 1 then
    LET left = spin(L,y)
  ELSE
    LET left = spin(x - 1,y)
  END IF
  IF x = L then
    LET right = spin(1,y)
  ELSE
    LET right = spin(x + 1,y)
  END IF
  IF y = 1 then
    LET down = spin(x,L)
  ELSE
    LET down = spin(x,y - 1)
  END IF
  IF y = L then
    LET up = spin(x,1)
  ELSE
    LET up = spin(x,y + 1)
  END IF
  LET sum = left + right + up + down
END SUB

```

```

SUB accept (x,y,M,E,sum,spin(, ),ratio)
  LET spin(x,y) = -spin(x,y)
  ! нормировка для последующего вычисления коэффициента принятия
  LET ratio = ratio + 1
  LET M = M + 2*spin(x,y)
  LET E = E - 2*spin(x,y)*sum
END SUB

```

```
SUB data (E,M,есum,e2cum,mcum,m2cum)
  ! накопление данных после каждого шага Монте-Карло на спин
  LET еsum = еsum + E
  LET e2cum = e2cum + E*E
  LET mcum = mcum + M
  LET m2cum = m2cum + M*M
END SUB

SUB output (N,nmcs,есum,e2cum,mcum,m2cum,ratio)
  LET norm = 1/(nmcs*N)          ! средние на спин
  LET ratio = ratio*norm
  LET eave = еsum*norm
  LET e2ave = e2cum*norm
  LET mave = mcum*norm
  LET m2ave = m2cum*norm
  PRINT "коэффициент принятия = ",ratio
  PRINT "средняя энергия на спин = ",eave
  PRINT "средний квадрат энергии на спин = ",e2ave
  PRINT "средняя намагниченность = ", mave
  PRINT "средний квадрат намагниченности = ",m2ave
END SUB
```

В некоторых вычислительных экспериментах процесс установления равновесия может занимать заметную часть общего машинного времени. Практичнее всего в качестве начальных условий выбирать «равновесную» конфигурацию из какого-нибудь прежнего расчета, отвечающего температуре, близкой к требуемой. Следующая подпрограмма, осуществляющая запоминание последней конфигурации пропускаемого варианта, может быть включена в программу **Ising** после подпрограммы **output**.


```

SUB save_config(N,L,T,spin(.))
  INPUT prompt "имя файла для последней конфигурации = ": file$
  OPEN #2: name file$, access output, create new
  PRINT #2: T
  FOR  $\gamma$  = 1 to L
    FOR x = 1 to L
      PRINT #2: spin(x, $\gamma$ )
    NEXT x
  NEXT  $\gamma$ 
  CLOSE #2
END SUB

```

Сохраненной конфигурацией можно воспользоваться в последующих расчетах, добавив в подпрограмму **initial** следующие инструкции.

```

INPUT prompt "старая конфигурация (  $\gamma$  (да)/н (нет) )?": old$
IF old$ = "γ" or old$ = "Y" then
  INPUT prompt "имя файла?": file$
  OPEN #1: name file$, access input
  INPUT #1: T
  FOR  $\gamma$  = 1 to L
    FOR x = 1 to L
      INPUT #1: spin(x, $\gamma$ )
    NEXT x
  NEXT  $\gamma$ 
  CLOSE #1
ENDIF

```

ЗАДАЧА 16.6. Установление равновесия двумерной модели Изинга

а. Входными параметрами программы **Ising** являются линейный размер решетки L , число шагов Монте-Карло на спин nmc s и температура тепловой бани T . Выполните программу **Ising** с $L = 8$ и $T = 2$, при этом задайте все начальные спины направленными вверх. Чему равна начальная «температура» системы? Постройте график зависимости энергии и намагниченности от «времени» (числа шагов Монте-Карло на спин). Сколько времени требуется системе для достижения равновесия? После достижения системой равновесия запомните какую-нибудь типичную конфигурацию.

б. Визуально исследуйте несколько равновесных конфигураций. Что можно сказать о системе: «упорядочена» она или «хаотична»?

в. Выполните программу **Ising** с $L = 8$ и $T = 1.5$ и выберите начальную конфигурацию спинов такой же, как в п. «а», т.е. направьте все спины вверх. За какое время эта система достигает равновесия? Задайте теперь в качестве начальной конфигурации конфигурацию, запомненную в п. «а» для $T = 2.0$. Сравните относительные времена достижения равновесия для обеих конфигураций.

г. Визуально исследуйте несколько равновесных конфигураций для $T = 1.5$. Являются ли эти конфигурации более или менее упорядоченными, чем конфигурации в п. «а»?

Теперь, когда мы получили типичные равновесные конфигурации, хотелось бы рассчитать средние значения некоторых рассматриваемых физических величин. Предположим, что требуется вычислить среднее значение физической величины A . Обычно расчет A занимает много времени и поэтому желательно вычислять ее значения не чаще, чем это необходимо. Ясно, что нам не нужно вычислять A после опрокидывания только одного спина, поскольку значения A в обеих конфигурациях были бы почти одинаковыми. Идеальным было бы вычислять A для конфигураций, которые статистически независимы. Однако, поскольку «время корреляции» конфигураций *a priori* неизвестно, следует провести предварительные расчеты для оценки времени корреляции.

Один из способов определения временных интервалов, на которых конфигурации коррелированы, состоит в вычислении зависящих от времени автокорреляционных функций $C_M(t)$ и $C_E(t)$, определяемых соотношениями

$$C_M(t) = \langle M(t)M(0) \rangle - \langle M \rangle^2, \quad (16.16a)$$

$$C_E(t) = \langle E(t)E(0) \rangle - \langle E \rangle^2. \quad (16.16b)$$

Величины $M(t)$ и $E(t)$ представляют собой значения намагниченности и полной энергии системы в «момент времени» t , которым является число шагов Монте-Карло на спин. Заметим, что при $t = 0$ функция $C_M(t)$ пропорциональна магнитной восприимчивости, а $C_E(t)$ пропорциональна теплоемкости. Для достаточно больших t функции $M(t)$ и $M(0)$ будут некоррелированы и $\langle M(t)M(0) \rangle \rightarrow \langle M(t) \rangle \langle M(0) \rangle = \langle M \rangle^2$. Следова-

тельно, $C_M(t)$ и $C_E(t)$ должны стремиться к нулю при $t \rightarrow \infty$. Обычно считается, что $C_M(t)$ и $C_E(t)$ убывают со временем по экспоненциальному закону. Оценкой времени корреляции τ служит время, за которое $C(t)$ уменьшается в e раз по сравнению со своим значением при $t = 0$. Поскольку конфигурации, отстоящие на времена, меньшие τ , являются статистически коррелированными, мы будем вычислять требуемые физические величины для временных интервалов порядка τ , а не после каждого шага Монте-Карло на спин.

ЗАДАЧА 16.7. Время корреляции

а. Модифицируйте программу `Ising` так, чтобы вычислялись равновесные значения $C_M(t)$ и $C_E(t)$. Рассмотрите $L = 8$ и $T = 3.0$ (высокие T), $T = 2.3$ и $T = 1.5$ (низкие T). В качестве начального состояния используйте в моделировании равновесную конфигурацию, сформированную в задаче 16.6. Оцените время корреляции τ для флуктуаций энергии и намагниченности. Насколько сопоставимы времена корреляции этих двух флуктуаций? Как соотносятся ваши оценки τ с оценками времени релаксации, найденными в задаче 16.6?

*б. Чтобы описать релаксацию к равновесию как можно более реалистично, выбор спинов, подлежащих опрокидыванию, производится случайным образом. Однако, если нас интересуют только равновесные свойства, можно, по-видимому, сэкономить машинное время, если выбирать спины последовательно. Выясните, как изменится время корреляции, если спины выбираются последовательно, а не случайно: будет ли оно больше, меньше или останется примерно таким же. Если время корреляции увеличивается, то экономится ли все же время ЦП при последовательном выборе спинов? Почему нежелателен последовательный выбор спинов в одномерной модели Изинга?

ЗАДАЧА 16.8. Сравнение с точными результатами

Поскольку в программах моделирования методом Монте-Карло точные результаты получаются, вообще говоря, только после выборки бесконечного числа конфигураций, как можно убедиться в том, что алгоритм и наша программа работают правильно? Не существует никакого общего правила для определения необходимого числа статистических выборок или времени установления равновесия. Но, поскольку

ку систематические ошибки программирования и ошибки набивки легко пропустить, мы должны быть уверены, что программа способна воспроизводить с известными ограничениями точные результаты. Один из способов проверки программы **Ising** состоит в рассмотрении небольшой системы, для которой сумму по состояниям, а значит, среднюю энергию и намагниченность можно вычислить аналитически.

а. Вычислите аналитически зависимость E , M , C и χ от температуры T для двумерной модели Изинга с $L = 2$ и периодическими краевыми условиями. (Результат этого расчета приведен в приложении 16Б).

б. Воспользуйтесь программой **Ising** с $L = 2$ и оцените E , M , C и χ для $J/k_B T = 2.0$ и 3.0 . С помощью соотношений (16.146) и (16.156) вычислите соответственно C и χ . Сравните вычисленные значения с точными результатами, полученными в п. «а». Сколько необходимо шагов Монте-Карло на спин для получения E и M с точностью 1%?

16.5. ФАЗОВЫЙ ПЕРЕХОД ИЗИНГА

Теперь, когда мы проверили свою программу двумерной модели Изинга в различных режимах, можно заняться исследованием ее свойств. Первым делом посмотрим на некоторые количественные характеристики ферромагнитных систем в нулевом внешнем поле. Известно, что при $T = 0$ спины прекрасно ориентируются в любом направлении, т.е. средняя намагниченность на спин $m(T) = \langle M \rangle / N$ равна $m(T = 0) = \pm 1$. По мере роста T намагниченность $m(T)$ непрерывным образом падает, и при $T = T_c$ $m(T)$ полностью исчезает (рис. 16.2). Поскольку $m(T)$ подходит к нулю

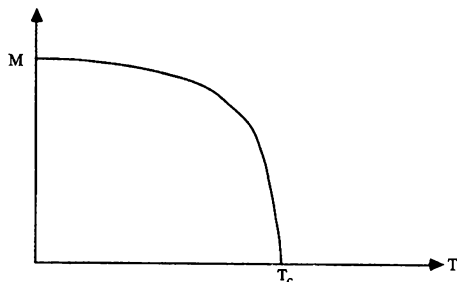


Рис. 16.2. Качественный вид температурной зависимости функции $m(T)$, намагниченности на спин, для непрерывного фазового перехода.

непрерывно, а не скачком, данный фазовый переход называется *непрерывным* в противоположность скачкообразному. (О последнем типе перехода говорят как о фазовом переходе *первого рода*.)

Каким же образом можно описать непрерывный магнитный фазовый переход? Точно так же, как мы описывали окрестность порога перколяции. Поскольку $m \neq 0$ означает, что суммарное число спинов ориентировано спонтанно, мы считаем m *параметром упорядочения* системы. В окрестности точки $T = T_c$ мы можем описать поведение многих физических величин степенным законом (см. табл. 12.1). Например, функцию m в окрестности T_c можно записать в виде

$$m(T) \sim (T_c - T)^\beta, \quad (16.17)$$

где β — критический показатель степени (не путайте его с величиной $1/k_B$). Хотя $m(T)$ в точке T_c обращается в нуль, термодинамические производные, такие, как теплоемкость и восприимчивость, при $T = T_c$ расходятся. Запишем

$$\chi \sim |T - T_c|^{-\gamma} \quad (16.18)$$

и

$$C \sim |T - T_c|^{-\alpha}. \quad (16.19)$$

Мы предположили, что χ и C описываются одними и теми же критическими показателями степени γ и α как слева, так и справа от точки T_c .

Другой мерой магнитных флуктуаций является линейный размер $\xi(T)$ характерного магнитного домена. Мы предполагаем, что при $T \gg T_c$ *длина корреляции* $\xi(T)$ по порядку величины равна периоду решетки. Поскольку по мере приближения T к T_c сверху корреляция в ориентации спинов увеличивается, $\xi(T)$ будет возрастать при приближении T к T_c . Расходимость $\xi(T)$ в окрестности T_c можно описать критическим показателем ν :

$$\xi(T) \sim |T - T_c|^{-\nu}. \quad (16.20)$$

Как было найдено в гл. 12 при рассмотрении перколяции, в конечной системе не может проявиться настоящий фазовый переход. Тем не менее можно ожидать, что если $\xi(T)$ меньше линейного размера L системы, то конечная система будет правильно передавать бесконечную систему. Иначе говоря, если T не слишком близка к T_c , то наши мо-

дельные расчеты должны давать результаты, соизмеримые с результатами для бесконечной системы. В следующей задаче мы получим предварительные данные о температурной зависимости m , $\langle E \rangle$, C и χ . Эти данные помогут нам понять качественную природу ферромагнитного фазового перехода в двумерной модели Изинга.

ЗАДАЧА 16.9. Качественное поведение двумерной модели Изинга

а. Модифицируйте программу **Ising** так, чтобы значения рассматриваемых физических величин вычислялись для статистически независимых конфигураций. Поскольку нам предстоит рассматривать модель Изинга для решеток с разными значениями L , удобно вычислять интенсивные характеристики, такие, как средняя энергия на спин, удельная теплоемкость (на спин) и восприимчивость на спин. Чтобы упростить обозначения, для экстенсивных и соответствующих интенсивных величин мы сохраним одинаковые обозначения.

б. С помощью модифицированной вами программы рассчитайте намагниченность на спин m , среднюю энергию на спин $\langle E \rangle$, удельную теплоемкость C и восприимчивость на спин χ . Примите $L = 4$ и рассмотрите T от 1.5 до 3.5 с шагом 0.2. В начальном состоянии для $T = 1.5$ задайте все спины ориентированными вверх. Для каждого варианта с температурой $T + \Delta T$ в качестве начальной конфигурации используйте равновесную конфигурацию из предыдущего варианта с температурой T . Поскольку в ходе ваших наблюдений все спины могут перевернуться и намагниченность поменяет знак, оценивайте среднее значение $|m|$, а не m . Используйте по крайней мере 200 шагов Монте-Карло на спин и оцените число равновесных конфигураций, необходимое для нахождения m и $\langle E \rangle$ с точностью примерно 5%. Нарисуйте графики зависимостей $\langle E \rangle$, $|m|$, C и χ от T и опишите качественную зависимость этих величин от температуры. Наблюдаете ли вы какие-нибудь признаки фазового перехода? Заметим, что можно сэкономить машинное время, воспользовавшись для решеток при разных температурах одним и тем же набором случайных чисел.

в. Повторите вычисления п. «б» для $L = 8$ и $L = 16$. Нарисуйте графики полученных оценок $\langle E \rangle$, $|m|$, C и χ как функций от T и опишите качественную зависимость этих величин от температуры. Наблюдаете ли вы какие-нибудь признаки фазового перехода? Для сравнения: опубликованные Ландау (см. список литературы) резуль-

таты по методу Монте-Карло для двумерной модели Изинга получены для L в диапазоне от 4 до 60 и с числом шагов Монте-Карло на спин от 10^4 до $2 \cdot 10^3$.

г. Для любого заданного значения L , например $L = 16$, выберите значение T , которое, по вашему мнению, отвечает температуре несколько ниже T_c . В начальной конфигурации вместо ориентации всех спинов вверх задайте случайные направления спинов. Чему равна начальная «температура» системы? Посмотрите, как эволюционируют спины во времени. Наблюдаете ли вы несколько доменов с положительной и отрицательной спонтанной намагниченностью? Как меняется намагниченность со временем? Наблюдаются ли в ней большие флуктуации? Опишите равновесную конфигурацию системы. Выявляется ли много доменов или только один? Почему мы предлагали в п. «б» начинать при низкой температуре и постепенно «нагревать» систему?

д. Поскольку все наши модельные расчеты выполнены для нулевого магнитного поля, то все направления эквивалентны. Отсюда для $T < T_c$ в предположении наличия спонтанной намагниченности можно было бы ожидать увидеть как положительные, так и отрицательные значения намагниченности. Пусть начальная конфигурация отвечает состоянию, в котором все спины ориентированы вверх и $T < T_c$. Рассчитываете ли вы увидеть отрицательные значения M ? Почему? Как вам кажется, наблюдаете ли вы больше или меньше отрицательных значений M для больших или меньших значений L ? Если бы вы исходили из начального состояния, в котором все спины были ориентированы вниз, то как вы думаете, наблюдали бы вы положительные значения M или отрицательные?

Наиболее серьезным ограничением для изучения в численном эксперименте фазовых переходов выступает относительно малый размер наших систем. Тем не менее мы видели в задаче 16.9, что даже системы всего лишь с $L = 4$ обнаруживают свойства, напоминающие фазовый переход. На рис. 16.3 приведены данные, полученные нами методом Монте-Карло для температурной зависимости удельной теплоемкости двумерной модели Изинга с $L = 8$ и $L = 32$. Видно, что C обнаруживает широкий максимум, который становится более четким для больших L . Ведет ли себя полученная вами зависимость C аналогичным образом?

Поскольку мы умеем моделировать только конечные решетки, то трудно получить оценки для α , β и γ непосредственно с помощью опре-

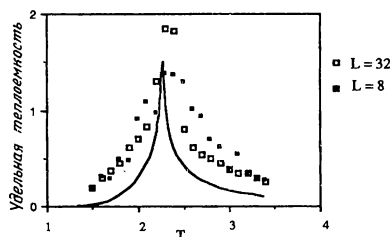


Рис. 16.3. Температурная зависимость удельной теплоемкости (на спин) в двумерной (квадрат $L \times L$) модели Изинга с периодическими краевыми условиями. Для каждого значения температуры использовалось тысяча шагов Монте-Карло на спин. Сплошной кривой показана температурная зависимость удельной теплоемкости в пределе бесконечной решетки.

делений (16.17)–(16.19). Вместо этого, как мы узнали в гл. 12, можно осуществлять перемасштабирование и экстраполировать результаты для конечного L на $L \rightarrow \infty$. Например, из рис. 16.3 видно, что температура, при которой C имеет максимум, лучше определяется для ббльших значений L . Такое поведение обеспечивает простой способ определения температуры фазового перехода $T_c(L)$ конечной системы. Согласно теории перемасштабирования, критическая температура $T_c(L)$ ведет себя как

$$T_c(L) - T_c(L = \infty) \sim \alpha L^{-1/\nu}, \quad (16.21)$$

где α — постоянная, а ν определена в (16.20). Поскольку, как мы ожидаем, конечность размера решетки важна, ибо

$$\xi(T) \sim L \sim |T - T_c|^{-\nu}, \quad (16.22)$$

то можно предположить, что температурная зависимость M , C и χ для конечных L заменяется на

$$m(T) \sim (T_c - T)^\beta \rightarrow L^{-\beta/\nu}, \quad (16.23)$$

$$C(T) \sim |T - T_c|^{-\alpha} \rightarrow L^{\alpha/\nu}, \quad (16.24)$$

$$\chi(T) \sim |T - T_c|^{-\gamma} \rightarrow L^{\gamma/\nu}. \quad (16.25)$$

В задаче 16.10 мы воспользуемся соотношениями (16.23)–(16.25) для оценки критических показателей.

ЗАДАЧА 16.10. Перемасштабирование и оценка критических свойств двумерной модели Изинга

а. Используя соотношение (16.21) с известным точным результатом $\nu = 1$, оцените величину T_c на бесконечной квадратной решетке. Поскольку получить точное значение T_c для небольших решеток трудно, в остальных пунктах данной задачи мы будем использовать точный ответ $k_B T_c / J = 2 / \ln(1 + \sqrt{2}) \approx 2.269$.

б. Определите удельную теплоемкость C , $|m|$ и восприимчивость χ при $T_c = 2.269$ для $L = 2, 4, 8$ и 16 . Количество шагов Монте-Карло на спин выберите из соображений практической целесообразности. Постройте графики зависимостей логарифмов $|m|$ и χ от L и с помощью масштабных соотношений (16.23)–(16.25) определите критические показатели β и γ . Примите, что точный результат равен $\nu = 1$. Получаются ли графики $|m|$ и χ в двойном логарифмическом масштабе близкими к прямым линиям? Сравните полученные оценки для β и γ с точными значениями, приведенными в табл. 12.1.

в. Начертите в двойном логарифмическом масштабе зависимость C от L . Если ваши данные для C достаточно точны, вы увидите, что получилась не прямая линия, а некоторая кривая. Причина этой кривизны связана с тем, что для двумерной модели Изинга величина α в формуле (16.19) равняется нулю и поэтому (16.24) необходимо интерпретировать как

$$C \sim C_0 \ln L. \quad (16.26)$$

Согласуются ли ваши данные для C с формулой (16.26)? Точное значение коэффициента пропорциональности C_0 в (16.26) приблизительно равно 0.4995.

До сих пор мы проводили все расчеты модели Изинга с квадратной решеткой. Поскольку природа не квадратна, то вас, возможно, интересует вопрос, зависят ли критическая температура и критические показатели от симметрии и размерности решетки. Опираясь на приобретенный вами в гл. 12 опыт по исследованию перколяционного перехода, вы, возможно, уже знаете ответ. Как вы считаете, больше ли T_c (при фиксированном J) для треугольной решетки, чем для квадратной решетки, или меньше?

***ЗАДАЧА 16.11. Влияние симметрии и размерности на критические свойства модели Изинга**

а. Природа треугольной решетки рассматривалась в гл. 11 (рис. 11.3). Основное отличие треугольной решетки от квадратной касается координационного числа. Проведите необходимые изменения в своей программе модели Изинга, например определите возможные переходы и значения коэффициента перехода. Вычислите C и χ для различных значений T в интервале $[1.0, 5.0]$. Примите, что $\nu = 1$ и с помощью перемасштабирования оцените T_c в пределе бесконечной треугольной решетки. Сравните полученные оценки T_c с известным значением (с точностью три десятичных знака) $k_B T_c / J = 3.641$. (Моделирование моделей Изинга с треугольной решеткой имеет отношение к пониманию экспериментально наблюдаемых фаз веществ, которые могут поглощаться на поверхности графита.)

б. Поскольку для трехмерной модели Изинга не имеется никаких точных результатов, расчеты трехмерной модели Изинга методом Монте-Карло представляют большой интерес. Составьте программу моделирования методом Монте-Карло модели Изинга в простой кубической решетке (координационное число равно шести). Вычислите C и χ для T от 3.2 до 5.0 с шагом 0.2 для различных значений L . Оцените $T_c(L)$ по максимуму C и χ . Какая температура $T_c(L)$ сильнее зависит от L — та, которая найдена из $C(T)$, или найденная из $\chi(T)$? Используя те значения $T_c(L)$, которые сильнее зависят от L , нарисуйте график $T_c(L)$ как функции $L^{-1/\nu}$ для разных значений ν в диапазоне от 0.5 до 1.0. Покажите, что экстраполированное значение $T_c(L = \infty)$ почти не зависит от величины ν . Сравните свою оценку для $T_c(L = \infty)$ с наилучшим известным значением (с точностью четыре десятичных знака) $k_B T_c / J = 4.5108$.

в. Вычислите $|m|$, C и χ при $T_c = 4.5108$ для различных значений L в простой кубической решетке. С помощью перемасштабирования оцените отношения β/ν , α/ν и γ/ν . Значения критических показателей для трехмерной модели Изинга приведены в табл. 12.1. Отметим, что Ландау рассматривал свойства простой кубической решетки Изинга конечных размеров для L от 6 до 20; для расчета средних после установления равновесия использовалось 2000—5000 шагов Монте-Карло на спин.

***ЗАДАЧА 16.12. Критический спад**

а. Рассмотрите двумерную модель Изинга на квадратной решетке с $L = 16$. Вычислите время корреляции τ для $T = 2.5, 2.4$ и 2.3 . Покажите, что по мере приближения к критической температуре величина τ возрастает—этот физический эффект называется *критическим спадом*.

б. Поскольку для бесконечной решетки величина τ в окрестности T_c неограниченно возрастает, можно ввести «динамический критический показатель» Δ , определяемый формулой $\tau \sim (T - T_c)^{-\Delta}$. В конечной решетке мы имеем при $T = T_c$ соотношение $\tau \sim L^z$. Используя аппарат перемасштабирования, получите связь z с Δ . Вычислите τ для разных значений L в квадратной решетке при $T = T_c$ и оцените z .

в. Величина τ , найденная нами в п.п. «а» и «б», частично зависит от выбора «динамики». Хотя мы генерировали пробное изменение, пытаясь опрокидывать один спин, не исключено, что другие способы пробных изменений, например одновременное опрокидывание двух и более спинов, могли бы оказаться более эффективными и дать меньшие времена корреляции. Поскольку критический спад затрудняет изучение фазовых переходов методом Монте-Карло, задача разработки более эффективных алгоритмов в окрестности фазовых переходов является в настоящее время очень важной.

16.6. ДРУГИЕ ПРИМЕНЕНИЯ МОДЕЛИ ИЗИНГА

Модель Изинга имеет столь обширные и широкие применения, что мы в состоянии упомянуть здесь только некоторые из них. Далее мы кратко опишем применения модели Изинга к фазовым переходам первого рода, решеточным газам, антиферромагнетизму и переходам порядок—беспорядок в бинарных сплавах.

До сих пор мы исследовали в модели Изинга непрерывные фазовые переходы и нашли, что энергия и намагниченность изменяются непрерывно с температурой, но термодинамические производные, такие, как удельная теплоемкость и восприимчивость, в пределе бесконечной решетки в окрестности T_c расходятся. В задаче 16.13 мы рассмотрим простой пример фазового перехода *первого рода*. Такие переходы сопровождаются конечным скачком термодинамических характеристик, таких как энергия и намагниченность.

***ЗАДАЧА 16.13. Двумерная модель Изинга во внешнем магнитном поле**

а. Модифицируйте свою двумерную программу модели Изинга, учтя в ней энергию взаимодействия с внешним магнитным полем H . Удобно измерять H в единицах $h = \mu_0 H / k_B$. Требуется вычислить среднюю намагниченность на спин m как функцию от h для $T < T_c$. Рассмотрите квадратную решетку с $L = 16$ и получите равновесное состояние при $T = 1.8$ и $h = 0$. Для нахождения $m(h)$ примите следующую процедуру.

- i. В качестве начальной конфигурации для $h_1 = \Delta h = 0.2$ используйте равновесную конфигурацию при $h = 0$.
- ii. Прежде чем вычислять средние, просчитайте эволюцию системы в течение 20 шагов по времени (шагов Монте-Карло на спин).
- iii. Усредните m по 80 временным шагам.
- iv. В качестве начальной конфигурации для $h_{n+1} = h_n + \Delta h$ используйте последнюю конфигурацию для $h_n = n\Delta h$.
- v. Повторяйте шаги (ii)–(iv) до тех пор, пока не получится $m \sim 0.95$.

Постройте график зависимости m от h . Соответствуют ли вычисленные значения m равновесным средним?

б. Уменьшайте h с шагом $\Delta h = 0.2$ до тех пор, пока h не пройдет через нуль и не получится $m \sim -0.95$. Остается ли m положительной при малых отрицательных h ? Соответствуют ли получаемые для отрицательных h значения m равновесным средним? Изобразите конфигурации спинов для нескольких значений h . Просматриваются ли признаки доменов? Продолжите полученный график зависимости m от h на отрицательные значения h .

в. Увеличивайте h до тех пор, пока кривая зависимости m от h не примет вид замкнутой петли. Чему равно значение m при $h = 0$? Это значение m и есть спонтанная намагниченность.

г. Фазовый переход первого рода характеризуется разрывностью (для бесконечной решетки) параметров порядка. Отличительным признаком перехода в данном случае является поведение m как функции от h . Чему равно полученное вами значение m для $h = 0.2$? Если функция $m(h)$ двузначна, то какое значение m отвечает равновесно-

му состоянию, т.е. абсолютному минимуму свободной энергии? Какое значение m соответствует *метастабильному* состоянию, т.е. локальному минимуму свободной энергии? Чему равны значения m для равновесного и метастабильного состояний при $h = -0.2$? Почему переход от положительных m к отрицательным отвечает фазовому переходу первого рода? Обратите внимание на то, что переходы первого рода обладают *гистерезисом* и свойства системы зависят от ее истории, например от того, является ли h возрастающей или убывающей функцией. Вследствие большого времени жизни метастабильных состояний в окрестности фазового перехода такие состояния можно ошибочно принять за равновесные. Поскольку вблизи непрерывного фазового перехода релаксация к равновесию становится очень долгой (см. задачу 16.12), система с непрерывным фазовым переходом может вести себя, как если бы она была в сущности в метастабильном состоянии. По этой причине различить вид фазового перехода, обращаясь только к численным экспериментам, очень трудно.

д. Повторите описанное выше моделирование для $T = 3.0$, т.е. для температуры выше T_c . Почему ваши результаты отличаются от результатов моделирования в п.п. «а» — «в», выполненных для $T < T_c$?

Модель Изинга может описывать и другие системы, которые, как может показаться, имеют мало общего с ферромагнетизмом. Например, мы можем интерпретировать модель Изинга как «решеточный газ», в котором состояние «вниз» обозначает узел решетки, занятый атомом, а состояние «вверх» — свободный узел. Каждый узел решетки может быть занят не более чем одним атомом. «Спины» взаимодействуют со своими ближайшими соседями, как раньше. Решеточный газ представляет собой грубую модель поведения реального газа атомов и имеет историческое значение как модель фазового перехода газ — жидкость и критической точки. Какие свойства решеточного и реального газов одинаковы? Какие свойства реального газа не учтены в решеточном газе?

Важным различием ферромагнетика и решеточного газа является то, что в последнем полное число атомов фиксировано, тогда как в ферромагнетике число спинов, находящихся в состояниях «вверх» и «вниз», может меняться. Поэтому для решеточного газа мы больше не вправе использовать динамику опрокидывания спина. Вариант динамики, при которой действительно сохраняется число спинов, ориентированных вверх и вниз, называется *динамикой спинового обмена*. В этой динамике

ке производится пробный обмен двух ближайших соседних спинов, а изменение энергии ΔE вычисляется. Критерий принятия или отвержения пробного изменения совпадает с использованным ранее. Какой физический процесс происходит в решеточном газе, если обмениваются два спина, один из которых ориентирован вверх, а другой вниз?

*ЗАДАЧА 16.14. Моделирование решеточного газа

а. Модифицируйте свою программу модели Изинга таким образом, чтобы вместо динамики опрокидывания спина использовалась динамика спинового обмена. Например, определите возможные значения ΔE на квадратной решетке, массив возможных значений вероятности перехода w и замените способ, которым осуществляется пробное изменение. Заметим, что число занятых узлов является сохраняющейся переменной и должно задаваться изначально. Если нас интересуют только средние значения статических величин, таких, как полная энергия, мы можем уменьшить время счета, отказавшись от обмена одинаковых спинов. Можно, например, завести список «связей» между занятыми и свободными узлами и производить пробные перемещения, выбирая связи из этого списка случайным образом. Однако, поскольку мы будем рассматривать только небольшие решетки, рекомендуется не беспокоиться о хранении указанного списка и строить пробное перемещение, выбирая просто спин и один из его ближайших соседей случайным образом.

б. Рассмотрите квадратную решетку с $L = 8$ и 32 узлами, в начальный момент занятыми. Определите среднюю энергию для T в диапазоне от 1.0 до 4.0. Изобразите зависимость средней энергии от температуры. Создается ли впечатление, что энергия изменяется непрерывно?

в. Повторите вычисления п. «б» с 44 занятыми в начальный момент узлами. Изобразите зависимость средней энергии от T . Меняется ли энергия непрерывно? Наблюдаете ли вы какие-нибудь признаки фазового перехода первого рода?

г. Поскольку спины соответствуют атомам, мы можем вычислить коэффициент одночастичной диффузии атомов. (Похожее моделирование см. в задаче 11.9.) Заведите массив для регистрации положения каждого спина (атома) как функции времени. Примите за нуль от-

счета времени какакой-нибудь момент после достижения равновесного состояния и вычислите $\langle R(t) \rangle^2$ — средний квадрат суммарного смещения на атом за t единиц времени. Если атомы испытывают случайное блуждание, коэффициент самодиффузии D в пределе $t \rightarrow \infty$ определяется выражением $D = (1/2dt) \langle R(t) \rangle^2$. Оцените величину D для разных температур и разного количества занятых узлов.

Хотя, по всей видимости, вы лучше знакомы с ферромагнетизмом, в действительности природа предоставляет нам больше примеров антиферромагнетизма. На языке модели Изинга антиферромагнетизм означает, что ближайшие соседние спины предпочитают ориентироваться в противоположных направлениях и параметр взаимодействия J отрицателен. Как мы увидим в задаче 16.15, антиферромагнитная модель Изинга на квадратной решетке по свойствам сходна с ферромагнитной моделью Изинга. Например, при нулевом магнитном поле энергия и удельная теплоемкость у них точно такие же и система обнаруживает фазовый переход при температуре Нееля T_N . С другой стороны, полная намагниченность и восприимчивость антиферромагнетика не проявляют никаких критических свойств вблизи T_N . Можно, однако, определить для квадратной решетки две подрешетки, как показано на рис. 16.4, и ввести «клеточную намагниченность» M_s , равную разности намагниченности этих двух подрешеток. Температурная зависимость M_s и соответствующая клеточная восприимчивость χ совпадают с аналогичными величинами в ферромагнитной модели Изинга.

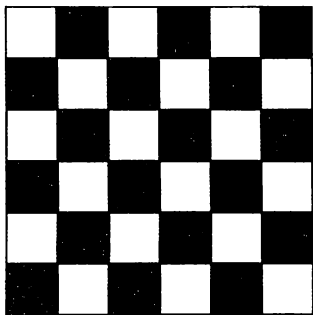


Рис. 16.4. Белые и черные клетки соответствуют двум подрешеткам.

*ЗАДАЧА 16.15. Антиферромагнитная модель Изинга

Поскольку в программе **Ising** величина J не присутствует явно, для моделирования антиферромагнетика нам надо поменять знак в вычислениях энергии. Это можно сделать, заменив в программе конструкцию $spin(x,y)*sum$ на $-spin(x,y)*sum$ везде, где она встречается. Для расчета клеточной намагниченности на квадратной решетке в качестве одной подрешетки мы определяем узлы (x,y) , для которых $\text{mod}(x,2) + \text{mod}(y,2) = 1$, а другая подрешетка образуется остальными узлами.

а. Модифицируйте программу **Ising** для моделирования антиферромагнитной модели Изинга на квадратной решетке. Рассмотрите $L = 16$, $h = 0$ и в качестве начального условия задайте все спины ориентированными вверх. Какая конфигурация спинов отвечает состоянию с наименьшей энергией? Вычислите температурные зависимости средней энергии, удельной теплоемкости, намагниченности и восприимчивости. Усматриваются ли на какой-нибудь из этих температурных зависимостей признаки фазового перехода?

б. Вычислите температурную зависимость M_s и клеточной восприимчивости χ , определяемой формулой [см. (16.15б)]

$$\chi_s = \frac{1}{k_B T} [\langle M_s^2 \rangle - \langle M_s \rangle^2]. \quad (16.27)$$

Удостоверьтесь, что температурная зависимость M_s для антиферромагнитной модели Изинга совпадает с температурной зависимостью M в ферромагнитной модели Изинга. Могли бы вы предсказать такое сходство, не проведя моделирования?

в. Вероятно, вы обратили внимание на то, что в п. «а» температурная зависимость функции χ обнаруживает точку заострения. Вычислите χ для разных значений L при $T = T_N = 2.269$. Проведите перемасштабирование и покажите, что χ не расходится.

г. Рассмотрите поведение антиферромагнитной модели Изинга на треугольной решетке. Задайте $L = 16$ и вычислите те же величины, что и в п. «а». Наблюдаете ли вы какие-нибудь признаки фазового перехода? Изобразите несколько конфигураций системы, относящихся к разным температурам. Наблюдаете ли вы при низких температурах

признаки множества малых доменов? Можете ли вы изобразить конфигурацию основного состояния? Единственное ли оно? Если вы не в силах найти единственное основное состояние, вы терпите то же фиаско, что и индивидуальные спины в треугольной антиферромагнитной модели Изинга. (В данном случае термин «потерпеть фиаско» описывает тот факт, что в треугольной решетке нет ни одной конфигурации, в которой в направлении какой-нибудь оси спины попеременно меняли бы знак.)

Другим важным применением модели Изинга является изучение разделения фаз в бинарном А—В-сплаве. Например, сплав, названный β -латунью, имеет низкотемпературную упорядоченную фазу, в которой оба компонента (медь и цинк) имеют одинаковые концентрации и образуют структуру хлорида цезия (см. книгу Киттеля). По мере увеличения температуры некоторые атомы цинка меняются местами с атомами меди, но система остается все еще «упорядоченной». Однако при температуре выше критической ($T_c = 742$ К) атомы цинка и меди перемешиваются и система «разупорядочивается». Этот переход является примером фазового перехода *порядок—беспорядок*. Как можно применить модель Изинга для перехода порядок—беспорядок?

Модель Изинга является лишь одной из нескольких моделей магнетизма. Модели Гейзенберга, Поттса, x — u и Ашкина—Теллера представляют собой примеры других моделей магнитных веществ и хорошо знакомы исследователям конденсированных сред. Моделирование методом Монте-Карло этих и других моделей внесло важный вклад в развитие нашего понимания фазовых переходов как в магнитных, так и в немагнитных веществах.

16.7. МОДЕЛИРОВАНИЕ КЛАССИЧЕСКИХ ЖИДКОСТЕЙ

Из повседневного опыта мы хорошо знаем о существовании трех фаз вещества: твердой, жидкой и газообразной (рис. 16.5). Исходя из этого опыта мы в состоянии различить эти три фазы по их свойствам. Например, твердые тела обладают жесткостью в противоположность газам и жидкостям, которые текут под действием касательного напряжения. Нашей задачей в этом разделе является получение с помощью метода Монте-Карло более детального представления о качественных различиях всех трех фаз.

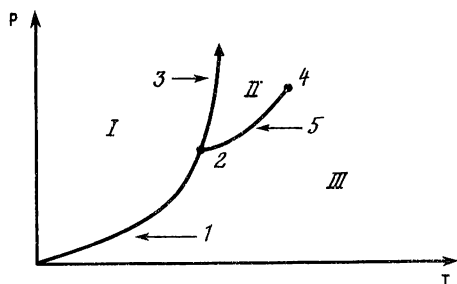


Рис. 16.5. Фазовая диаграмма для простого вещества.

Моделирование классических жидкостей методом Монте-Карло существенно упрощается за счет того обстоятельства, что координаты скорости (импульса) ни на что не влияют. Например, вклад координат скорости в среднюю энергию составляет $\frac{1}{2} k_B T$ на степень свободы. Следовательно, нам нужно производить выборку только положений молекул, т.е. «конфигурационных» степеней свободы. Возможно ли такое упрощение для квантовых систем?

К числу характеристик жидкости, представляющих интерес с точки зрения физики, относятся средняя энергия, удельная теплоемкость и уравнение состояния. Еще одной важной величиной является *парная корреляционная функция* $g(r)$, которую мы определим ниже. Предположим, что в области объема V содержится N частиц, так что средняя плотность равна $\rho = N/V$. (В двумерном и одномерном случаях объем V заменяется соответственно на площадь и длину.) Выберите одну из частиц, которую назовем опорной частицей, в качестве начала системы координат. Тогда вероятность обнаружить вторую частицу в интервале между r и $r + \Delta r$ равна $\rho g(r) dr$, где элемент «объема» $dr = 4\pi r^2 dr$ ($d = 3$), $2\pi r dr$ ($d = 2$) и $2 dr$ ($d = 1$). Мы предполагаем, что $g(r) \rightarrow 0$ при $r \rightarrow 0$, поскольку частицы не могут проникать друг в друга. Кроме того, мы предполагаем, что $g(r) \rightarrow 1$ при $r \rightarrow \infty$, поскольку неоднородность в окрестности частицы ограничена конечным размером области. Заметим, что если проинтегрировать $\rho g(r) dr$ по dr , то получится $N - 1$, т.е. полное число частиц в системе за вычетом опорной частицы. В задачах 16.16–16.18 мы найдем, что $g(r)$ является мерой флуктуаций плотности и тем самым локального «порядка» системы.

Исходя из функции $g(r)$ можно также получить несколько термодинамических характеристик. Например, если в системе присутствуют только двухчастичные взаимодействия, то можно показать, что средняя потенциальная энергия на частицу может быть представлена в виде

$$\frac{U}{N} = \frac{\rho}{2} \int g(r) V(r) dr. \quad (16.28)$$

Можно также воспользоваться вириалом силы (см. гл. 6) и записать уравнение состояния в виде

$$\frac{\beta P}{\rho} = 1 - \frac{\beta \rho}{2d} \int g(r) r \frac{dV(r)}{dr} dr. \quad (16.29)$$

Твердые сферы в d -мерном случае. С целью разделить эффекты короткодействующего отталкивания и длиннодействующего притяжения исследуем сначала модель *твердых сфер* с потенциалом межчастичного взаимодействия

$$V(r) = \begin{cases} +\infty, & r < \sigma, \\ 0, & r \geq \sigma. \end{cases} \quad (16.30)$$

Такая модель широко изучалась в одномерной (твердые стержни), двумерной (твердые диски) и трехмерной постановках как с помощью метода Монте-Карло, так и метода молекулярной динамики.

Поскольку взаимодействие (16.30) вообще не содержит притягивающей части, то нет и никакого фазового перехода газ—жидкость. Имеется ли фазовый переход между жидкими фазами при низких плотностях и твердыми фазами при высоких плотностях? Может ли твердая фаза образоваться при отсутствии притягивающего взаимодействия? В дальнейшем мы получим ответы на эти вопросы.

Какие физические величины имеет смысл рассматривать для системы твердых сфер? Никаких тепловых характеристик, таких, как средняя потенциальная энергия, нет, поскольку для твердых сфер эта величина всегда равна нулю. Наибольший интерес представляет $g(r)$, поскольку она дает информацию о корреляциях частиц и уравнении состояния. Если потенциал имеет вид (16.30), то можно показать, что выражение (16.29) приводится к виду

$$\frac{\beta P}{\rho} = \begin{cases} 1 + \frac{2\pi}{3} \rho \sigma^3 g(\sigma), & d = 3, \\ 1 + \frac{\pi}{2} \rho \sigma^2 g(\sigma), & d = 2, \\ 1 + \rho \sigma g(\sigma), & d = 1. \end{cases} \quad (16.31a)$$

$$(16.31b)$$

$$(16.31v)$$

Мы будем вычислять $g(r)$ для различных значений r и затем экстраполировать получаемые результаты на $r = \sigma$.

Алгоритм Метрополиса для твердых сфер легко реализовать, выбирая случайным образом частицу и перемещая ее в новое пробное положение. Если новое положение перекрывает другую частицу, данное перемещение отвергается и сохраняется старая конфигурация; в противном случае перемещение принимается. Максимальное смещение δ разумно, хотя и не обязательно оптимально, выбирать из тех соображений, чтобы при данном выборе δ принималась приблизительно половина пробных состояний. Самую большую трудность в реализации этого алгоритма представляет выявление пересечения двух частиц. Если число частиц не слишком велико, достаточно вычислять расстояния между пробной частицей и всеми остальными частицами. Эта простая процедура используется в программе `hard_disk` для системы твердых дисков. Для систем большего числа частиц данная процедура занимает слишком много времени. Как говорилось в гл. 6, лучше применять другой метод: разбивать систему на «ячейки» и проверять частицы только в своей и соседних ячейках.

```
PROGRAM hard_disk      ! алгоритм Метрополиса для твердых дисков
DIM x(50), y(50), gcum(100)
RANDOMIZE
CALL initial(N, x, y, nmcs, Lx, Ly, kx, ky, dr, bin, dxmax, dymax)
FOR imcs = 1 to nmcs
    CALL move(N, x, y, Lx, Ly, kx, ky, dxmax, dymax, accept)
    CALL correl(N, x, y, gcum, Lx, Ly, kx, ky, bin)
NEXT imcs
CALL output(N, nmcs, gcum, Lx, Ly, dr, bin, accept)
CALL save_config(N, Lx, Ly, x, y)
END
```

Выбрать начальные координаты дисков сложнее, чем может показаться на первый взгляд. Один метод заключается в размещении дисков в ящике случайным образом. Если какой-нибудь диск накладывается на уже имеющийся, то его бросают куда-то в другое место. Таким способом удастся довольно быстро вычислить возможную начальную конфигурацию в случае низкой плотности. Что касается достижения более высоких плотностей, то можно поступать следующим образом. Представьте себе, что стенки сдвигаются по направлению к центру ящика до тех пор, пока одна из стенок едва не коснется одного из дисков. Затем

диски перемещаются какое-то число шагов Монте-Карло и стенки вновь сдвигаются. Очевидно, что данная процедура тем тяжелее, чем больше плотность.

Другой метод отправляется от упорядоченного пространственного распределения частиц. Можно, например, начать с дисков, образующих треугольную решетку максимальной плотности, которая нас интересует. Далее можно расширить решетку, пересчитав пропорционально координаты дисков. В результате в новой конфигурации не возникнет ни одного пересечения дисков. Данная процедура осуществляется в подпрограмме **initial**. Заметим, что для выполнения условия, чтобы каждый диск имел одинаковое число ближайших соседей, необходимо рассматривать только четное число частиц. Систему единиц удобно выбрать такой, чтобы расстояния измерялись в радиусах твердого кб σ . Приведенная плотность равна $\rho^* = \rho/\sigma^d$, где d — размерность системы.

```
SUB initial (N, x(), y(), nmcs, Lx, Ly, kx, ky, dr, bin, dxmax, dymax)
  INPUT prompt "число частиц (четное) = ": N
  INPUT prompt "число стержней = ": Ny
  LET Nx = N/Ny          ! Nx должно быть целым
  INPUT prompt "длина ящика = ": Lx
  INPUT prompt "ширина ящика = ": Ly
  INPUT prompt "число шагов Монте-Карло на частицу = ": nmcs
  INPUT prompt "dr = ": dr      ! dr используется для вычисления g(r)
  LET bin = 1/dr
  INPUT prompt "максимальное перемещение по x = ": dxmax
  INPUT prompt "максимальное перемещение по y = ": dymax
  LET kx = 2/Lx          ! используем периодические краевые условия
  LET ky = 2/Ly
  INPUT prompt "старая конфигурация (y (да)/n (нет))? ": old$
  IF old$ = "y" or old$ = "Y" then
    INPUT prompt "Какое имя файла? ": fnm$
    OPEN #1: name fnm$, access input
    INPUT #1: Lxold, Lyold
    LET xexpand = Lx/Lxold
    LET yexpand = Ly/Lyold
```

```

FOR i = 1 to N
  INPUT #1: x(i), y(i)
  LET x(i) = x(i)*xexpand      ! расширяем старый ящик
  LET y(i) = y(i)*yexpand
NEXT i
CLOSE #1
ELSE                          ! создаем новую треугольную решетку
  LET dx = Lx/Nx              ! dx и dy должны быть > 1
  LET dy = Ly/Ny
  FOR j = 1 to Ny step 2
    FOR i = 1 to Nx
      LET m = m + 2
      LET x(m - 1) = (i - 0.75)*dx
      LET y(m - 1) = (j - 0.5)*dy
      LET x(m) = (i - 0.25)*dx
      LET y(m) = (j + 0.5)*dy
    NEXT i
  NEXT j
END IF
END SUB

```

В подпрограмме `move` выбирается случайная частица, генерируются ее пробные координаты с поправкой на периодические краевые условия и определяется ее пересечение с другими частицами.

```

SUB move (N, x(), y(), Lx, Ly, kx, ky, dxmax, dymax, accept)
  FOR i = 1 to N
    LET itrial = int(N*rnd + 1)      ! номер случайной частицы
    LET xtrial = x(itrial) + (2*rnd - 1)*dxmax ! пробная координата по x
    LET ytrial = y(itrial) + (2*rnd - 1)*dymax ! пробная координата по y
    CALL cell(xtrial, ytrial, Lx, Ly, kx, ky)
    CALL overlap(N, itrial, xtrial, ytrial, x, y, Lx, Ly, kx, ky, accept)
  NEXT i
END SUB

```

Существуют по крайней мере два способа обработки периодических краевых условий. До сих пор для проверки положения частицы мы прибегали к инструкциям `IF`. В подпрограммах `cell` и `separation` применя-

ется другой метод. Он основан на использовании свойств функции `truncate`, имеющейся в языке True BASIC. Данная функция, когда ее второй аргумент равен нулю, обрезает все знаки дробной части первого аргумента. Например, $\text{truncate}(9.3, 0) = 9$, $\text{truncate}(-9.3, 0) = -9$ и $\text{truncate}(9.7, 0) = 9$. В качестве побочного упражнения читателю предоставляется возможность определить, какой метод на вашем компьютере работает быстрее.

Заметим, что функция `truncate(x, 0)` языка True BASIC эквивалентна функции `int(x)` в Фортране. В противоположность этому функция `int` языка True BASIC выдает наибольшее целое, не превосходящее значения ее аргумента. Следовательно, в языке True BASIC имеем $\text{int}(9.3) = 9$, $\text{int}(-9.3) = -10$ и $\text{int}(9.7) = 9$.

```
SUB cell (xtrial, ytrial, Lx, Ly, kx, ky)
  ! Lx - ширина ящика, Ly - длина ящика
  ! правильные положения частиц заключены между 0 и L
  LET xtrial = xtrial - Lx*truncate(xtrial*kx - 1, 0)
  LET ytrial = ytrial - Ly*truncate(ytrial*ky - 1, 0)
END SUB

SUB separation (dx, dy, Lx, Ly, kx, ky)
  ! правильные положения частиц заключены между 0 и L/2
  LET dx = dx - Lx*truncate(kx*dx, 0)
  LET dy = dy - Ly*truncate(ky*dy, 0)
END SUB
```

Теперь, когда определено «правильное» положение пробного перемещения, мы обращаемся к подпрограмме `overlap` с целью определить, не перекрывается ли данным пробным перемещением положение какого-нибудь другого диска.

```

SUB overlap (N, itrial, xtrial, ytrial, x(), y(), Lx, Ly, kx, ky, accept)
  ! overlap if distance <= 1
  FOR j = 1 to N
    IF itrial <> j then
      LET dx = x(j) - xtrial
      LET dy = y(j) - ytrial
      CALL separation(dx, dy, Lx, Ly, kx, ky)      ! правильное расстояние
      LET R2 = dx*dx + dy*dy
      IF R2 < 1 then EXIT SUB                    ! диски перекрываются
    END IF
  NEXT j
  LET accept = accept + 1
  LET x(itrial) = xtrial
  LET y(itrial) = ytrial
END SUB

```

В подпрограмме **correl** вычисляется массив *gcum*, содержащий полное число частиц, отстоящих на расстояние от r до $r + \Delta r$ от заданной опорной частицы. Обратите внимание на то, что массив *gcum* находится для $\frac{1}{2}N$ опорных частиц. Несмотря на то что подпрограмма **correl** вызывается после каждого шага Монте-Карло, возможно, стоит вычислять корреляции пореже.

```

SUB correl (N, x(), y(), gcum(), Lx, Ly, kx, ky, bin)
  ! параметр bin равен 1/dr, где dr - расстояние между кольцами
  FOR i = 1 to N - 1
    FOR j = i + 1 to N
      LET dx = x(i) - x(j)
      LET dy = y(i) - y(j)
      CALL separation(dx, dy, Lx, Ly, kx, ky)
      LET R2 = dx*dx + dy*dy
      LET r = truncate(bin*sqr(R2), 0)
      LET gcum(r) = gcum(r) + 1
    NEXT j
  NEXT i
END SUB

```

В подпрограмме **output** получается нормированная парная корреляционная функция $g(r)$, для чего массив *gcum* делится на число выборок, плотность и площадь $2\pi r dr$ кольца, отстоящего от опорной частицы на

расстояние r . Поскольку $gcum$ найдено для $\frac{1}{2}N$ опорных частиц, мы делим этот массив также на $\frac{1}{2}N$, избегая тем самым двойного счета.

```

SUB output (N,nmcs,gcum(),Lx,Ly,dr,bin,accept)
  ! вычисляется нормированная g(r) и печатаются результаты
  ! максимальное расстояние между частицами равно Ly/2
  LET rmax = bin*Ly/2
  LET density = N/(Lx*Ly)           ! плотность
  PRINT "плотность = "; density
  LET accept = accept/(nmcs*N)
  PRINT "коэффициент принятия = "; accept
  ! g(r) вычисляется после каждого шага Монте-Карло на частицу
  LET norm = 2.0/(density*nmcs*N)   ! нормировка для g(r)
  FOR ir = 1 to rmax
    IF gcum(ir) > 0 then
      LET r = ir*dr + 0.5*dr        ! вычисляем r в середине кольца
      LET area = 2*pi*r*dr          ! площадь кольца
      LET g = gcum(ir)*norm/area
      PRINT r,g
    END IF
  NEXT ir
END SUB

```

Последнюю конфигурацию просчитанного варианта следует сохранить в файле, с тем чтобы можно было ее использовать в качестве начальной конфигурации в следующем варианте.

```

SUB save_config(N,Lx,Ly,x(),y())
  INPUT prompt "Имя файла для последней конфигурации = ": fnm$
  OPEN #2: name fnm$, access output, create new
  PRINT #2: Lx, ",", Ly
  FOR i = 1 to N
    PRINT #2: x(i), ",", y(i)
  NEXT i
END SUB

```

В качестве проверки программы `hard_disk` рассмотрим сначала одномерную систему твердых стержней. В силу простоты этой системы уравнение состояния и $g(r)$ могут быть вычислены точно. Например, уравнение состояния имеет вид

$$\frac{P}{N k_B T} = \frac{1}{L - N\sigma}, \quad (16.32)$$

при этом плотность равна $\rho = N/L$. Поскольку твердые стержни не могут проходить друг сквозь друга, то легко видеть, что исключаемый объем составляет $N\sigma$ и, значит, доступный объем равен $L - N\sigma$. Заметим, что форма уравнения состояния (16.32) совпадает с уравнением Ван-дерВаальса (см. гл. 6) при ненулевом вкладе притягивающей части взаимодействия.

ЗАДАЧА 16.16. Моделирование твердых стержней методом Монте-Карло

а. Модифицируйте программу `hard_disk` так, чтобы она годилась для системы твердых стержней. Задайте $L = 12$ и $N = 10$. Как согласуется плотность с максимальной возможной плотностью? Выберите начальные положения так, чтобы они образовывали одномерную сетку. Чему равен коэффициент принятия для варианта $dx_{max} = 0.1$? Сколько приблизительно требуется шагов Монте-Карло на частицу для достижения равновесия? Вычислите парную корреляционную функцию $g(r)$ с $r = |x|$. Не забудьте, что r измеряется в единицах σ .

б. Постройте график $g(r)$ как функцию от r . Почему при значениях $r < 1$ функция $g(r) = 0$? Почему значения $g(r)$ для $r > L/2$ не имеют смысла? Какой физический смысл имеют пики на кривой $g(r)$? Экстраполируя полученные для $g(r)$ результаты на точку $r = 1$ и используя формулы (16.31), определите давление.

в. Получите функцию $g(r)$ для нескольких меньших значений плотности, взяв за основу какую-нибудь равновесную конфигурацию из предыдущего варианта и увеличивая L . Как меняются размеры и расположение пиков на кривой $g(r)$?

ЗАДАЧА 16.17. Моделирование твердых дисков методом Монте-Карло

а. Основная цель этой задачи заключается в получении зависимости функции $g(r)$ от плотности. Какова возможная максимальная плотность твердых дисков, иначе говоря, сколько дисков удастся упаковать в ящик площадью A ? Начните с моделирования системы, имеющей плотность чуть меньше максимальной. Возьмите число дисков

$N = 16$ и область размерами $L_x = 4.41$ и $L_y = 0.5\sqrt{3}L_x$. Вычислите приведенную плотность ρ^* и сравните ее с максимально возможной плотностью. Чему равно среднее расстояние между частицами? Выберите начальные координаты частиц так, чтобы они образовывали треугольную решетку. Для начала имеет смысл величины пробных шагов взять равными $dx_{max} = dy_{max} = 0.1$. Получается ли при таком выборе шагов коэффициент принятия равным примерно 50%? Сравните функции $g(r)$ для $\rho^* = 0.95, 0.92, 0.88, 0.85, 0.80, 0.70, 0.60$ и 0.30 . Сохраняя неизменным отношение L_x/L_y , используйте последнюю конфигурацию предыдущего варианта в качестве начальной конфигурации нового варианта с более низкой ρ^* . Подождите по крайней мере 20 шагов Монте-Карло на частицу, чтобы дать системе возможность прийти в равновесное состояние, и усредните функцию $g(r)$ для $n_{mcs} \geq 100$.

б. Каково качественное поведение функции $g(r)$ при высокой и низкой плотностях? Опишите, например, количество и высоту пиков, наблюдаемых у $g(r)$.

в. Используя соотношение (16.31б), вычислите зависимость давления от ρ^* . Изобразите график отношения $PV/Nk_B T$ как функцию от ρ^* . «Объем» равен $V = L_x L_y$. Является ли $PV/Nk_B T$ возрастающей или убывающей функцией от ρ ? Можно предположить, что при низких плотностях система ведет себя как идеальный газ с точностью до замены объема на $V - N\sigma$. Сравните результаты, полученные для самой низкой плотности, с этим прогнозом.

г. Получите «снимки» дисков в интервале от десяти до двадцати шагов Монте-Карло на частицу. Наблюдаются ли какие-нибудь признаки плавления твердой фазы в жидкую при низких плотностях?

д. Вычислите «эффективный коэффициент диффузии» D путем определения суммарного среднего квадрата смещения $\langle R(t)^2 \rangle$ частиц после того, как равновесие достигнуто. «Время» t можно отождествить с числом шагов Монте-Карло на частицу. Находя D из соотношения $D = \langle R(t)^2 \rangle / 4t$, оцените D для значений плотности, рассмотренных в п. «а». Изобразите график зависимости $\rho^ D$ от ρ^* . Как зависит D от ρ^* для разреженного газа? Попробуйте определить область значений ρ^* , где D резко падает. Наблюдаются ли какие-нибудь признаки фазового перехода?

Непрерывные потенциалы. Проведенное моделирование твердых дисков привело нас к предварительному заключению о существовании фазового перехода из жидкой фазы с низкими плотностями в твердую фазу с более высокими плотностями. Этот результат совпадает с полученным в расчетах методами молекулярной динамики и Монте-Карло с большими системами. Хотя существование перехода жидкость—твердое тело для систем твердых сфер вполне признано, не следует забывать, что в любом вычислительном эксперименте используется относительно малое количество частиц и что результаты такого рода нельзя считать доказательством независимо от какого бы то ни было теоретического обоснования.

Существование перехода жидкость—твердое тело для твердых сфер указывает на то, что этот переход обуславливается главным образом отталкивающей частью потенциала. Далее мы рассмотрим систему, в которой имеется как отталкивание, так и притяжение. Нашей основной задачей будет определение влияния притягивающего потенциала на структуру жидкости.

Примем в качестве модели потенциал межчастичного взаимодействия Леннарда—Джонса

$$V(r) = 4\epsilon \left[\left(\frac{\sigma}{r} \right)^{12} - \left(\frac{\sigma}{r} \right)^6 \right]. \quad (16.33)$$

Свойства этого потенциала были описаны в гл. 6. Для атомов аргона параметры Леннарда—Джонса равны $\sigma = 3.405 \text{ \AA}$ и $\epsilon/k_B = 119.8 \text{ K}$. Удобно ввести безразмерные температуру и давление посредством соотношений $T^* = k_B T / \epsilon$ и $P^* = P \sigma^2 / \epsilon$.

Применим теперь алгоритм Метрополиса к непрерывному потенциалу. Главное изменение, которое необходимо внести в программу `hard_disk`, состоит в замене подпрограммы `overlap` на подпрограмму `test`. В этой подпрограмме основной интерес представляет величина *petest*, которая представляет собой новую потенциальную энергию взаимодействия частицы с номером *itrial*, находящейся в пробной точке с координатами (*xtrial*, *ytrial*). Для простоты вычисление потенциальной энергии частицы *itrial* производится путем подсчета энергии взаимодействия со всеми остальными $N-1$ частицами, а не только с частицами в области действия потенциала ($\sim 2.3\sigma$). Величина *petest* сравнивается с *peold*—потенциальной энергией взаимодействия частицы *itrial* в точке (*x(itrial)*, *y(itrial)*). Отметим, что величина *pe* должна вычисляться заранее и передаваться в подпрограмму `test`.

```

SUB test(N, itrial, xtrial, ytrial, x(), y(), pe, beta, Lx, Ly, kx, ky, accept)
  ! алгоритм Метрополиса для непрерывных потенциалов
  DECLARE DEF V
  ! itrial - номер частицы, выбранной случайно в подпрограмме move
  LET petest = 0
  LET peold = 0
  FOR j = 1 to N
    IF itrial <> j then
      LET dx = x(j) - xtrial
      LET dy = y(j) - ytrial
      ! вычисляем правильное расстояние
      CALL separation(dx, dy, Lx, Ly, kx, ky)
      LET R2 = dx*dx + dy*dy
      LET r = sqr(R2)
      ! petest - пробная энергия взаимодействия частицы itrial
      ! с остальными частицами
      LET petest = petest + V(r)
    END IF
  NEXT j
  FOR j = 1 to N
    IF itrial <> j then
      LET dx = x(j) - x(itrial)
      LET dy = y(j) - y(itrial)
      CALL separation(dx, dy, Lx, Ly, kx, ky)
      LET R2 = dx*dx + dy*dy
      LET r = sqr(R2)
      LET peold = peold + V(r)
    END IF
  NEXT j
  LET de = petest - peold
  IF de > 0 then
    IF exp(-beta*de) < md then
      EXIT SUB ! пробная конфигурация не принимается
    END IF
  END IF
  LET accept = accept + 1
  LET x(itrial) = xtrial ! itrial обозначает номер частицы
  LET y(itrial) = ytrial
  LET pe = pe + de
END SUB

```

ЗАДАЧА 16.18. Моделирование простых жидкостей и твердых тел методом Монте-Карло

а. Рассмотрим простую модель системы многих частиц, взаимодействующих с потенциалом (16.33). Такую модель часто называют леннард—джонсовской, чтобы отличать ее от более реалистичных моделей жидкостей. Модифицируйте программу `hard_disk`, чтобы можно было вычислять среднюю энергию, давление и двумерную леннард—джонсовскую парную корреляционную функцию с помощью алгоритма Метрополиса. (Само собой, напишите подпрограмму вычисления начальной потенциальной энергии взаимодействия.) Для простоты вычисляйте средние после каждого шага Монте-Карло на частицу. Используйте безразмерные единицы $E^* = E/\epsilon$, $T^* = k_B T/\epsilon$ и $\rho^* = \rho/\sigma^2$. Какое будет при этом определение P^* ? (Не забудьте, что для идеального газа $PV = Nk_B T$.)

б. Прежде чем применять алгоритм Метрополиса при $T^* \neq 0$, нужно найти E_0^* —энергию основного состояния на частицу при $T^* = 0$. Чтобы выполнить это вычисление по программе, разработанной вами в п. «а», поместите перед инструкциями, содержащими переменную `beta`, признаки комментария (!). Тогда перемещения будут приниматься только в том случае, если потенциальная энергия убывает. Возьмите $N = 16$, $L_x = 4.5$, $L_y = 6.0 \cdot \sqrt{3}/2$ и разместите частицы на треугольной решетке. Поскольку данная конфигурация частиц близка к равновесной, нет необходимости усреднять E_0^* более чем по десяти шагам Монте-Карло на частицу.

в. Используйте те же начальные условия, что в п. «б», но примите $T^* = 1.5$. Возьмите $dx_{max} = dy_{max} = 0.15$, $nmcs \geq 100$ и $dr = 0.1$. Вычислите среднюю энергию на частицу E^* . Повторите моделирование для $T^* = 2.5$ и $T^* = 3.5$. Как предсказывает гармоническая теория твердого тела, полная энергия системы определяется вкладом члена с $T = 0$ и члена, отвечающего гармоническим колебаниям атомов. Вклад этой последней части должен быть пропорционален температуре. Сравните свои результаты для температурной зависимости E^* с этим предсказанием.

г. Опишите качественный вид $g(r)$ для твердого тела Леннарда—Джонса. Сопоставьте это с вашими результатами для $g(r)$, полученными для твердых дисков. Для определения среднего давления воспользуйтесь формулой (16.29).

д. Чему равны соответствующие температура, энергия и давление в единицах CGS в проведенных выше вычислительных экспериментах для твердого аргона?

е. Уменьшите плотность, умножая L_x , L_y и все координаты частиц на 1.5. Оцените число шагов Монте-Карло на частицу, необходимое для получения P^* и E^* при $T^* = 3.5$ с точностью 10%. Сравните P^* и E^* с соответствующими значениями для идеального газа. Положительна ли полная энергия или отрицательна? Следуя методу, рассмотренному в задаче 16.17, вычислите эффективный коэффициент диффузии. Является ли система жидкостью или твердым телом? Постройте график функции $g(r)$ в зависимости от r/σ и сравните $g(r)$ с результатами, полученными для твердых дисков при той же плотности. Моделирование систем бóльшего размера позволяет вычислить $g(r)$ для бóльших значений r . Если позволяет время, рассмотрите $N = 36$ и вычислите $g(r)$ при той же плотности и температуре. Как качественно ведет себя $g(r)$? Как интерпретировать пики на $g(r)$ с точки зрения структуры жидкости?

ж. Вычислите среднюю энергию, давление и функцию $g(r)$ для $L_x = 10$, $L_y = 10$, $dr = 0.1$, $dx_{max} = dy_{max} = 1.0$, $nmcs \geq 100$ и $T^* = 3.0$. Эти условия отвечают разреженному газу. Как ваши результаты для давления соотносятся с результатом для идеального газа? Как $g(r)$ соотносится с результатами, полученными вами для жидкости?

*ЗАДАЧА 16.19. Обратный степенной закон взаимодействия

Рассмотрим потенциал, обратно пропорциональный степени расстояния

$$V(r) = V_0(\sigma/r)^n. \quad (16.34)$$

Система твердых сфер представляет собой частный случай (16.34) при $n \rightarrow \infty$. Какие фазы, по вашему мнению, будут наблюдаться при произвольных n ? Сравните качественные особенности $g(r)$ для «мягкого» потенциала, скажем с $n = 4$, со случаем твердых дисков при одинаковой плотности.

16.8. ДРУГИЕ ПРИЛОЖЕНИЯ

По-видимому, нет надобности убеждать вас, что методы Монте-Карло отличаются мощностью, гибкостью и применимы для самых разнообразных систем. Обобщения методов Монте-Карло, которые мы еще не рассматривали, включают в себя многочастичные движения, вынужденные движения, где частицы стремятся двигаться в направлении действующей на них силы, операции с битами для моделей типа Изинга, использование специальных процессоров для конкретных систем и применение матричных процессоров, которые позволяют изменять одновременно разные части большой системы.

Другим приложением, которого мы не касались, является моделирование систем с дальнедействующими потенциалами, таких, как кулоновские системы. Для таких потенциалов необходимо разрабатывать методы учета взаимодействий частиц в центре ячейки с бесконечной системой периодических копий. Одно из наиболее свежих впечатляющих приложений метода Монте-Карло связано с анализом поведения теорий калибровки на решетках—теорий, на которые возлагаются большие надежды в деле понимания фундаментальных взаимодействий материи.

Данную главу мы завершаем обсуждением методов Монте-Карло применительно к задачам, которые, как могло бы показаться, имеют мало общего с типами задач, которые мы рассмотрели. Эти задачи называются *многомерной* или *комбинаторной* оптимизацией—модный способ сказать: «Как вы находите минимум функции, зависящей от многих параметров?». Мы объясняем сущность этого типа задач на примере, известном как *задача о коммивояжере*, которую мы будем называть *задачей о коробейнике*.

Задачу о коробейнике можно сформулировать следующим образом. Предположим, что некий коробейник хочет посетить N городов и следовать таким маршрутом, чтобы ни один город не посещать более одного раза, закончить поездку в том же пункте, откуда он ее начал, и сократить общее пройденное расстояние до минимума. Какой маршрут является оптимальным? На рис. 16.6 показан пример N городов и один из возможных маршрутов. Задачи такого типа возникают во всех областях планирования и проектирования. Во всех известных точных методах определения оптимального пути требуемое машинное время растет как e^N . Поэтому на практике точное решение можно находить только для задач, насчитывающих несколько сотен городов или меньше. Задача о коробейнике принадлежит к большому классу задач, называемых *NP-полнота*—этим термином вы можете воспользоваться, чтобы произвести впечатле-

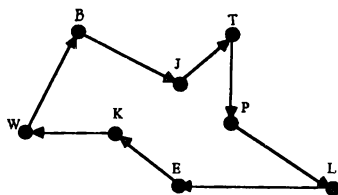


Рис. 16.6. Какой маршрут является оптимальным для данного случайного расположения $N = 8$ узлов? Маршрут начинается и заканчивается в узле W. Показан один из возможных маршрутов.

ние на своих приятелей. С помощью карандаша и бумаги отыщите оптимальный путь для случайного размещения восьми городов. Каково, по вашей оценке, предельное количество городов, которое целесообразно рассматривать, не прибегая к компьютеру?

Чтобы понять характер различных подходов к задаче о коробейнике, рассмотрим изображенную на рис. 16.7 «энергетическую» функцию $E(a)$. Мы можем связать $E(a)$ с длиной пути и интерпретировать a как параметр, отражающий порядок посещения городов. В общем случае функция $E(a)$ имеет несколько локальных минимумов и один абсолютный минимум. Имеется ли какой-нибудь хороший метод для отыскания абсолютного минимума функции $E(a)$? Один путь — это точный расчет, т.е. изменять a и находить значение E везде. Этот метод соответствовал бы определению длины пути коробейника для каждого возможного маршрута, и очевидно, что такая задача непосильна, если число городов велико. Следовательно, нужно использовать какой-то *эвристический метод*, т.е. приближенный метод отыскания пути, близкого к абсолютному минимуму. Один из методов состоит в том, что выбирается некое значение a , ге-

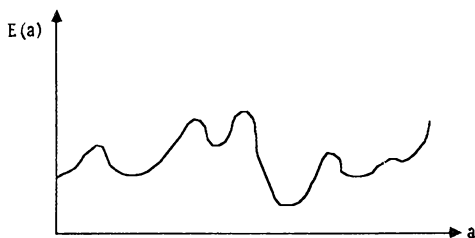


Рис. 16.7. График зависимости функции $E(a)$ от параметра a .

нерируется небольшое случайное число δa и данное изменение принимается, если $E(a + \delta a)$ не превышает $E(a)$. Такая стратегия итерационного уточнения соответствует поиску шагов, которые ведут к спуску. Поскольку, как правило, этот поиск застревает в локальном, а не глобальном минимуме, обычно начинают с нескольких начальных вариантов a и оставляют самый лучший результат. Как можно было бы применить этот метод к задаче о коробейнике?

Рассмотрим теперь, казалось бы, совсем иную задачу. Предположим, что мы хотим изготовить идеальный монокристалл. Возможно, вам известно, что для этого необходимо сначала расплавить вещество, а затем очень медленно охлаждать его до требуемой низкой температуры. Если понижать температуру слишком быстро (быстрое «закачивание»), образующийся кристалл будет обладать многочисленными дефектами или же вовсе не вырастет. Постепенное понижение температуры называют *отжигом*.

Каким образом можно применить метод отжига к задаче отыскания минимума $E(a)$? Пусть выбрано некоторое значение a , сгенерировано небольшое случайное число δa и вычислено $E(a + \delta a)$. Если $E(a + \delta a)$ меньше или равно $E(a)$, данное изменение, как и ранее, принимается. Однако если $\Delta E = E(a + \delta a) - E(a) > 0$, данное изменение принимается с вероятностью $P = \exp(-\Delta E/k_B T)$, где T — эффективная температура. Эта процедура является хорошо знакомым алгоритмом Метрополиса, причем температура играет здесь роль управляющего параметра. Процесс *вынужденного отжига* состоит из начального «расплавления» системы с последующим постепенным понижением температуры. При каждой температуре моделирование должно длиться достаточно долго, чтобы система достигала устойчивого состояния. Режим отжига, т.е. скорость понижения температуры, определяет качество решения.

Смысл метода модельного отжига понятен — чтобы попасть в долину, иногда нужно взбираться на горку. Первым применением метода модельного отжига было оптимальное проектирование компьютеров. Пожалуй, вам стоит подумать и о других приложениях. В задаче 16.20 этот метод используется в задаче о коробейнике.

*ЗАДАЧА 16.20. Модельный отжиг и задача о коробейнике

Сгенерируйте случайное расположение восьми городов в квадрате со стороной $10^{1/2}$ и выполните трудоемкий расчет по отысканию оптимального пути. Затем составьте программу для решения этой задачи

методом модельного отжига. Например, заведите массив для хранения координат каждого города и массив для хранения расстояний между ними. Состояние системы, т.е. путь, представляющий последовательность городов, можно хранить в другом массиве. Длина этого пути связывается с энергией воображаемой тепловой системы. Из какого критерия целесообразно исходить при выборе начальной температуры системы? Как можно генерировать случайные перегруппировки пути? Один из методов — выбрать случайно два города и поменять местами очередность посещений. Возьмите этот метод или иной, который вы придумаете, и найдите подходящий режим отжига. Во всех случаях, когда представится возможным, сравните результаты своего отжига с точными результатами. Распространите свои результаты на большие значения N , например $N = 12, 24$ и 48 . Для каждого данного режима отжига определите вероятность отыскания пути данной длины.

ЛИТЕРАТУРА

K. Binder, ed., *Monte Carlo Methods in Statistical Physics*, 2-nd edition, Springer-Verlag, 1986. См. также *K. Binder*, ed., *Applications of the Monte Carlo Method in Statistical Physics*, Springer-Verlag, 1984.

M. Bishop, *C. Bruin*, The pair correlation function: a probe of molecular order, *Am. J. Phys.* **52**, 1106 (1984). В статье вычисляется парная корреляционная функция для двумерной модели Леннарда — Джонса.

J. Kertsz, *J. Cserti*, *J. Szp*, Monte Carlo simulation programs for micro-computer, *Eur. J. Phys.* **6**, 232 (1985).

C. Kittel, *Introduction to Solid State Physics*, 6th ed., John Wiley & Sons, 1986. [Имеется перевод 4-го изд.: *Киттель Ч.*, Введение в физику твердого тела. — М.: Наука, 1978.] В гл. 16 этого классического учебника рассматриваются ферромагнетизм и антиферромагнетизм.

S. Kirkpatrick, *C. D. Gelatt*, *M. P. Vecchi*, Optimization by simulated annealing, *Science* **220**, 671, (1983). См. также *S. Kirkpatrick*, *G. Toulouse*, Configuration space analysis of traveling salesman problems, *J. Physique* **46**, 1277 (1985).

D. P. Landau, Finite-size behavior of the Ising square lattice, *Phys. Rev.* **B13**, 2997 (1976). Ясно написанная работа о конечномерном масштабном анализе данных Монте-Карло. См. также *D. P. Landau*, Finite-size behavior of the simple-cubic Ising lattice, *Phys. Rev.* **B14**, 255 (1976).

D. P. Landau, R. Alben, Monte Carlo calculations as an aid in teaching statistical mechanics, *Am. J. Phys.* **41**, 394 (1973).

N. Metropolis, A. W. Rosenbluth, M. N. Rosenbluth, A. H. Teller, E. Keller, Equation of state calculations for fast computing machines, *J. Chem. Phys.* **6**, 1087 (1953).

J. Marro, R. Toral, Microscopic observations on a kinetic Ising model, *Am. J. Phys.* **54**, 1114 (1986).

O. G. Mouritsen, Computer Studies of Phase Transitions and Critical Phenomena, Springer-Verlag, 1984.

H. E. Stanley, Introduction to Phase Transitions and Critical Phenomena, Oxford University Press, 1971. См. приложение В, где четко рассмотрено точное решение модели Изинга с нулевым магнитным полем для двумерной решетки.

J. P. Valleau, S. G. Whittington, A guide to Monte Carlo for statistical mechanics: 1. Highways, in: Statistical Mechanics, Part A, *B. J. Berne*, ed., Plenum Press, 1977. См. также *J. P. Valleau, G. M. Torrie*, A guide to Monte Carlo for statistical mechanics: 2. Byways, там же.

ДОПОЛНИТЕЛЬНАЯ ЛИТЕРАТУРА

Матвеев А. Н., Молекулярная физика. — М.: Высшая школа, 1981. Рассматривается понятие канонического ансамбля.

Хуанг К., Статистическая механика. — М.: Мир, 1966. В этой книге подробно обсуждается понятие канонического ансамбля.

Дайсон Ф., Монролл Э., Кац М., Фишер М., Устойчивость и фазовые переходы. — М.: Мир, 1973. В лекции Э. Монролла обсуждается устойчивость двумерной модели Изинга.

Климонтович Ю. Л., Статистическая физика. — М.: Наука, 1982. Излагается модель твердых сфер (дисков).

Вильсон К., Козул Дж., Ренормализационная группа и ϵ -разложения. — М.: Мир, Москва, 1975. Рассматривается перенормировка в модели Изинга. Книга рассчитана на подготовленного читателя.

ПРИЛОЖЕНИЕ 16А. ФЛУКТУАЦИИ В КАНОНИЧЕСКОМ АНСАМБЛЕ

Получим сначала соотношение, связывающее удельную теплоемкость при постоянном объеме с энергией флуктуаций в каноническом ансамбле. Для простоты примем обозначение $U = \langle E \rangle$. Из определения удельной теплоемкости (16.14а) имеем

$$C = \frac{\partial U}{\partial T} = - \frac{1}{k_B T^2} \frac{\partial U}{\partial \beta}. \quad (16.35)$$

Из (16.3)

$$U = - \frac{\partial}{\partial \beta} \ln z \quad (16.36)$$

и

$$\begin{aligned} \frac{\partial U}{\partial \beta} &= - \frac{1}{Z^2} \frac{\partial Z}{\partial \beta} \sum_s E_s \exp(-\beta E_s) - \frac{1}{Z} \sum_s E_s^2 \exp(-\beta E_s) = \\ &= \langle E \rangle^2 - \langle E^2 \rangle. \end{aligned} \quad (16.37)$$

Соотношение (16.14б) вытекает из (16.35) и (16.37). Обратите внимание на то, что речь идет об удельной теплоемкости при постоянном объеме, коль скоро частные производные брались при условии, что энергетические уровни E_s остаются постоянными.

Связь магнитной восприимчивости с флуктуациями намагниченности можно получить аналогичным образом. Предположим, что энергия может быть записана в виде

$$E_s = E_{0,s} - H M_s, \quad (16.38)$$

где $E_{0,s}$ — энергия в отсутствие магнитного поля, H — наложенное внешнее поле и M_s — намагниченность в s -состоянии. Средняя намагниченность равна:

$$\langle M \rangle = \frac{1}{Z} \sum_s M_s \exp(-\beta E_s). \quad (16.39)$$

Поскольку $\partial E_s / \partial H = -M_s$, имеем

$$\frac{\partial Z}{\partial H} = \sum_s \beta M_s \exp(-\beta E_s). \quad (16.40)$$

Отсюда получаем

$$\langle M \rangle = \frac{1}{\beta} \frac{\partial}{\partial H} \ln z. \quad (16.41)$$

Используя (16.39) и (16.41), найдем

$$\begin{aligned} \frac{\partial \langle M \rangle}{\partial H} &= -\frac{1}{Z^2} \frac{\partial Z}{\partial H} \sum_s M_s \exp(-\beta E_s) + \frac{1}{Z} \sum_s \beta M_s^2 \exp(-\beta E_s) = \\ &= \beta \langle M \rangle^2 + \beta \langle M^2 \rangle. \end{aligned} \quad (16.42)$$

Соотношение (16.156) для восприимчивости при нулевом поле вытекает из (16.42) и определения

$$\chi = \lim_{H \rightarrow 0} \frac{\partial \langle M \rangle}{\partial H}. \quad (16.43)$$

ПРИЛОЖЕНИЕ 16Б. ТОЧНЫЙ РАСЧЕТ МОДЕЛИ ИЗИНГА ДЛЯ РЕШЕТКИ 2×2

Поскольку число возможных состояний, или конфигураций, модели Изинга растет как 2^N , возможные конфигурации удастся просчитать только для малых N . В качестве примера вычислим различные представляющие интерес величины для модели Изинга на квадратной решетке 2×2 . На рис. 16.8 изображены две различные конфигурации, у которых два спина ориентированы вверх. В табл. 16.1 мы группируем состояния в соответствии с их полной энергией и намагниченностью. Теперь пользуясь табл. 16.1, можно вычислять все интересующие нас величины.

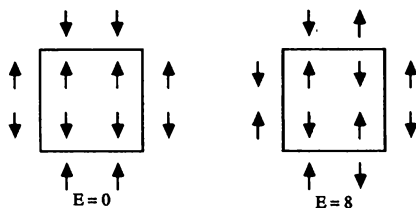


Рис. 16.8. Примеры конфигураций в модели Изинга на квадратной решетке 2×2 . Обратите внимание на использование периодических краевых условий.

ТАБЛИЦА 16.1. Энергия и намагненность 2^4 состояний модели Изинга для нулевого магнитного поля на квадратной решетке 2×2

Число спинов вверх	Вырожденность	Энергия	Намагненность
4	1	-8	4
3	4	0	2
2	4	0	0
2	2	8	0
1	4	0	-2
0	1	-8	-4

Сумма по состояниям имеет вид

$$Z = 2e^{8\beta J} + 12 + 2e^{-8\beta J}. \quad (16.44)$$

Используя соотношения (16.36) и (16.44), найдем

$$U = -\frac{1}{Z} [2(8)e^{8\beta J} + 2(-8)e^{-8\beta J}]. \quad (16.45)$$

Поскольку остальные рассматриваемые величины можно найти аналогичным образом, приведем только окончательные результаты:

$$\langle E^2 \rangle = \frac{1}{Z} [2(64)e^{8\beta J} + 2(64)e^{-8\beta J}], \quad (16.46)$$

$$\langle M \rangle = \frac{1}{Z} (0) = 0, \quad (16.47)$$

$$\langle |M| \rangle = \frac{1}{Z} [2(4)e^{8\beta J} + 8(2)], \quad (16.48)$$

$$\langle M^2 \rangle = \frac{1}{Z} [2(16)e^{8\beta J} + 8(4)]. \quad (16.49)$$

Зависимость C и χ от βJ может быть найдена с помощью соотношений (16.45), (16.46) и (16.47), (16.49) соответственно.

КВАНТОВЫЕ СИСТЕМЫ

17

В этой главе изучаются численные решения стационарного уравнения Шредингера, полученные методом Эйлера—Кромера. Движение волнового пакета образуется из решений для стационарных состояний с использованием принципа суперпозиции. Затем описываются метод случайного блуждания для стационарного уравнения Шредингера и вариационный метод Монте-Карло для вычисления характеристик основного состояния.

17.1. ВВЕДЕНИЕ

До сих пор мы моделировали поведение физических систем с помощью методов Монте-Карло и молекулярной динамики. В методе молекулярной динамики рассчитывается зависимость от времени классической траектории (координата и импульс) каждой частицы. Однако для квантовых систем нельзя пользоваться методами молекулярной динамики, так как координата и импульс частицы не могут быть точно определены одновременно. Поскольку фундаментальное описание природы по сути является квантовомеханическим, то возникает проблема, которая заключается в том, что невозможно в конечном счете промоделировать природу на компьютере.

Конечно, квантовая механика позволяет нам *анализировать* вероятности. Чтобы понять трудности, связанные с таким анализом, рассмотрим сначала простую вероятностную систему, описываемую одномерным уравнением диффузии (см. гл. 11)

$$\frac{\partial P(x, t)}{\partial t} = D \frac{\partial^2 P(x, t)}{\partial x^2}, \quad (17.1)$$

где $P(x, t)$ — плотность вероятности того, что в момент времени t частица находится в точке с координатой x . Один из способов вычисления $P(x, t)$ основан на дискретизации переменных x и t . Предположим, размер сетки по x выбирается из тех соображений, чтобы получить вероятность для m значений переменной x . Если выбрать значение m порядка 10^3 , то понятно, что для прямого вычисления $P(x, t)$ потребуется приблизительно 10^3 точек для каждого значения t . По сравнению с этим, для соответствующего расчета методом молекулярной динамики, использующим второй закон Ньютона, потребовалась бы одна точка.

Трудоемкость метода прямого вычисления становится еще более очевидной, если у системы имеется много степеней свободы. Например, если имеется N -частичная одномерная система, то нам нужно вычислять вероятность $P(x_1, x_2, \dots, x_N, t)$, где x_i — координата i -й частицы. Поскольку для каждой координаты x_i необходимо выбрать m -точечную сетку, то в каждый момент времени t требуется точно определять N^m конфигураций. Обычно значение m выбирается того же порядка, что N , поскольку бывает полезно знать вероятности в каждой точке пространства. В результате для получения требуемой вероятности в каждый момент времени необходимо рассчитывать порядка N^N конфигураций. Следовательно, удвоение размеров системы $N \rightarrow 2N$ приводит к экспонен-

циальному росту машинного времени и объема требуемой памяти.

Хотя применение метода прямого расчета ограничивается системами с небольшим числом степеней свободы, простота этого метода поможет нам понять поведение одномерных квантовых систем. После краткого изложения общих свойств квантовомеханических систем в разд. 17.2 мы рассмотрим этот метод в разд. 17.3 применительно к стационарному уравнению Шредингера. В разд. 17.4 мы используем решения для стационарных состояний и принцип суперпозиции для построения решений нестационарного уравнения Шредингера в виде волновых пакетов.

Существуют ли другие методы описания систем, имеющих вероятностную природу? Поскольку мы уже познакомились с тем, как уравнение (17.1) можно сформулировать в виде задачи случайного блуждания, вас не должно удивлять то, что уравнение Шредингера можно исследовать аналогичным методом. В разд. 17.5 мы познакомимся с методами Монте-Карло для квантовых систем. В разд. 17.6 также используются методы Монте-Карло для получения вариационных решений основного состояния.

17.2. ОБЗОР КВАНТОВОЙ ТЕОРИИ

Для простоты рассмотрим одномерные нерелятивистские квантовые системы, состоящие из одной частицы. Состояние системы полностью описывается *волновой функцией* $\Psi(x, t)$, которая интерпретируется как *амплитуда вероятности*. Поскольку частица может находиться в любой точке пространства, то $P(x, t)dx$ — вероятность того, что частица находится в элементе «объема» dx с центром в точке x в момент времени t , — равна

$$P(x, t)dx = C |\Psi(x, t)|^2 dx, \quad (17.2)$$

где C — нормировочная постоянная. Вероятностная интерпретация $\Psi(x, t)$ означает, что удобно использовать нормированные волновые функции, удовлетворяющие условию

$$\int_{-\infty}^{\infty} dx \Psi^*(x, t) \Psi(x, t) = 1, \quad (17.3)$$

где $\Psi^*(x, t)$ — функция комплексно-сопряженная $\Psi(x, t)$. Тогда постоянная C в выражении (17.2) равна 1.

Если частица движется в потенциале $V(x, t)$, то временная эволюция функции $\Psi(x, t)$ описывается нестационарным уравнением Шредингера

$$i\hbar \frac{\partial \Psi(x, t)}{\partial t} = -\frac{\hbar^2}{2m} \frac{\partial^2 \Psi(x, t)}{\partial x^2} + V(x, t) \Psi(x, t), \quad (17.4)$$

где m — масса частицы, а \hbar — постоянная Планка, поделенная на 2π .

Физические величины, такие, как импульс, можно представить операторами. Математическое ожидание, или среднее значение наблюдаемой величины A определяется выражением

$$\langle A \rangle = \int dx \Psi^*(x, t) A_{op} \Psi(x, t), \quad (17.5)$$

где A_{op} — оператор, соответствующий величине A . Например, оператор, соответствующий импульсу p , имеет вид $p_{op} = -i\hbar \partial / \partial x$.

Если потенциал не зависит от времени, то для уравнения (17.4) можно получить решения вида

$$\Psi(x, t) = \phi(x) \exp(-iEt/\hbar). \quad (17.6)$$

Частица, находящаяся в состоянии (17.6), имеет вполне конкретное значение энергии E . Если подставить выражение (17.6) в (17.4), то получим стационарное уравнение Шредингера

$$-\frac{\hbar^2}{2m} \frac{\partial^2 \phi(x)}{\partial x^2} + V(x)\phi(x) = E_n \phi(x). \quad (17.7)$$

Заметим, что $\phi(x)$ — *собственная функция* оператора Гамильтона (гамильтониана)

$$H_{op} = -\frac{\hbar^2}{2m} \frac{\partial^2}{\partial x^2} + V(x), \quad (17.8)$$

соответствующая *собственному значению* E , т.е.

$$H_{op} \phi(x) = E \phi(x). \quad (17.9)$$

Чтобы различать возможные значения энергии E , будем отмечать состояния ϕ индексом n .

Общее решение $\Psi(x, t)$ можно выразить в виде суперпозиции собственных функций оператора, отвечающего той или иной физической наблюдаемой величине. Например, если H не зависит от времени, то можно записать

$$\Psi(x, t) = \sum_n c_n \phi_n(x) \exp(-iEt/\hbar), \quad (17.10)$$

где ϕ_n — собственные функции оператора H , а знак \sum обозначает сумму по всем дискретным состояниям и интеграл по непрерывному спектру. Коэффициенты c_n в формуле (17.10) можно определить из значения $\Psi(x, t)$ в любой момент времени t . Например, если нам известна $\Psi(x, t)$ при $t = 0$, то можно воспользоваться свойством ортогональности собственных функций любого физического оператора и получить

$$c_n = \int dx \phi_n^*(x) \Psi(x, 0). \quad (17.11)$$

Коэффициент c_n можно интерпретировать как амплитуду вероятности измерения полной энергии, при котором получается значение E_n .

17.3. СТАЦИОНАРНОЕ УРАВНЕНИЕ ШРЕДИНГЕРА

Рассмотрим решения стационарного уравнения Шредингера (17.7), соответствующие связанным состояниям. Основным наш результат будет заключаться в том, что допустимые решения уравнения (17.7) существуют только тогда, когда собственные значения квантованы, т.е. ограничены дискретным набором энергий. Чтобы решение было допустимым, функции $\phi_n(x)$ должны быть конечны для всех значений x и ограничены для больших значений $|x|$ так, чтобы функцию $\phi_n(x)$ можно было нормировать. Для конечной функции $V(x)$ требуется, чтобы функции $\phi_n(x)$ и $\phi'_n(x) = d\phi_n(x)/dx$ были непрерывны, конечны и однозначны для всех x .

Поскольку стационарное уравнение Шредингера является дифференциальным уравнением второго порядка, то для получения единственного решения необходимо, вообще говоря, задать два краевых условия. Для упрощения анализа рассмотрим симметричные потенциалы, удовлетворяющие условию

$$V(x) = V(-x). \quad (17.12)$$

Как следует из условия (17.12), можно считать, что функции $\phi(x)$ обладают определенной четностью. Для четных решений $\phi(x) = \phi(-x)$; для нечетных решений $\phi(-x) = -\phi(x)$. Определенная четность $\phi(x)$ позволяет задать либо ϕ , либо ϕ' при $x = 0$.

Чтобы был понятен наш выбор подходящего алгоритма численного решения уравнения (17.7), напомним, что решение (17.7) с $V(x) = 0$ можно представить в виде линейной комбинации косинусов и синусов. Колебательный характер этого решения позволяет надеяться, что алгоритм Эйлера—Кромера, рассмотренный в гл. 3, будет давать удовлетворительные результаты и в случае $V(x) \neq 0$. Алгоритм Эйлера—Кромера реализуется следующим образом:

1. Разбиваем область изменения x на N отрезков длиной Δx . Введем следующие обозначения: $x_r = r\Delta x$, $\phi_r = \phi(x_r)$ и $\phi'_r = \phi'(x_r)$.
2. Задаем четность функции $\phi(x)$. Для четного решения выбираем $\phi(0) = 1$ и $\phi' = 0$; для нечетного выбираем $\phi(0) = 0$ и $\phi' = 1$. Ненулевые значения $\phi(0)$ или ϕ' произвольны.
3. Задаем начальное приближение для E .
4. Вычисляем ϕ'_{r+1} и ϕ_{r+1} , используя алгоритм:

$$\phi'_{r+1} = \phi'_r + \phi''_r \Delta x, \quad (17.13a)$$

$$\phi_{r+1} = \phi_r + \phi'_r \Delta x. \quad (17.13b)$$

5. Проводим итерации $\phi(x)$ по возрастанию x до тех пор, пока $\phi(x)$ не начнет расходиться.
6. Изменяем величину E и повторяем шаги (2)—(4). Окаймляем значение E , изменяя его до тех пор, пока при значении E чуть меньше текущего ϕ не будет расходиться в одном направлении, а при значении E чуть больше — в противоположном направлении.

Данная процедура реализована в программе **eigen** для прямоугольной потенциальной ямы, описываемой формулой

$$V(x) = \begin{cases} 0 & \text{при } |x| \leq a, \\ V_0 & \text{при } |x| > a. \end{cases} \quad (17.14)$$

В качестве входных параметров задаются V_0 и a , четность собственной функции, предполагаемое значение энергии E , величина шага Δx и x_{\max} — максимальное значение x , отражаемое на графике.

```

PROGRAM eigen
CALL parameters(V0,a,xmax,dx)
CALL plot_potential(V0,a,xmax)
CALL Euler(V0,a,dx,xmax)
END

SUB parameters(V0,a,xmax,dx)
  INPUT prompt "глубина ямы = ": V0
  INPUT prompt "полуширина ямы = ": a
  INPUT prompt "размер шага dx = ": dx
  INPUT prompt "максимальное выводимое на график значение x = ": xmax
END SUB

SUB plot_potential(V0,a,xmax)
  SET window -xmax,xmax,-10,10
  PLOT LINES: -xmax,-6; -a,-6; -a,-9; a,-9; a,-6; xmax,-6
END SUB

SUB Euler(V0,a,dx,xmax)
  DECLARE DEF V
  INPUT prompt "четная или нечетная функция (1 или -1) ": parity
  DO
    INPUT prompt "E = ": E
    IF E = 0 then EXIT SUB
    IF parity = -1 then
      LET phi = 0           ! начальные значения при x = 0
      LET dphi = 1
    ELSE
      LET phi = 1
      LET dphi = 0
    END IF
    LET x = 0
    DO                     ! вычисление волновой функции
      LET x = x + dx
      LET d2phi = 2*(V(x,V0,a) - E)*phi      ! безразмерные единицы
      LET dphi = dphi + d2phi*dx
      LET phi = phi + dphi*dx
      PLOT POINTS: x,phi; -x,phi*parity      ! график волновой функции
    LOOP until x > xmax or abs(phi) > 10
  LOOP
END SUB

```

```

DEF V(x, V0, a)                                ! потенциал V(x)
  IF abs(x) > a then
    LET V = V0
  ELSE
    LET V = 0
  END IF
END DEF

```

ЗАДАЧА 17.1. Бесконечная прямоугольная потенциальная яма

а. Рассмотрите бесконечную прямоугольную потенциальную яму, или «задачу о частице в ящике», с потенциалом $V(x)$ в форме (17.14) при $V_0 \rightarrow \infty$. Покажите аналитически, что собственные значения энергии равны $E_n = n^2 \pi^2 \hbar^2 / 8ma^2$, где n — натуральное число. Покажите также, что нормированные собственные функции имеют вид

$$\phi_n(x) = \frac{1}{\sqrt{a}} \cos \frac{n\pi x}{a} \quad (\text{четные функции, } n = 1, 3, \dots), \quad (17.15a)$$

$$\phi_n(x) = \frac{1}{\sqrt{a}} \sin \frac{n\pi x}{a} \quad (\text{нечетные функции, } n = 2, 4, \dots). \quad (17.15b)$$

Какова четность основного состояния? Почему волновая функция основного состояния не имеет ни одного узла?

б. Положите $a = 1$ и $V_0 = 150$ и с помощью программы `eigen` найдите энергию основного состояния E_1 частицы в бесконечной прямоугольной потенциальной яме. Во всех задачах выбрана такая система единиц, что $m = \hbar = 1$. Положите $\Delta x = 0.01$ и $x_{\max} = 5$; количество отрезков N равно $N = x_{\max}/\Delta x$. Попробуйте начальные приближения энергии $E = 1.5$ и $E = 1.0$. Как узнать, что энергия основного состояния лежит между этими значениями E ? Сколько вам необходимо попыток для получения E_1 с двумя десятичными знаками?

в. Влияет ли выбор Δx на численное значение E_1 ? Определите величину шага Δx , при котором E_1 получается с тремя десятичными знаками. Достаточно ли велико значение V_0 , чтобы значение E_1 можно было считать приблизительно равным энергии основного состояния бесконечной прямоугольной потенциальной ямы? Зависит ли полученный результат для E от величины ϕ при $x = 0$?

г. Напишите программу нормировки функции ϕ . Чему равно значение

ϕ при $x = 0$? Сравните полученную численно волновую функцию основного состояния с аналитическим решением.

д. Если принять во внимание число узлов (нулей) функции ϕ , то можно найти собственные функции и собственные значения возбужденных состояний, применив метод, аналогичный использованному для основного состояния. Например, первому возбужденному состоянию отвечает один узел и квантовое число $n = 2$. Модифицируйте программу `eigen` так, чтобы в ней вычислялось число узлов, и найдите первые два возбужденных состояния бесконечной прямоугольной потенциальной ямы.

е. Предположим, вы хотите найти собственное значение, соответствующее десятому возбужденному состоянию частицы в потенциальной яме. Можно ли использовать то же значение V_0 , что и в п. «б»?

Заметим, что невозможно получить численное решение для функции ϕ , которое не расходилось бы при достаточно больших значениях x . Иначе говоря, поскольку ϕ можно вычислить только с конечной точностью, вычисляемая функция ϕ всегда будет расходиться, если выполнить достаточно много шагов. Однако всегда можно вычислить ϕ с требуемой точностью, выбрав надлежащий алгоритм и достаточно малый шаг.

ЗАДАЧА 17.2. Влияние малого возмущения на основное состояние бесконечной прямоугольной потенциальной ямы

а. Определите влияние малого возмущения на собственные состояния и собственные значения бесконечной прямоугольной потенциальной ямы. Поместите маленький прямоугольный горб с полушириной b и высотой V_b симметрично относительно точки $x = 0$ (рис. 17.1). Вы-

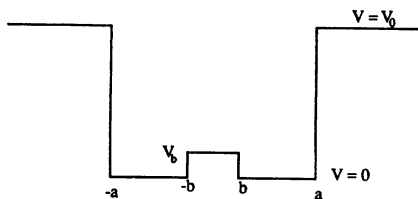


Рис. 17.1. Прямоугольная потенциальная яма с горбом высотой V в середине.

берите $V_b \ll V_0$ и $b \ll a$ и определите, как изменяются энергия основного состояния и собственная функция в зависимости от V_b и b . Каково относительное изменение E_1 для $V_b = 10$, $b = 0.1$ и $V_b = 20$, $b = 0.1$? (Положите $V_0 = 150$ и $a = 1$.) Пусть ϕ_b обозначает собственную функцию основного состояния при $b \neq 0$, а ϕ_0 — при $b = 0$. Каково относительное изменение величины

$$\int_0^a dx \phi_b(x) \phi_0(x). \quad (17.16)$$

Как связано это изменение с относительным изменением энергии?

б. Вычислите энергию основного состояния для $V_b = 20$ и $b = 1$. Как соотносится это значение E_1 с найденным в п. «а» для $V_b = 10$ и $b = 1$?

ЗАДАЧА 17.3. Конечная прямоугольная потенциальная яма

а. Рассмотрите прямоугольную потенциальную яму конечной глубины с $V_0 = 10$ и $a = 1$. Вычислите собственную функцию и собственное значение основного состояния, определяя такое значение E , при котором функция $\phi(x)$ не имеет ни одного узла и приблизительно равна нулю для больших значений x . На рис. 17.2 показана зависимость ϕ от x для двух различных значений энергии E . Заметим, что ϕ стремится к $+\infty$ при $E = 0.77$ и к $-\infty$ при $E = 0.87$.

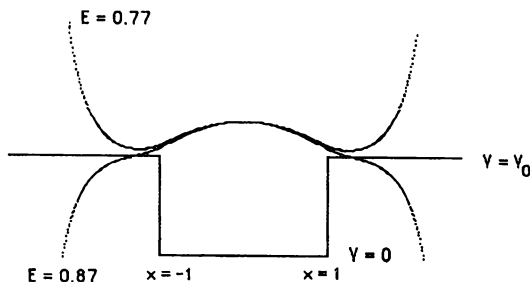


Рис. 17.2. Два численных решения стационарного уравнения Шредингера для оценок $E = 0.77$ и $E = 0.87$. Прямоугольная потенциальная яма имеет параметры $V_0 = 10$ и $a = 1$.

б. Поскольку глубина ямы при $x = |a|$ конечна, то функция ϕ не равна нулю в запрещенной классической механикой области, в которой $E < V_0$ и $x > |a|$. Определите «глубину проникания» как расстояние от $x = a$ до точки, где функция ϕ составляет $\sim 1/e \approx 0.37$ от своего максимального значения. Определите качественную зависимость глубины проникания от величины V_0 .

в. Рассчитайте возбужденные собственные состояния и собственные значения. Сколько всего имеется возбужденных состояний? Почему полное число связанных состояний конечно?

ЗАДАЧА 17.4. Другие одномерные потенциалы

а. Получите численные решения для нескольких первых собственных значений и собственных функций гармонического осциллятора с $V(x) = \frac{1}{2}x^2$. Каким необходимо выбрать шаг Δx для получения энергии основного состояния с точностью 0.1%?

б. Получите численное решение для ангармонического осциллятора с $V(x) = \frac{1}{2}x^2 + bx^4$. Для этого случая не существует аналитического решения и численные решения представляют большой интерес. Вычислите энергию основного состояния для $b = 0.1, 0.2$ и 0.5 . Как зависит энергия основного состояния от b при малых значениях b ? Как зависит собственная функция основного состояния от b ?

в. Получите численное решение для основного состояния линейного потенциала

$$V(x) = |x|. \quad (17.17)$$

Квантовомеханическое рассмотрение этого потенциала может быть использовано для изучения энергетического спектра связанной системы кварк — антикварк, называемой кварконием.

17.4. НЕСТАЦИОНАРНОЕ УРАВНЕНИЕ ШРЕДИНГЕРА

Хотя нам удалось, используя численные методы, проинтегрировать «обыкновенное» дифференциальное уравнение (17.7), решить численно нестационарное уравнение Шредингера (17.4) уже не столь просто.

Первый шаг «наивного» метода численного решения нестационарного

уравнения Шредингера заключается во введении сетки для временной координаты так же, как для пространственной. Воспользуемся обозначениями $t_n = t_0 + n\Delta t$, $x_r = x_0 + r\Delta x$ и $\Psi_r(n) \equiv \Psi(r_n, t_n)$. Задача заключается в построении алгоритма, который связывает $\Psi_r(n+1)$ со значением $\Psi_r(n)$ для каждого значения x_r . Примером алгоритма первого порядка точности по времени является

$$\frac{1}{\Delta t}[\Psi_r(n+1) - \Psi_r(n)] = \frac{1}{(\Delta x)^2} \delta^2 \Psi_r(n), \quad (17.18)$$

где $\delta^2 \Psi_r(n)$ представляет собой конечно-разностную аппроксимацию второй производной Ψ по x :

$$\delta^2 \Psi_r(n) = \Psi_{r+1}(n) - 2\Psi_r(n) + \Psi_{r-1}(n). \quad (17.19)$$

Этот метод является примером *явной* схемы, поскольку, зная Ψ в момент времени t_n , можно вычислить значение Ψ в момент времени t_{n+1} . К сожалению, этот явный метод приводит к неустойчивым решениям, т.е. численное решение Ψ отклоняется от точного (см. книгу Кунина).

Один из способов избежать указанного явления состоит в том, что не меняя формулу схемы (17.18), вычислять производную по координате в правой части в момент времени t_{n+1} , а не t_n :

$$\frac{1}{\Delta t}[\Psi_r(n+1) - \Psi_r(n)] = \frac{1}{(\Delta x)^2} \delta^2 \Psi_r(n+1). \quad (17.20)$$

Уравнение (17.20) представляет собой *неявную* схему, поскольку неизвестная функция $\Psi_r(n+1)$ входит в обе части. Для получения значений $\Psi_r(n+1)$ необходимо на каждом шаге по времени решать систему линейных уравнений. Дополнительные детали этого метода и доказательство того, что (17.20) дает устойчивые решения, можно найти в литературе, приведенной в конце главы.

Вместо данного прямого численного метода предположим, что нам уже известны собственные функции и собственные значения стационарного уравнения Шредингера. Наш подход будет состоять в том, чтобы построить зависимость от времени, используя эти собственные функции и собственные значения и применяя принцип суперпозиции. Будем рассматривать задачу о движении волнового пакета в окрестности потенциала.

Известно, что если $V(x)$ не зависит от времени, то можно выразить произвольное решение нестационарного уравнения Шредингера в виде

(17.10), где функции $\phi_n(x)$ представляют собой решения уравнения Шредингера с потенциалом $V(x)$ для стационарных состояний. Сначала рассмотрим движение свободной частицы. В этом случае собственные функции характеризуются волновым вектором k и пропорциональны $\exp(ikx)$ и $\exp(-ikx)$. Наша задача — построить волновой пакет, соответствующий частице, локализованной в окрестности точки $x = x_0$ со средним импульсом $\hbar k_0$, направленным вправо. Один из способов построения такого волнового пакета состоит в составлении линейной комбинации плоских волн с весовой функцией

$$c_k = N \exp[-i(k - k_0)x_0] \exp[-\alpha(k - k_0)^2]. \quad (17.21)$$

Величина N является нормировочной постоянной. Причины такой формы коэффициентов c_k и смысл величины α будут обсуждаться в задаче 17.5. Поскольку включить в сумму можно только конечное число плоских волн n_k , приближенная форма волнового пакета при $t = 0$ имеет вид

$$\Psi(x, 0) \approx \sum_k c_k e^{ikx}, \quad (17.22)$$

где суммирование проводится по $k_n = k_0 + n\Delta k$ с $-n_k/2 \leq n \leq n_k/2$. Поскольку собственные значения гамильтониана свободной частицы равны $E_k = \hbar^2 k^2 / 2m$, временная эволюция функции $\Psi(x, t)$ определяется приближенным соотношением

$$\Psi(x, t) \approx \sum_k c_k \exp[i(kx - k^2 t / 2)]. \quad (17.23)$$

В программе **wavepacket** используются выражения (17.21) и (17.23) для наблюдения за временной эволюцией волнового пакета в отсутствие потенциала. Из-за наличия комплексных чисел необходимо в языке True BASIC рассматривать отдельно вещественные и мнимые части c_k и Ψ и использовать соотношение

$$e^{ikx} = \cos kx + i \sin kx. \quad (17.24)$$

Входными параметрами в программе **wavepacket** являются $x0$ — координата центра волнового пакета в момент времени $t = 0$, delta_x — начальная

ширина пакета в пространстве координат, k_0 —средний импульс волнового пакета и n_k —число волновых векторов, учитываемых в сумме (17.23). Для каждого значения t функция $\Psi(x, t)$ вычисляется в точках $x_r = x_0 + r dx$, где dx —входной параметр и $-\frac{1}{2}n_x < r < \frac{1}{2}n_x$. Программа рисует график $|\Psi(x, t)|^2$ для моментов времени от $t = 0$ до $t = t_{\max}$ с шагом dt и для значений x , лежащих в интервале от x_{\min} до x_{\max} .

PROGRAM wavepacket

! расчет движения гауссова волнового пакета

DIM Re_c(-50 to 50), Im_c(-50 to 50), prob(-50 to 50)

DIM Re_phi(-50 to 50, -50 to 50)

DIM Im_phi(-50 to 50, -50 to 50)

CALL parameters(k0, delta_x, x0, xmin, dx, dk, dt, tmax, n_k, n_x)

CALL packet(delta_x, x0, n_k, dk, Re_c, Im_c)

CALL free(n_k, n_x, k0, dk, dx, Re_c, Im_c, Re_phi, Im_phi)

DO

LET t = t + dt

CALL move(t, n_k, n_x, k0, dk, dx, Re_phi, Im_phi, prob)

CALL plot(prob, xmin, n_x, dx)

LOOP until t > tmax

END

SUB parameters(k0, delta_x, x0, xmin, dx, dk, dt, tmax, n_k, n_x)

INPUT prompt "минимальное значение x = ": xmin

INPUT prompt "максимальное значение x = ": xmax

INPUT prompt "начальная координата пакета = ": x0

INPUT prompt "ширина волнового пакета в x-пространстве = ": delta_x

INPUT prompt "размер шага = ": dx ! не путать с delta_x

LET n_x = (xmax - xmin)/dx ! число точек, в которых вычисляется psi

INPUT prompt "центральный волновой вектор пакета = ": k0

INPUT prompt "число волновых векторов = ": n_k

LET dk = 2/(delta_x*n_k)

LET n_k = n_k/2

LET n_x = n_x/2

INPUT prompt "шаг по времени = ": dt

INPUT prompt "полное время = ": tmax

LET ymax = 0.5

SET window xmin, xmax, -0.1, ymax

END SUB

```

SUB packet(delta_x, x0, n_k, dk, Re_c(), Im_c())
  ! вычисляются коэффициенты c(k)
  LET sum = 0                                ! используется для нормировки c(k)
  FOR ik = -n_k to n_k
    LET delta_k = ik*dk                      ! разность между k и k0
    LET c = exp(-delta_k*delta_k*delta_x*delta_x/4)
    ! фаза обусловлена стартом в точке x0 не равной 0
    LET Re_c(ik) = c*cos(-delta_k*x0)
    LET Im_c(ik) = c*sin(-delta_k*x0)
    LET sum = sum + Im_c(ik)*Im_c(ik)
    LET sum = sum + Re_c(ik)*Re_c(ik)
  NEXT ik
  LET norm = dk*sqr(1/sum)
  FOR ik = -n_k to n_k
    LET Re_c(ik) = norm*Re_c(ik)
    LET Im_c(ik) = norm*Im_c(ik)
  NEXT ik
END SUB

SUB free(n_k, n_x, k0, dk, dx, Re_c(), Im_c(), Re_phi(, ), Im_phi(, ))
  ! вычисляются величины a(k) exp(ikx)
  FOR ik = -n_k to n_k
    LET k = k0 + ik*dk
    FOR ix = -n_x to n_x
      LET arg = k*ix*dx
      LET fr = cos(arg)
      LET fi = sin(arg)
      LET Re_phi(ik, ix) = fr*Re_c(ik) - fi*Im_c(ik)
      LET Im_phi(ik, ix) = fr*Im_c(ik) + fi*Re_c(ik)
    NEXT ix
  NEXT ik
END SUB

```

```

SUB move(t,n_k,n_x,k0,dk,dx,Re_phi(,),Im_phi(,),prob())
! вычисление суммы по k выражений psi(k,x)exp(-iEt) и расчет psi(x,t)
FOR ix = -n_x to n_x
  LET Re_psi = 0
  LET Im_psi = 0
  FOR ik = -n_k to n_k
    LET k = k0 + ik*dk
    LET arg = -0.5*k*k*t
    LET fr = cos(arg)
    LET fi = sin(arg)
    ! вычисление действительной и мнимой частей psi
    LET Re_psi = Re_psi + Re_phi(ik,ix)*fr
    LET Re_psi = Re_psi - Im_phi(ik,ix)*fi
    LET Im_psi = Im_psi + Re_phi(ik,ix)*fi
    LET Im_psi = Im_psi + Im_phi(ik,ix)*fr
  NEXT ik
  LET prob(ix) = Re_psi*Re_psi + Im_psi*Im_psi
NEXT ix
END SUB

SUB plot(prob(,),xmin,n_x,dx)
  CLEAR
  FOR i = -n_x to n_x
    LET x = i*dx
    PLOT x,prob(i);
  NEXT i
  PLOT
  LET xmax = x
  PLOT LINES: xmin,0;xmax,0
  FOR i = -n_x to n_x
    LET x = i*dx
    PLOT LINES: x,0; x,0.01
  NEXT i
END SUB

```

ЗАДАЧА 17.5. Движение свободного волнового пакета

а. Приведите явные формулы для коэффициентов c_k , использованных в программе **wavpacket**? При каком значении k коэффициент c_k максимален? При каком значении k величина $|c_k|^2$ в e раз меньше сво-

его максимального значения? Что целесообразно считать шириной $|c_k|^2$? Как в программе **wavepacket** определяется ширина пакета в k -пространстве? Как обосновать соотношение $dk = 2/\text{delta_x} * n_k$, используемое в программе **wavepacket**? Напомним, что мы приняли систему единиц с $\hbar = m = 1$.

б. Используя программу **wavepacket**, проследите за движением свободной частицы в области без потенциала. Задайте $xmin = -10$, $xmax = 10$, $x_0 = -7$, $\text{delta_x} = 1$, $dx = 0.5$, $k_0 = 2$, $n_k = 20$, $dt = 0.5$ и $tmax = 10$. Как меняется форма волнового пакета с течением времени? Зависит ли форма волнового пакета от выбора параметров dx и n_k ?

в. Модифицируйте программу **wavepacket** так, чтобы можно было непосредственно измерять зависимость от времени величин $x_0(t)$ и $\Delta x(t)$ — координаты и ширины пакета. Как имеет смысл определить $\Delta x(t)$? (Выберите такое определение $\Delta x(t)$, чтобы $\Delta x(t=0) = \text{delta_x}$.) Как качественно зависят $x_0(t)$ и $\Delta x(t)$ от t ? Как изменятся ваши результаты, если начальную ширину волнового пакета уменьшить в четыре раза?

Теперь проследим за движением волнового пакета при наличии потенциала. Необходимо заменить собственные функции $\exp(ikx)$ в формуле (17.23) на собственные функции гамильтониана $H = p^2/2m + V(x)$. Рассмотрим одномерный потенциальный барьер при $x = 0$:

$$V(x) = \begin{cases} 0, & x \leq 0, \\ V_0, & x > 0. \end{cases} \quad (17.25)$$

Собственные функции при $E > V_0$ имеют вид

$$\phi(x) = \begin{cases} \exp(ik_1x) + B \exp(-ik_1x), & x \leq 0, \\ S \exp(ik_2x), & x > 0, \end{cases} \quad (17.26a)$$

где

$$k_1^2 = \frac{2mE}{\hbar^2},$$

$$k_2^2 = \frac{2m(E - V_0)}{\hbar^2},$$

$$B = \frac{k_1 - k_2}{k_1 + k_2},$$

$$S = \frac{2k_1}{k_1 + k_2}. \quad (17.266)$$

Если $E < V_0$, то решение является аналитическим продолжением функции (17.26) с заменой $k_2 \rightarrow ik$. Основное изменение, которое необходимо сделать в программе **wavepacket**, заключается в замене подпрограммы **free** подпрограммой **step**. Заметим, что в подпрограмме **step** форма собственных функций зависит от положения и энергии волнового пакета относительно барьера.

```

SUB step(n_k,n_x,k0,dk,dx,Re_c(),Im_c(),Re_phi(),Im_phi(),)
  ! вычисляются c(k) [exp(ik1x) + Bexp(-ik1x)] при x <= 0
  ! c(k) S exp(ik2x) при x > 0
  PRINT "барьер при x = 0"
  INPUT prompt "высота барьера = ": V0
  FOR ik = -n_k to n_k
    LET k1 = k0 + ik*dk
    LET E = .5*k1*k1
    ! масса равна единице
    IF E > V0 then
      LET k2 = sqrt(2*(E - V0))
      LET Re_B = (k1 - k2)/(k1 + k2)
      LET Im_B = 0
      LET Re_S = 2*k1/(k1 + k2)
      LET Im_S = 0
    ELSE
      LET k2 = sqrt(2*(V0 - E))
      LET denom = k1*k1 + k2*k2
      LET Re_B = (k1*k1 - k2*k2)/denom
      LET Im_B = 2*k1*k2/denom
      LET Re_S = 2*k1*k1/denom
      LET Im_S = -2*k1*k2/denom
    END IF
    ! вычисление psi(k,x, t = 0)
  FOR ix = -n_x to 0
    LET arg = k1*ix*dx
    LET cs = cos(arg)
  
```

```

LET si = sin(arg)
! используем равенства sin(-kx) = -sin(kx) и cos(-kx) = cos(kx)
LET Re_g = cs + Re_B*cs + Im_B*si
LET Im_g = si + Im_B*cs - Re_B*si
LET Re_phi(ik,ix) = Re_g*Re_c(ik) - Im_g*Im_c(ik)
LET Im_phi(ik,ix) = Re_g*Im_c(ik) + Im_g*Re_c(ik)
NEXT ix
FOR ix = 1 to n_x
  LET arg = k2*ix*dx
  IF E > V0 then          ! осциллирующая волновая функция
    LET cs = cos(arg)
    LET si = sin(arg)
  ELSE
    LET cs = exp(-arg)    ! экспоненциальное затухание
    LET si = 0
  END IF
  LET Re_g = Re_S*cs - Im_S*si
  LET Im_g = Im_S*cs + Re_S*si
  LET Re_phi(ik,ix) = Re_g*Re_c(ik)-Im_g*Im_c(ik)
  LET Im_phi(ik,ix) = Re_g*Im_c(ik)+Im_g*Re_c(ik)
NEXT ix
NEXT ik
END SUB

```

ЗАДАЧА 17.6. Падение волнового пакета на потенциальный барьер

а. Воспользуйтесь программой **wavepacket**, заменив в ней подпрограмму **free** на подпрограмму **step**, и рассчитайте движение волнового пакета, падающего на потенциальный барьер, расположенный при $x = 0$. Положите $xmin = -10$, $xmax = 10$, $x_0 = -7$, $delta_x = 1$, $dx = 0.5$, $k0 = 2$, $n_k = 20$, $dt = 0.5$, $tmax = 10$ и высоту барьера $V_0 = 2$. Качественно опишите движение волнового пакета. Сохраняется ли форма пакета гауссовой для всех t ? Измерьте высоту и ширину отраженного и проходящего волновых пакетов. Определите время t_i столкновения падающей волны с барьером в точке $x = 0$. Распадается ли частица на части при $x = 0$? Определите время t_r , за которое отраженная волна возвращается в точку $x = x_0$. Выполняется ли равенство $t_i = t_r$? Если эти времена не равны, объясните причину расхождения.

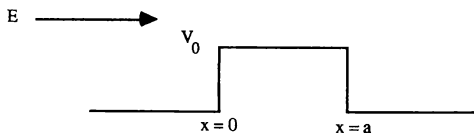


Рис. 17.3. Потенциальный барьер высотой V , расположенный между $x = 0$ и $x = a$.

б. Проведите аналогичное исследование для значения $V_0 = 10$. Выполняется ли условие $t_i \approx t_r$ в этом случае?

в. Как движется классическая частица с кинетической энергией, соответствующей центральному волновому вектору $k = k_0$?

ЗАДАЧА 17.7. Рассеяние волнового пакета на потенциальном барьере

а. Рассмотрим потенциальный барьер вида (рис. 17.3)

$$V(x) = \begin{cases} 0, & x < 0, \\ V_0, & 0 \leq x \leq a, \\ 0, & x > a. \end{cases} \quad (17.27)$$

Собственные функции при $E < V_0$ имеют вид (для частицы, налетающей слева)

$$\phi(x) = \begin{cases} \exp(ikx) + B \exp(-ikx), & x < 0, \\ C \exp(-\kappa x) + D \exp(\kappa x), & 0 \leq x \leq a, \\ E \exp(ikx), & x > a, \end{cases} \quad (17.28a)$$

где

$$k^2 = \frac{2mE}{\hbar^2},$$

$$\kappa^2 = \frac{2m(V_0 - E)}{\hbar^2},$$

$$B = \frac{i(k^2 + \kappa^2) \operatorname{sh} \kappa a}{\Delta},$$

$$C = \frac{k(ik + \kappa) \exp(-\kappa a)}{\Delta},$$

$$D = \frac{-k(ik - \kappa) \exp(\kappa a)}{\Delta},$$

$$E = \frac{2k\kappa \exp(-\kappa a)}{\Delta},$$

$$\Delta = 2k\kappa \operatorname{ch} \kappa a - i(k^2 - \kappa^2) \operatorname{sh} \kappa a. \quad (17.286)$$

Используйте эти собственные функции и сделайте соответствующие изменения в подпрограмме **step**. Создайте серию «снимков», показывающих, как пакет приближается к барьеру и затем взаимодействует с ним, порождая проходящий и отраженный пакеты. Положите $a = 1$ и $V_0 = 2$ и рассмотрите поведение волнового пакета для значений $k_0 = 1, 1.5, 2$ и 3 . При каких значениях k_0 движение волнового пакета качественно совпадает с движением классической частицы? Расплывается ли пакет со временем? Как зависит изменение ширины пакета от его энергии?

б. Рассмотрите прямоугольную потенциальную яму с $V_0 = -2$ и обсудите те же вопросы, что и в п. «а».

ЗАДАЧА 17.8. Движение волнового пакета внутри бесконечной потенциальной ямы

а. Рассмотрите движение волнового пакета, которое ограничено стенками в бесконечной потенциальной ямы шириной $2a$. Постройте волновой пакет, состоящий из первых десяти собственных функций бесконечной потенциальной ямы. Центр волнового пакета расположите в точке $x = 0$, функции c_k выберите гауссовыми, ширину $|c_k|^2$ задайте равной a . Учтите в сумме как положительные, так и отрицательные волновые векторы. Заметим, что значения k дискретны и должны равняться $n\pi/2a$. Опишите временную эволюцию волнового пакета. Какое движение соответствует классическому движению частицы в ящике?

б. Выберите начальную ширину пакета намного меньшей $2a$. Чем отличается движение волнового пакета от движения в п. «а»?

в. Повторите вычисления п. «а», учитывая только положительные волновые векторы. Изменится ли движение волнового пакета?

17.5. АНАЛИЗ КВАНТОВЫХ СИСТЕМ С ПОМОЩЬЮ МЕТОДА СЛУЧАЙНЫХ БЛУЖДЕНИЙ

Познакомимся теперь с методом Монте-Карло, в основе которого лежит связь уравнения Шредингера с процессом диффузии в мнимом времени. Наш метод следует статье Андерсона (см. список литературы). Чтобы понять указанную связь, произведем в нестационарном уравнении Шредингера для свободной частицы замену $\tau = i t / \hbar$ и получим

$$\frac{\partial \Psi(x, \tau)}{\partial \tau} = \frac{\hbar^2}{2m} \frac{\partial^2 \Psi(x, \tau)}{\partial x^2}. \quad (17.29)$$

Заметим, что уравнение (17.29) совпадает по форме с уравнением диффузии (17.1) и, следовательно, мы можем интерпретировать волновую функцию как плотность вероятности диффузионного процесса с коэффициентом диффузии $D = \hbar^2 / 2m$.

Может показаться, что формальная аналогия между процессом диффузии и уравнением Шредингера для свободной частицы только усложнит решение последнего. Однако вспомним, что в гл. 11 мы установили возможность использования метода случайного блуждания для моделирования решения уравнения диффузии. Следовательно, вместо того, чтобы решать непосредственно уравнение Шредингера, можно заменить его на эквивалентную задачу о случайном блуждании.

Поскольку уравнение Шредингера линейно, можно представить себе не одного пешехода, а много пешеходов, которые движутся независимо друг от друга. Как можно объяснить роль потенциальной энергии $V(x)$? Перепишем уравнение Шредингера в следующем виде:

$$\frac{\partial \Psi(x, \tau)}{\partial \tau} = \frac{\hbar^2}{2m} \frac{\partial^2 \Psi(x, \tau)}{\partial x^2} - V(x) \Psi(x). \quad (17.30)$$

Если бы член $V\Psi$ отсутствовал, то выражение (17.30) было бы обычным уравнением диффузии. С другой стороны, если бы в правой части был только один член $V\Psi$, то (17.30) можно было бы понимать как кинетическое уравнение, описывающее ветвящиеся процессы, такие как радиоактивный распад или экспоненциальные процессы рождения и смерти в популяции. Следовательно, в целом это уравнение можно трактовать как комбинацию диффузионного и ветвящегося процессов, в которых число пешеходов увеличивается или уменьшается в точке x в зависимости от знака функции $V(x)$. Заметим, что в диффузионной интерпре-

тации требуется, чтобы амплитуда вероятности была неотрицательной.

Теперь, когда мы начали понимать правила игры, давайте посмотрим, как можно определить волновую функцию и энергию основного состояния. Известно, что если провести замену $\tau = it/\hbar$, то общее решение уравнения Шредингера можно записать [см. (17.10)] в виде

$$\Psi(x, t) = \sum_n c_n \phi(x) \exp(-E_n \tau). \quad (17.31)$$

Для достаточно больших τ основной вклад в сумму (17.31) дает член, соответствующий состоянию с наименьшей энергией. Отсюда, имеем

$$\Psi(x, \tau) \rightarrow c_0 \phi_0(x) \exp(-E_0 \tau). \quad (17.32)$$

Из соотношения (17.32) видно, что пространственная зависимость распределения вероятностей случайных пешеходов для больших значений τ пропорциональна собственной функции основного состояния ϕ_0 . Однако мы также видим, что плотность вероятности $\Psi(x, t)$ того, что пешеход находится в точке x , в конце концов стремится к нулю, если E_0 не равно нулю. Эту проблему можно обойти, отсчитывая E_0 от произвольной начальной энергии V_{ref} , которая выбирается так, чтобы достигалось приблизительно стационарное распределение.

Разработаем метод нахождения энергии основного состояния. Хотя можно попытаться подогнать зависимость от t вычисляемого распределения вероятностей случайных пешеходов под формулу (17.32) и исключить E_0 , эта процедура не привела бы к получению точных значений энергии основного состояния E_0 . В дальнейшем мы покажем, что E_0 можно найти из соотношения

$$E_0 = \langle V \rangle = \frac{\sum n_i V(x_i)}{\sum n_i}, \quad (17.33)$$

где n_i — число пешеходов, находящихся в точке x_i в момент времени τ . Тогда можно получить оценку для E_0 , вычисляя сумму в (17.33) для нескольких значений τ , если получено стационарное распределение.

Для вывода формулы (17.33) перепишем (17.30) и (17.32) в явном виде, вводя начало отсчета потенциала (опорный потенциал) V_{ref} :

$$\frac{\partial \Psi(x, \tau)}{\partial \tau} = \frac{\hbar^2}{2m} \frac{\partial^2 \Psi(x, \tau)}{\partial x^2} - [V(x) - V_{ref}] \Psi(x) \quad (17.34)$$

и

$$\Psi(x, \tau) \approx c_0 \phi_0(x) \exp [-(E_0 - V_{ref})\tau]. \quad (17.35)$$

Проинтегрируем сначала (17.34) по x . Поскольку $\partial\Psi/\partial x$ при $|x| \rightarrow \infty$ стремится к нулю, получим

$$\int \frac{\partial\Psi(x, \tau)}{\partial\tau} dx = - \int V(x)\Psi(x, \tau) dx + V_{ref} \int \Psi(x, \tau) dx. \quad (17.36)$$

Если продифференцировать выражение (17.35) по τ , то получим соотношение

$$\frac{\partial\Psi(x, \tau)}{\partial\tau} = (V_{ref} - E_0)\Psi(x, \tau). \quad (17.37)$$

Подставим выражение (17.37) для $\frac{\partial\Psi(x, \tau)}{\partial\tau}$ в (17.36) и получим

$$\int (V_{ref} - E_0)\Psi(x, t) dx = - \int V(x)\Psi(x, t) dx + V_{ref} \int \Psi(x, t) dx. \quad (17.38)$$

Следовательно, если в (17.38) сократить члены, содержащие V_{ref} , то получим

$$E_0 \int \Psi(x, \tau) dx = \int V(x)\Psi(x, \tau) dx \quad (17.39)$$

или

$$E_0 = \frac{\int V(x)\Psi(x, t) dx}{\int \Psi(x, \tau) dx}. \quad (17.40)$$

Искомый результат (17.33) получается отсюда связыванием функции $\Psi(x)$ с числом пешеходов в точке x .

Хотя вывод формулы (17.33), возможно, показался довольно сложным, правила случайного блуждания просты:

1. Помещаем N_0 пешеходов в начальные точки x_i , где x_i необязательно совпадают с узлами сетки.
2. Вычисляем опорный потенциал $V_{ref} = \frac{1}{N_0} \sum_i V_i$.

3. Случайным образом передвигаем пешехода вправо или влево на заданный шаг Δs . Длина шага Δs связана с шагом по времени Δt соотношением $(\Delta s)^2 = D\Delta t$. (Заметим, что $D = \frac{1}{2}$ в системе единиц с $\hbar = m = 1$.)
4. Вычисляем $\Delta V = [V(x) - V_{ref}]\Delta t$ и случайное число r на отрезке $[0, 1]$. Если $\Delta V > 0$ и $r < \Delta V$, пешеход удаляется. Если $\Delta V < 0$ и $r < -\Delta V$, то добавляем другого пешехода в точку x . (Эта процедура является точной только в пределе бесконечно малого шага по времени.)
5. Повторяем шаги 3–4 для каждого из N_0 пешеходов. Затем заменяем опорный потенциал на

$$V_{ref} = \langle V \rangle - \frac{1}{N_0}(N - N_0)\Delta t, \quad (17.41)$$

где N —новое число пешеходов, а $\langle V \rangle$ —средняя потенциальная энергия N пешеходов. Это среднее значение V представляет собой оценку энергии основного состояния.

6. Повторяем шаги 3–5 до тех пор, пока энергия основного состояния не достигнет стационарного значения со случайными флуктуациями. Затем усредняем эти оценки по многим испытаниям для получения средней энергии основного состояния и рассчитываем распределение случайных пешеходов.

В программе **qwalk** данный алгоритм реализован для потенциала гармонического осциллятора. Пешеходы распределяются случайным образом на расстоянии *srange* от начала координат. Другими входными параметрами являются N —требуемое число пешеходов, *ntrial*—число испытаний на одного пешехода и *ds*—размер шага. Программа выводит десять значений энергии, каждое из которых получено усреднением по 10% испытаний. Первые 10% испытаний при усреднении не учитываются.

```

PROGRAM qwalk
DIM xsite(2000)
RANDOMIZE
CALL initial(N,srange,ds,ntrials,Vref,xsite)
CALL walk(N,srange,ds,ntrials,Vref,xsite)
END

```



```

SUB initial (N,srange,ds,ntrials,Vref,xsite())
  DECLARE DEF V
  INPUT prompt "требуемое число пешеходов = ": N
  INPUT prompt "введите длину шага = ": ds
  INPUT prompt "число испытаний = ": ntrials
  INPUT prompt "начальная область размещения пешеходов = ": srange
  FOR i = 1 to N
    LET xsite(i) = srange*(0.5 - rnd)      ! координата i-го пешехода
    LET Vref = Vref + V(xsite(i))
  NEXT i
  LET Vref = Vref/N
  PRINT "энергия","число пешеходов","опорный потенциал"
  PRINT
END SUB

SUB walk (N,srange,ds,ntrials,Vref,xsite())
  DECLARE DEF V
  DIM psi(-500 to 500)
  LET ninitial = N      ! число пешеходов в начале испытания
  LET nwalker = N      ! число пешеходов в ходе испытания
  LET dt = ds*ds      ! шаг по времени
  LET nbar = 10      ! число усреднений
  LET nave = ntrials/nbar ! число испытаний для усреднения энергии
  LET nequil = 0.1*ntrials ! число испытаний до "устойчивого состояния"
  LET dx_plot = 0.2      ! шаг графика волновой функции
  LET scale = srange/dx_plot ! масштабирование для индекса массива psi
  FOR itrial = 1 to ntrials
    LET vsum = 0.0      ! полная потенциальная энергия
    FOR iwalker = ninitial to 1 step -1
      IF rnd < 0.5 then
        LET xsite(iwalker) = xsite(iwalker) + ds
      ELSE
        LET xsite(iwalker) = xsite(iwalker) - ds
      END IF
    
```

```

LET potential = V(xsite(iwalker))      ! потенциал в xsite
LET dv = potential - Vref
IF dv < 0.0 then      ! проверка на добавление пешехода
  IF rnd < -dv*dt then
    LET nwalker = nwalker + 1
    LET xsite(nwalker) = xsite(iwalker)      ! новый пешеход
    LET vsum = vsum + 2*potential
  ELSE
    LET vsum = vsum + potential ! только для старого пешехода
  END IF
ELSE
  IF rnd < dv*dt then      ! проверка на удаление пешехода
    ! замена пешехода новым
    LET xsite(iwalker) = xsite(nwalker)
    LET nwalker = nwalker - 1
  ELSE
    LET vsum = vsum + potential
  END IF
END IF
NEXT iwalker
LET vave = vsum/nwalker      ! средний потенциал
! принятие решения о печати средних
IF mod(itrial,nave) = 0 then
  LET Eave = (vave + esum)/nave
  PRINT Eave,nwalker,Vref
  LET esum = 0
ELSE
  LET esum = esum + vave
END IF
IF ntrials > nequil then
  FOR iwalker = 1 to ninitial
    LET x = xsite(iwalker)*scale
    LET psi(x) = psi(x) + 1      ! число пешеходов в каждом узле
  NEXT iwalker
END IF
LET Vref = vave - (nwalker - N)/(N*dt) ! новый опорный потенциал
LET ninitial = nwalker
NEXT itrial
CALL wave_function(psi,scale)
END SUB

```

```

SUB wave_function (psi(), scale)
  INPUT prompt "строить график волновой функции ?": choice$
  IF choice$ = "Y" then
    FOR i = -100 to 100
      LET sum = sum + psi(i)*psi(i)
    NEXT i
    LET norm = sqr(sum/scale)
    SET window -100,100,-0.5,2
    PLOT LINES: -100,0;100,0
    PLOT LINES: -scale,-0.1;-scale,0.1 ! нанесение деления в xsite = -1
    PLOT LINES: scale,-0.1;scale,0.1 ! нанесение деления в xsite = 1
    FOR i = -99 to 100
      PLOT LINES: i-1,psi(i-1)/norm;i,psi(i)/norm
    NEXT i
  END IF
END SUB

DEF V (xsite)
  LET V = 0.5*xsite*xsite
END DEF

```

ЗАДАЧА 17.9. Основные состояния гармонического и ангармонического осцилляторов

а. С помощью программы `qtmwalk` вычислите энергию основного состояния и собственную функцию для потенциала $V(x) = \frac{1}{2}x^2$. Положите начальное число пешеходов равным $N = 50$, длину шага $ds = 0.1$ и $ntrials = 100$. Разместите пешеходов случайным образом в области $-1 < x < 1$, т.е. задайте $srange = 2$. Сравните свою оценку E_0 , полученную методом Монте-Карло, с точным значением $E_0 = 0.5$.

б. Положите $ntrials = 400$. Насколько улучшится оценка энергии E_0 в этом случае? Сколько потребуется испытаний для получения значения E_0 с точностью 1%? Постройте график пространственного распределения случайных пешеходов и сравните его с точной волновой функцией основного состояния.

в. Получите численное решение для ангармонического осциллятора с потенциалом

$$V(x) = \frac{1}{2}x^2 + bx^3 \quad (b \ll 1). \quad (17.42)$$

Рассмотрите следующие значения параметра: $b = 0.1, 0.2$ и 0.5 . Для данного потенциала не существует аналитических решений, и поэтому численные решения имеют большое значение. Оценка влияния члена x^3 необходима для изучения аангармоничности колебаний той или иной физической системы, например колебательного спектра двухатомных молекул.

ЗАДАЧА 17.10. Основное состояние прямоугольной потенциальной ямы

а. С помощью программы `qmwalk` вычислите энергию и волновую функцию основного состояния прямоугольной потенциальной ямы (17.14). Положите $V_0 = 5$, $N = 100$, $ds = 0.1$, $srange = 2$ и $ntrials = 300$.

б. Увеличивайте значения V_0 и найдите зависимость энергии основного состояния от V_0 . Используя полученные результаты, оцените предельное значение энергии основного состояния при $V_0 \rightarrow \infty$.

в. Модифицируйте потенциал так, чтобы $V(x) = V_b$ при $|x| \leq 0.2$, $V(x) = 0$ при $0.2 < |x| \leq 1$ и $V(x) = V_0$ при $|x| > 1$. Положите $V_0 = 10$ и $V_b = 2$. Как изменятся энергия и собственная функция основного состояния?

ЗАДАЧА 17.11. Основное состояние цилиндрического потенциала

Вычислите энергию и волновую функцию основного состояния для двумерного цилиндрического потенциала

$$V(r) = \begin{cases} 0, & r \leq 1, \\ -V_0, & r > 1, \end{cases} \quad (17.43)$$

где $r^2 = x^2 + y^2$. Модифицируйте программу `qmwalk`, используя двумерные декартовы координаты, например добавьте в нее массив `ysite` для хранения y -компонент координат пешеходов. Что получится, если начать моделирование не с аксиальносимметричного исходного распределения пешеходов?

Приведенный выше алгоритм случайного блуждания можно улучшить, вводя метод выборки по значимости. Идея заключается в использовании такого начального приближения волновой функции, которое заставит пешеходов проводить больше времени в наиболее значимых областях потенциала. Если записать $f(x, \tau) = \Psi(x, \tau) \phi_T(x, \tau)$, то можно показать, что уравнение движения для функции $f(x, \tau)$ по форме аналогично исходному уравнению Шредингера и его можно исследовать на основе процесса случайного блуждания с наложенными дрейфовым и ветвящимся членами. Применение хорошей пробной волновой функции ϕ_T уменьшает дисперсию и делает возможным решение уравнения Шредингера для нескольких сотен частиц. Другое усовершенствование метода случайного блуждания основано на переформулировании задачи диффузии, которое полностью исключает систематические ошибки, обусловленные конечностью шага по времени. Такая формулировка называется методом Монте-Карло для функции Грина. Более подробное изложение методов выборки по значимости и Монте-Карло для функции Грина можно найти в литературе, приведенной в конце главы.

17.Е. ВАРИАЦИОННЫЕ МЕТОДЫ МОНТЕ-КАРЛО ДЛЯ КВАНТОМЕХАНИЧЕСКИХ СИСТЕМ

Другой подход к моделированию квантовомеханических систем заключается в использовании вариационного метода для получения верхних границ значений энергии основного состояния. Такой вариационный метод имеет многочисленные приложения, особенно в атомной и молекулярной физике, ядерной физике и физике конденсированного состояния. Поскольку вариационный метод рассмотрен в нескольких учебниках, мы только кратко опишем здесь основы этого метода. Затем обсудим некоторые методы Монте-Карло для реализации вариационного метода.

Рассмотрим физическую систему с гамильтонианом H_{op} вида (17.8). Предположим, что собственные значения E_n и соответствующие собственные функции ϕ_n не известны. Вариационный принцип утверждает, что для произвольной функции $\phi(x)$ выполняется неравенство

$$\langle H \rangle = \frac{\int \phi^*(x) H_{op} \phi(x) dx}{\int \phi^*(x) \phi(x) dx} \geq E_0, \quad (17.44)$$

где E_0 — энергия основного состояния системы. Это неравенство превращается в точное равенство только в том случае, когда $\phi(x)$ является собственной функцией гамильтониана H , соответствующей собственному значению E_0 . Величину $\langle H \rangle$ можно трактовать как математическое ожидание полной энергии для приближенной волновой функции $\phi(x)$.

Неравенство (17.44) лежит в основе вариационного метода. Процедура состоит в выборе пробной волновой функции, форма которой физически обоснована и которая зависит от одного или более параметров. Вычисляется величина $\langle H \rangle$ и параметры варьируются до тех пор, пока не получится минимальное значение $\langle H \rangle$. Это значение $\langle H \rangle$ представляет собой верхний предел истинной энергии основного состояния.

Сначала опишем метод Монте-Карло для получения пробных волновых функций для систем малой размерности. Для простоты рассмотрим одномерную систему и введем сетку с шагом Δx такую, что функция ϕ определена только в точках $x_r = r\Delta x$. Процедуру метода Монте-Карло можно сформулировать следующим образом:

1. Выбираем правдоподобные значения функции ϕ в точках x_r .
2. Выбираем случайную точку x_r и изменяем значение $\phi_r = \phi(x_r)$ на случайную величину из интервала $[-\delta, \delta]$.
3. Вычисляем изменение ΔE для $\langle H \rangle$. Если $\Delta E \leq 0$, то предпринятое изменение $\phi(x_r)$ принимается, в противном случае оно отвергается.
4. Повторяем шаги 2 и 3 до тех пор, пока $\langle H \rangle$ не перестанет существенно меняться.

Данная процедура реализована в программе **variation** для одномерных симметричных потенциалов. Вклад кинетической энергии при $x = x_r$ аппроксимируется выражением $\phi_r(2\phi_r - \phi_{r+1} - \phi_{r-1})/[(2\Delta x)^2]$. Заметим, что поскольку кинетическая энергия зависит от кривизны функции ϕ , то изменение функции ϕ в точке $x = x_r$ меняет энергию не только в точке $x = x_r$, но и в точках $x = x_{r-1}$ и $x = x_{r+1}$.

PROGRAM variation

! получение основного состояния с помощью вариационного метода
! варьируется волновая функция и принимаются только те изменения,
! которые уменьшают энергию

DIM phi(-200 to 200)

CALL initial(dx, N, delta, E, norm, phi, nmove)

CALL changes(dx, N, delta, E, norm, phi, nmove)

END

SUB initial(dx, N, delta, E, norm, phi(), nmove)

DECLARE DEF V

INPUT prompt "максимальное значение x = ": xmax

INPUT prompt "число точек в каждом направлении = ": N

LET dx = xmax/N

INPUT prompt "максимальное изменение волновой функции = ": delta

SET window -xmax, xmax, -0.5, 1

LET delta = 2*delta

LET nmove = 10*N ! число испытаний между графиками

FOR i = 1 to N

LET x = i*dx

LET phi(i) = exp(-.5*x*x) ! начальная функция

LET phi(-i) = phi(i)

LET norm = norm + 2*phi(i)*phi(i) ! нормировка

NEXT i

LET phi(0) = 1

! "интегрирующий" множитель dx опускается в pe, ke и нормировке

LET norm = norm + phi(0)*phi(0)

FOR i = 1 to N ! вычисление начальной энергии

LET x = i*dx

LET pe = phi(i)*phi(i)*V(x)

LET ke = .5*phi(i)*(2*phi(i) - phi(i+1) - phi(i-1))/(dx*dx)

LET E = E + 2*(pe + ke) ! множитель 2 для phi(i) and phi(-i)

NEXT i

LET E = E + V(0)*phi(0)^2

LET E = E + 0.5*phi(0)*(2*phi(0) - phi(1) - phi(-1))/(dx*dx)

END SUB

```

SUB changes (dx,N,delta,E,norm,phi(),nmove)
DECLARE DEF V
DO
  LET Ecum = 0
  FOR imove = 1 to nmove
    LET i = int(N*rnd + 1)          ! координата
    LET dphi = (rnd - 0.5)*delta    ! изменение волновой функции
    LET phi_trial = phi(i) + dphi ! новая волновая функция в i-м узле
    LET dphi2 = phi_trial*phi_trial - phi(i)*phi(i)
    ! изменение потенциальной энергии dV
    ! изменение кинетической энергии dK
    ! dK содержит изменение кинетической энергии в «узлах» i+1
    ! и i-1, обусловленное изменением волновой функции в i-м узле
    LET dV = dphi2*V(i*dx)
    LET dK = (dphi2 - dphi*(phi(i+1) + phi(i-1)))/dx^2
    IF i = 0 then
      LET Etrial = E + dK+ dV      ! пробная энергия
      LET norm_trial = norm + dphi2 ! новая константа нормировки
    ELSE
      ! вносим изменение, обусловленное phi(i) и phi(-i)
      LET Etrial = E + 2*(dv + dk) ! новая энергия
      LET norm_trial = norm + 2*dphi2 ! новая константа нормировки
    END IF
    LET dE = (Etrial/norm_trial) - (E/norm)
    IF dE < 0 then ! если энергия уменьшилась, phi_trial принимается
      LET phi(i) = phi_trial
      LET phi(-i) = phi_trial
      LET norm = norm_trial
      LET E = Etrial
    END IF
    LET Ecum = Ecum + E/norm      ! накопление энергий
  NEXT imove
  LET Eave = Ecum/nmove          ! средняя энергия
  PRINT "среднее значение энергии основного состояния ="; Eave
  INPUT prompt "рисовать, стоп или продолжать (p/s/c) ": choice$
  IF choice$ = "p" then CALL plot_phi(phi,N,dx,norm)
  LOOP UNTIL choice$ = "s"
END SUB

```



```

SUB plot_phi (phi(),N,dx,norm)
  DECLARE DEF V
  ! построение графика волновой функции и потенциала
  CLEAR
  LET dy = 0.01                ! высота делений
  PLOT LINES: -N*dx, 0; N*dx, 0
  PLOT
  FOR i = -N to N              ! нанесение делений
    LET x = i*dx
    PLOT LINES: x, 0; x, dy
    PLOT LINES: -x, 0; -x, dy
  NEXT i
  PLOT
  LET scale = 0.025            ! зависит от высоты потенциала
  FOR i = -N to N
    LET x = i*dx
    PLOT POINTS: x, V(x)*scale
  NEXT i
  LET norm_phi = 1/sqr(norm)   ! нормировка для phi
  PLOT
  FOR i = -N to N              ! построение графика волновой функции
    LET x = i*dx
    LET iabs = abs(i)
    PLOT x, phi(iabs)*norm_phi;
  NEXT i
END SUB

DEF V(x)
  LET V = .5*x*x              ! потенциал гармонического осциллятора
END DEF

```

ЗАДАЧА 17.12. Вычисление методом Монте-Карло основного состояния гармонического и ангармонического осцилляторов

а. Воспользуйтесь программой **variation** для вычисления энергии основного состояния гармонического осциллятора. Входные параметры: N —число точек, в которых вычисляется пробная функция ϕ , и x_{\max} —максимальное значение x . Величина *delta* определяет максимальное изменение функции ϕ . Программа печатает значение ΔE и

строит график функции ϕ после каждых $100N$ испытаний. Положите $N = 40$, $x_{\max} = 4$ и $\delta = 0.1$. Какая в программе **variation** начальная пробная волновая функция основного состояния?

б. Выберите в качестве пробной волновой функции $\phi(x) = \cos x$. Сколько потребуется испытаний для получения оценки энергии основного состояния с точностью 1%? Какова итоговая точность вычисленной пробной волновой функции?

в. Выберите в качестве пробной волновой функции $\phi(x) = 1$. Сколько потребуется испытаний для получения оценки энергии основного состояния с точностью 1%? С какой функцией ϕ получается более быстрая сходимость к результату?

г. Рассмотрим ангармонический потенциал $V(x) = \frac{1}{2}x^2 + bx^4$. Постройте график функции $V(x)$ для значения $b = 1/8$. Используя теорию возмущений первого порядка, вычислите в наинизшем порядке изменение энергии основного состояния, обусловленное членом x^4 . Затем воспользуйтесь программой **variation** для вычисления E_0 и сравните полученное значение с результатом для гармонического осциллятора ($b = 0$). Выберите значения параметров N и x_{\max} такими, чтобы энергия получалась с точностью до 1%. Каков разумный критерий выбора значения параметра δ ?

д. Рассмотрите ангармонический потенциал из п. «г» с $b = -1/8$. Постройте график функции $V(x)$. Используя теорию возмущений первого порядка, вычислите в наинизшем порядке изменение энергии основного состояния, обусловленное членом x^4 . Затем воспользуйтесь программой **variation** и получите оценку E_0 . Имеют ли ваши оценки нижний предел или нет? Почему?

ЗАДАЧА 17.13. Расчеты методом Монте-Карло прямоугольных потенциальных ям

а. Воспользуйтесь программой **variation** для оценки энергии основного состояния прямоугольной потенциальной ямы (17.14) с параметрами $V_0 = 2$ и $a = 1$. Положите $x_{\max} = 2$, $N = 40$ и $\delta = 0.1$ и рассмотрите начальные пробные волновые функции $\phi(x) = \cos x$, $\exp(-x^2/2)$ и 1. Какое из начальных приближений для функции ϕ дает более быструю сходимость к результату? Чему равна «глубина проникания» вычисленной волновой функции?

б. Замените потенциал $V(x)$ на

$$V(x) = \begin{cases} V_b, & |x| \leq 0.2, \\ 0, & 0.2 < |x| \leq 1, \\ 2, & |x| > 1. \end{cases} \quad (17.45)$$

Как, по вашему мнению, будет зависеть энергия основного состояния от V_b ? Предположите, что $V_b \ll V_0$, и найдите новую собственную функцию и собственное значение, начав с пробной функции $\phi(x) = \cos x$. Как соотносятся вычисленные значения с ожидаемыми? Изменяется ли глубина проникания в область $|x| > 1$ за счет замены функции V ?

в. Рассмотрите двойную потенциальную яму ($x > 0$) вида

$$V(x) = \begin{cases} +40, & |x| > 4, \\ -40, & 3 \leq |x| \leq 4, \\ 0, & 2 < |x| < 3, \\ -20, & 1 \leq |x| \leq 2, \\ 0, & |x| < 1. \end{cases} \quad (17.46)$$

Постройте график функции $V(x)$ и изобразите схематически на этом же графике форму начальной пробной функции основного состояния. Рассчитайте функцию основного состояния, используя значения параметров $x_{max} = 4.5$, $N = 45$, $delta = 0.05$ с начальной функцией $\phi = 1$ при $|x| \leq 1.5$ и $\phi = 0$ при $|x| > 1.5$. Постройте график получившейся оценки пробной волновой функции и сравните его с ожидаемой формой функции ϕ . Почему вы не получили разумную оценку функции ϕ ?

г. Повторите вычисления п. «в», но с начальной пробной функцией $\phi = \exp(-x^2)$ (положите $delta = 0.05$). Объясните, почему полученные вами результаты для $\langle H \rangle$ и ϕ отличаются от тех, которые получены в п. «в». Придумайте другой метод пробных изменений функции ϕ , например одновременное изменение функции ϕ для многих значений x , который ускорит сходимость результатов даже при плохом выборе начальной пробной волновой функции.

Вместо расчета функции ϕ на сетке можно предложить вариационный метод на базе разложения ϕ по полному набору базисных функций f_n :

$$\phi = \sum_n c_n f_n. \quad (17.47)$$

В этом методе Монте-Карло будет сохраняться конечное число членов в (17.47) и варьироваться коэффициенты c_n до тех пор, пока не получится разумная оценка минимума $\langle H \rangle$. Хотя этот метод наиболее полезен в случае больших размерностей, мы рассмотрим его только для одномерного случая. Чтобы осуществить данную процедуру, необходимо вычислить матричные элементы вида

$$T_{nm} = \langle n | T | m \rangle, \quad (17.48a)$$

$$V_{nm} = \langle n | V | m \rangle, \quad (17.48b)$$

где T и V — операторы кинетической и потенциальной энергии соответственно. Для полноты заметим, что матричные элементы $\langle n | A | m \rangle$ оператора A в одномерном случае определяется как

$$\langle n | A | m \rangle = \int dx f_n^* A f_m. \quad (17.49)$$

В случае больших размерностей пространства для вычисления матричных элементов можно воспользоваться методами Монте-Карло. Если матричные элементы известны, то математическое ожидание $\langle H \rangle$ вычисляется по формуле

$$\langle H \rangle = \frac{\sum_{n,m} c_n^* c_m (T_{nm} + V_{nm})}{\sum_{n,m} c_n^* c_m}. \quad (17.50)$$

Данный метод Монте-Карло использован в задаче 17.14..

***ЗАДАЧА 17.14.** Вычисление методом Монте-Карло энергии основного состояния для потенциала Морза

Рассмотрим потенциал Морза

$$V(x) = (1 - e^{-x})^2, \quad (17.51)$$

для которого можно получить аналитические решения для собственных значений и волновой функции. Поскольку $V(x) \approx x^2$ для малых значений x , разумно в качестве базисных функций выбрать решения соответствующей задачи о гармоническом осцилляторе:

$$\phi_n = N_n H_n \exp(-x^2/2). \quad (17.52)$$

Полиномы Эрмита имеют вид

$$H_n(x) = (-1)^n \exp(x^2) \frac{d^n}{dx^n} \exp(-x^2). \quad (17.53)$$

Нормировочная постоянная N_n равна

$$N_n = \left[\frac{1}{\pi^{1/2} 2^n n!} \right]^{1/2}. \quad (17.54)$$

Для данного выбора базисных функций матричные элементы T_{nm} можно вычислить аналитически. В результате имеем

$$\langle n | T | m \rangle = \begin{cases} n + \frac{1}{2}, & \text{если } n = m, \\ \frac{1}{2} \sqrt{(n+1)(n+2)}, & \text{если } n = m + 2, \\ 0 & \text{в остальных случаях.} \end{cases} \quad (17.55)$$

Матричный элемент V_{nm} можно выразить через матричный элемент

$$\begin{aligned} \langle n | e^{-ax} | m \rangle &= \pi^{1/2} \exp(a^2/4) n! m! N_n N_m \times \\ &\times \sum_{p=|n-m|}^{n+m} \frac{(-a)^p}{[\frac{1}{2}(n+m-1)]! [\frac{1}{2}(n+1-m)]! [\frac{1}{2}(m+1-n)]!}. \end{aligned}$$

а. Убедитесь в правильности этих матричных элементов. Указание: для вычисления матричного элемента оператора V используйте соотношение

$$\exp(-s^2 + 2sx) = \sum_{n=0}^{\infty} \frac{H_n(x)}{n!} s^n.$$

б. Воспользуйтесь приведенными выше выражениями для матричных элементов и напишите программу, реализующую тот же алгоритм, что и в программе `variation`. Иначе говоря, изменяем c_n на произвольную величину из отрезка $[-\delta, \delta]$, вычисляем ΔE и принимаем предпринятое изменение, только если $\Delta E < 0$. Положите начальные значения $c_0 = 1$ и $c_n = 0$ для $n \neq 0$ и оставьте в суммах (17.50) только первые пять членов. Проведите несколько прогонов программы и вычислите наилучшую оценку для энергии основного состояния E_0 . Сравните свою оценку с точным результатом $E_0 \approx 0.582$. Как изменятся ваши результаты, если в сумме сохранить десять членов? Предполагается, что в своих первых прогонах программы вы положили $\delta = 0.1$. Как изменится оценка энергии основного состояния, если уменьшить δ ?

Придумайте другие задачи, для которых соединение вариационного подхода с методами Монте-Карло даст полезные результаты. Для получения еще лучшей пробной волновой функции имеется другая возможность, состоящая в соединении вариационных методов Монте-Карло с методами случайного блуждания.

ЛИТЕРАТУРА

E. E. Anderson, Modern Physics and Quantum Mechanics, W. B. Saunders, 1971. В этой книге, рассчитанной на подготовленного читателя, рассматриваются примеры движения волнового пакета.

J. B. Anderson, A random walk simulation of the Schrödinger equation: H_3^+ , J. Chem. Phys. **63**, 1499 (1975); Quantum chemistry by random walk. H^2P , $H_3^+D_{3h}^1A_1'$, $H_2^3\Sigma_u^+$, $H_4^1\Sigma_g^+$, Be 1S , J. Chem. Phys. **65**, 4121 (1976); Quantum chemistry by random walk: Higher accuracy, J. Chem. Phys. **73**, 3897 (1980). В этих трех статьях описаны основная процедура метода случайного блуждания, обобщенные версии метода более высокой точности и приложения к задачам о простых молекулах.

G. Baym, Lectures on Quantum Mechanics, W. A. Benjamin, 1973. В гл. 3 хорошо рассмотрено уравнение Шредингера с мнимым временем.

H. A. Bethe, Intermediate Quantum Mechanics, W. A. Benjamin, 1964. Обсуждаются приложения квантовой механики к атомным системам.

J. S. Bolemon, Computer solutions to a realistic 'one-dimensional' Schrödinger equation, Am. J. Phys. **40**, 1511 (1972).

S. Brandt, *H. D. Dahmen*, The Picture Book of Quantum Mechanics, John Wiley & Sons, 1985. Приведено большое число картинок волновых функций, полученных с помощью компьютера для различных задач.

D. M. Ceperley, *B. J. Alder*, Quantum Monte Carlo, Science **231**, 555 (1986). Обзорная статья, посвященная некоторым приложениям методов Монте-Карло в физике и химии.

D. F. Coker, *R. O. Watts*, Quantum simulation of systems with nodal surfaces, Mol. Phys. **58**, 1112 (1986).

R. M. Eisberg, Applied Mathematical Physics with Programmable Pocket Calculators, McGraw-Hill, 1976. В гл. 8 обсуждается численное решение одномерного уравнения Шредингера, полученное прямым методом.

R. M. Eisberg, *R. Resnick*, Quantum Physics, John Wiley & Sons, 1974. См. приложение F, в котором обсуждается численное решение уравнения Шредингера.

R. P. Feynman, Simulating Physics with Computers, Int. J. Theor. Phys. **21**, 467 (1982). Стимулирующее обсуждение трудностей моделирования квантовых систем.

J. P. Killingbeck, Microcomputer Quantum Mechanics, Adam Hilger, 1983. Книга по численным методам и квантовой механике.

S. E. Koonin, Computational Physics, Benjamin/Cummings, 1986. В гл. 7 в контексте параболических уравнений в частных производных рассматривается нестационарное уравнение Шредингера. В гл. 8 обсуждается метод Монте-Карло для функции Грина. [Готовится перевод: Кунин С. Вычислительная физика. — М.: Мир, 1991].

P. K. MacKeown, Evaluation of Feynman path integrals by Monte Carlo methods, Am. J. Phys. **53**, 880 (1985). Авторы обсуждают планы исследований, пригодные для работы аспирантов и студентов старших курсов.

J. R. Merrill, Using Computers in Physics, Houghton Mifflin Co., 1976. Наше рассмотрение нестационарного уравнения Шредингера основывается на этой книге.

W. H. Press, B. P. Flannery, S. A. Teukolsky, W. T. Vetterling, Numerical Recipes, Cambridge University Press, 1986. [Имеется перевод: Пресс В., Фланнери Б., Тьюкольски С., Веттерлинг В., Численные рецепты. — М.: Мир, 1990]. В разд. 17.2 обсуждается численное решение нестационарного уравнения Шредингера.

P. J. Reynolds, D. M. Ceperley, B. J. Alder, W. A. Lester Jr., Fixed-node quantum Monte Carlo for molecules, J. Chem. Phys. **77**, 5593 (1982). В этой статье описан алгоритм случайного блуждания, применяемый в молекулярных системах, с методом выборки по значимости и трактовкой статистики Ферми.

D. Saxon, Elementary Quantum Mechanics, Holden-Day, 1968. В гл. 6 обсуждаются численные решения уравнения Шредингера.

L. I. Schiff, Quantum Mechanics, 3rd. ed., McGraw-Hill, 1968. В гл. 8 обсуждаются приближенные методы для случая ограниченных состояний. [Имеется перевод 2-го изд.: Шифф Л., Квантовая механика. — М.: ИЛ, 1959].

ДОПОЛНИТЕЛЬНАЯ ЛИТЕРАТУРА

Давыдов А. С., Квантовая механика. — М.: Наука, 1973. В книге подробно рассматриваются задачи, связанные с потенциальными барьерами и ямами.

Флюгге К., Задачи по квантовой механике, т. 1. — М.: Мир, 1973. В этой книге приведены аналитические решения для многих задач, рассматриваемых в этой главе.

Крайнов А. В., Мигдал А. Б., Качественные методы в квантовой механике. — М.: Наука, 1976. Очень полезная книга, в которой рассматриваются различные приближенные методы решения квантовомеханических задач.

Мигдал А. Б., Качественные методы в квантовой теории. — М.: Наука, 1975.

Методы Монте-Карло в статистической физике/Под ред. *К. Биндера*. — М.: Мир, 1982. В четвертой статье подробно рассматриваются вариационный метод и метод Монте-Карло для функции Грина.

ЭПИЛОГ:
ОДИНАКОВЫЕ ПРОГРАММЫ – ОДИНАКОВЫЕ РЕШЕНИЯ

Особо выделяется тот факт, что методы, которые были рассмотрены в задачах классической механики и статистической физики, можно применять для самых различных природных явлений.

18.1. ЕДИНСТВО ФИЗИКИ

Мы с вами рассмотрели уже множество тем и приложений, и все же это составляет лишь крохотную часть возможностей численного моделирования физического мира. Нам известно, однако, что одинаковые принципы и алгоритмы применяются ко многим типам явлений и, как убедительно выразился Фейнман, *«одинаковые уравнения — одинаковые решения»*. Поэтому методы Монте-Карло, те же самые, которые мы применили для моделирования диффузии частиц в газе и для анализа квантовомеханических волновых функций, можно использовать и для диффузии нейтронов в графите. Быть может, не столь очевидно, что сходные методы Монте-Карло можно использовать для изучения проблем удержания кварков и химической кинетики. Фактически возрастание роли компьютера в научных исследованиях в настоящее время усиливается взаимодействием различных разделов физики и связью физики с другими науками.

Компьютер не только позволил нам полностью осознать единство физики, но помог по-новому рассматривать явления, что дополняет традиционные методы. Коль скоро мы обсудили множество примеров из области физики, приведем пример из теоретической экологии. Рассмотрим простую модель хищник — жертва, представляющую акул и рыбу. Предположим, что коэффициент рождаемости рыбы не зависит от численности акул и что количество уничтожаемой каждой акулой рыбы пропорционально полному числу рыб. Последнее допущение было бы верным, если бы район охоты каждой акулы был неизменным, районы охоты разных акул не пересекались и каждая акула находила в своем районе постоянную часть рыбного косяка. Если предположить, что функция $F(t)$ — количество рыбы в момент времени t , — непрерывна, то можно записать

$$\frac{dF(t)}{dt} = [b_1 - d_1 S(t)]F(t), \quad (18.1)$$

где $S(t)$ — численность акул в момент времени t , а b_1 и d_1 — постоянные коэффициенты, не зависящие от F и S . Далее, нам необходимо уравнение для скорости изменения числа акул. Допустимо предположить, что число детенышей, рожденных каждой акулой, пропорционально количеству рыбы, съеденной данной акулой. Если, кроме того, считать смертность постоянной, то имеем

$$\frac{dS(t)}{dt} = [b_2 F(t) - d_2]S(t). \quad (18.2)$$

Уравнения (18.1) и (18.2) известны как уравнения Лотки — Вольтерра.

Заметим, что сделанное в уравнении (18.2) предположение о том, что скорость роста пропорциональна произведению численности обеих популяций, аналогично принципу действия массы в химической кинетике, согласно которому скорости реакции растут как произведение концентраций молекул. Систему этих двух уравнений можно исследовать стандартными методами и решить численно с помощью алгоритма Эйлера — Кромера. В состоянии ли вы объяснить, почему динамика поведения решений уравнений (18.1) и (18.2) обладает цикличностью?

В приведенной модели хищник — жертва предполагалось, что число хищников и жертв меняется непрерывным образом. Теперь очертим кратко другую модель хищник — жертва, которую проще всего сформулировать в виде вычислительного алгоритма. Модель, которую мы опишем, является двумерным клеточным автоматом, называемым Ва — Тор (см. список литературы).

- i. Исходя из требуемой концентрации рыб и акул, они помещаются случайным образом в узлы прямоугольной сетки. Всем рыбам и акулам присваивается случайный возраст.
- ii. На временном шаге t_n рассматривается по очереди каждая рыба. Определяется число ближайших соседних узлов, которые не заняты в момент времени t_{n-1} , и рыба передвигается в один из свободных узлов случайным образом. Если все четыре соседних узла заняты, рыба не перемещается.
- iii. Если рыба выживает за время, кратное числу шагов f_{breed} , у рыбы появляется один потомок. Новая рыба помещается в старую позицию рыбы-родителя.
- iv. На временном шаге t_n рассматривается по очереди каждая акула. Если все ближайшие к акуле соседние узлы в момент времени t_{n-1} не заняты, акула перемещается в один из четырех свободных узлов случайным образом. Если хоть в одном соседнем узле находится рыба, акула перемещается случайным образом в один из этих занятых узлов и поедает рыбу.
- v. Если за n_{starve} шагов акула ничего не съедает, она погибает. Если акула выживает в течение времени, кратного s_{bread} шагов, то она обзаводится одним детенышем. Новая акула помещается в предыдущую позицию своего родителя.

Каково динамическое поведение Ва — Тор? Проявляют ли Ва — Тор и уравнения Лотки — Вольтерра сходное поведение? Реалистична ли модель

Ва—Тор? Каковы достоинства и недостатки каждого подхода? Указания о численных значениях параметров посмотрите в литературе из приведенного списка. Представляете ли вы, как можно было бы модифицировать модель Ва—Тор, чтобы смоделировать химическую реакцию?

18.2. ПЕРКОЛЯЦИЯ И ГАЛАКТИКИ

Помимо того что компьютер позволяет рассматривать более сложные нелинейные задачи, он усилил современную тему в физике, а именно объединяющую роль коллективного поведения. Иными словами, системы, образованные из множества составных частей, при определенных условиях проявляют одинаковые свойства, причем природа этих частей и вид взаимодействия их друг с другом могут различаться. Ниже мы приведем два примера коллективного поведения, относящиеся к областям эпидемиологии и структуры спиральных галактик. Изложение почти повторяет статью Шульмана и Сейдена (см. список литературы).

Рассмотрим воображаемую болезнь под названием *перколит*. Эта болезнь не вырабатывает никакого иммунитета, и продолжительность ее инкубационного периода и собственно заболевания составляет по 24 часа каждый. Предположим, что болезнь протекает в столь легкой форме, что страдающие ею в состоянии вступить в контакт со всеми людьми из своего окружения. В момент времени $t = 0$ один человек заражается болезнью от источника, находящегося вне данного круга лиц. Пусть все население составляет N человек, время t измеряется в днях, вероятность заразиться равна p и $n(t)$ — математическое ожидание числа больных в момент времени t . Убедитесь в том, что для $N = 1000$ и $p = 0.0005$ шанс обнаружить заразившегося этой болезнью неделю спустя исчезающе мал. С другой стороны, предположим, что $p = 0.002$. Тогда $n(t=1) = 2$, $n(t=2) \approx 4$ и можно гарантировать, что через некоторое время среднее число больных составит приблизительно 800 человек. Можете ли вы определить критическую вероятность p_c такую, что при $p < p_c$ среднее число больных равно нулю, а при $p \geq p_c$ не равно нулю?

Заметим, что мы не делали никаких предположений относительно того, какие именно лица заразятся. Модель перколита является примером простейшей перколяции, для которой возможно аналитическое решение. Какие модификации модели могли бы сделать ее более реалистичной? Скажутся ли эти модификации на коллективном поведении модели? Возможны ли полные аналитические решения в общем случае?

В завершение разговора о модели перколяции Шульмана и Сейдена мы приводим ниже листинг программы **galaxy** — реализацию модели на языке True BASIC — и призываем вас исследовать ее свойства. Основным фактором, позволяющим использовать модель перколяции для галактик, является распространяющееся формирование звезд. Область галактики может содержать необходимые ингредиенты для формирования звезды — молекулярный газ, надлежащие температуры и плотности, но будучи предоставлены сами себе, они ничего не образуют. Однако если через газ проходит ударная волна от сверхновой, то имеется много шансов для образования звезды. Сама сверхновая является результатом предшествующего образования по соседству с ней звезды. Поэтому мы вправе считать, что данная область галактики сходна с восприимчивым к перколиту индивидуумом — без источника никакого перколита нет. Вместо того чтобы определять, какие области обладают необходимыми условиями для образования звезды, мы суммируем всю неопределенность и изменчивость в одном параметре p , представляющем вероятность того, что взрыв сверхновой в одной области приведет к образованию звезды в соседней области.

Другое важное наблюдение, которое нам необходимо сделать в отношении спиральных галактик, состоит в том, что галактики не вращаются жестко (с постоянной угловой скоростью), а с хорошей точностью вращаются с постоянной круговой скоростью. Свойства случайного самораспространяющегося образования звезды и постоянной круговой скорости вводятся в программу **galaxy** следующим образом. Представим себе, что вся галактика разбита на концентрические кольца, которые в свою очередь делятся на ячейки одинаковой площади (рис. 18.1).

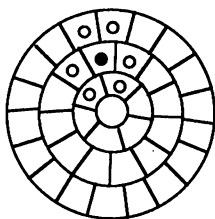


Рис. 18.1. Вид полярной сетки, применяемой в программе **galaxy**. Все ячейки имеют одинаковую площадь и в среднем по шесть ближайших соседей. Черным кружком обозначена активная область формирования звезды. На следующем шаге по времени она может вызвать образование звезд в ячейках, отмеченных белыми кружками. Заметим, что с течением времени соседи в смежных кольцах будут меняться за счет дифференциального вращения.



Рис. 18.2. Структура галактики, полученная программой **galaxy**. Параметры задачи: количество колец $N_r = 49$, начальное число активных ячеек $N_c = 500$, круговая скорость $v = 1.0$ (200 км/с), вероятность индуцированного образования звезды $p = 0.18$, шаг по времени $dt = 10$ (10^7 лет) и длительность варианта $tmax = 50$. (Проверьте свою программу для $N_r = 10$ и $N_c = 75$.)

Каждая ячейка соответствует области пространства размером с гигантское молекулярное облако и движется с одной и той же круговой скоростью v . Угловая скорость составляет $\omega = v/r$, где r — расстояние кольца от центра галактики. В момент времени t_n каждая ячейка является либо занятой (активной или пассивной), либо свободной. Занятая ячейка остается активной в течение только одного временного шага. В момент времени t_{n+1} соседи каждой активной ячейки активируются с вероятностью p . Дополнительные подробности моделирования показаны на рис. 18.1 и в программе **galaxy**. На рис. 18.2 приведен результат типичного моделирования галактики.

```
PROGRAM galaxy
! модель, предложенная Шульманом и Сейденом
DIM L(0 to 50,300)
DIM Lnew(0 to 50,300)
CALL parameter(Nr,Nc,v,dt,tmax,p)
CALL initial(Nr,Nc,L,Lnew)
DO
  CALL rotate(Nr,v,dt,L)
  CALL growth(Nr,p,L,Lnew)
  CALL spiral(Nr,L)
  LET t = t + dt
LOOP UNTIL t > tmax
END

SUB parameter (Nr,Nc,v,dt,tmax,p)
  INPUT prompt "число колец = " : Nr
  INPUT prompt "число исходных звездных скоплений = ": Nc
  INPUT prompt "круговая скорость = ": v
  INPUT prompt "вероятность роста = ": p
  INPUT prompt "шаг по времени = ": dt
  INPUT prompt "максимальное время = ": tmax
END SUB
```



```

SUB initial (Nr,Nc,L(,),Lnew(,))
  RANDOMIZE
  ! вычисляется начальное положение каждой ячейки
  FOR ir = 1 to Nr
    LET da = 2*pi/(6*ir)
    FOR ia = 1 to 6*ir
      LET L(ir,ia) = ia*da
    NEXT ia
  NEXT ir
  ! ячейки занимаются случайным образом
  FOR ic = 1 to Nc
    LET ix = int(Nr*rnd + 1)
    LET iy = int(Nr*rnd + 1)
    LET ir = sqr(ix*ix + iy*iy)
    IF ir <= Nr then
      LET ia = int(1 + 6*ir*rnd)
      LET L(ir,ia) = -L(ir,ia)      ! занятые ячейки отрицательны
      LET Lnew(ir,ia) = 1          ! активные ячейки
    END IF
  NEXT ic
END SUB

SUB rotate (Nr,v,dt,L(,))
  FOR ir = 1 to Nr
    ! v постоянна, а угловая скорость w меняется
    LET w = v/ir
    FOR ia = 1 to 6*ir
      LET anew = abs(L(ir,ia)) + w*dt
      LET anew = mod(anew,2*pi)      ! новый угол
      LET L(ir,ia) = sgn(L(ir,ia))*anew
    NEXT ia
  NEXT ir
END SUB

```

```

SUB growth (Nr, p, L(, ), Lnew(, ))
! Lnew - новая занятость
FOR ir = 1 to Nr
! максимальный острый угол между ближайшими соседними ячейками
LET acute = 2*pi/(5.95*ir)
! минимальный угол (между 180 и 360 град.)
LET reflex = 2*pi - acute
FOR ia = 1 to 6*ir      ! ia - номера ячеек
  IF Lnew(ir, ia) = 1 then
    LET Lnew(ir, ia) = 0 ! ячейка активна только один шаг по времени
    FOR jr = ir - 1 to ir + 1
      IF (jr > 0) and (jr <= Nr) then
        FOR ja = 1 to 6*jr
          IF L(jr, ja) > 0 then
            LET dis = abs(abs(L(ir, ia)) - abs(L(jr, ja)))
            IF (dis < acute) or (dis > reflex) then
              IF rnd < p then
                LET L(jr, ja) = -abs(L(jr, ja)) ! новая занятая ячейка
                LET Lnew(jr, ia) = 1           ! новая активная ячейка
              END IF
            END IF
          END IF
        NEXT ja
      END IF
    NEXT jr
  END IF
NEXT ia
NEXT ir
END SUB

```

```

SUB spiral (Nr,L(.))
  ! вывод результатов
  CLEAR
  SET window -85,85,-55,55
  FOR ir = 1 to Nr
    FOR ia = 1 to 6*ir
      IF L(ir,ia) < 0 then
        LET a = abs(L(ir,ia))
        LET x = ir*cos(a)
        LET y = ir*sin(a)
        FOR i = 1 to 30          ! изображаем занятые ячейки
          LET xp = x + 0.5 - rnd
          LET yp = y + 0.5 - rnd
          PLOT POINTS: xp,yp
        NEXT i
      END IF
    NEXT ia
  NEXT ir
END SUB

```

Разумеется, проведенный краткий разговор о галактиках не ставит целью убедить вас, что механизм, предложенный Шульманом и Сейденом, правилен. Скорее наша задача состоит в том, чтобы показать, как иная точка зрения может навести на мысль о новых подходах в других областях.

18.3. КАК КОМПЬЮТЕРЫ ВЛИЯЮТ СЕГОДНЯ НА ФИЗИКУ?

Наверное, нет необходимости убеждать вас в том, что компьютеры изменяют сегодня способ нашего восприятия физического мира. Вопрос «Как можно сформулировать данную задачу на компьютере?» уже заставил по-новому взглянуть на старые задачи и позволяет нам рассматривать новые.

Как повлияют компьютеры на обучение физике? На сегодняшний день наиболее распространенным применением компьютеров является оказание помощи студентам в понимании тем, которые проходились в курсе обучения в течение многих лет. До сих пор компьютер не изменил качественно ни метод учебы, ни сами изучаемые темы. Что произойдет, когда компьютеры станут столь же обычным явлением, как микрокалькуляторы?

Не заставят ли численное моделирование и вычислительные методы потесниться аналитические методы? Не превратятся ли таблицы интегралов в такой же анахронизм, как логарифмическая линейка? Должен ли математический анализ удерживать традиционно важное место в учебном плане? Следует ли считать, что мы «понимаем» природное явление, если в состоянии построить его вычислительную модель, которая позволяет осуществлять прогнозы, согласующиеся с экспериментом? Нужно ли по крайней мере получать хоть какие-то аналитические результаты? Какой вам видится роль компьютеров в обучении?

ЛИТЕРАТУРА

Мы не рассмотрели многие разделы физики и смежные области, в которых численное моделирование приобретает в настоящее время все большее значение. В дополнение к литературе, приведенной к концу каждой главы, ниже указаны некоторые дополнительные источники, относящиеся ко всей книге.

R. M. Anderson, Population ecology of infectious disease agents, pp.318—355. В кн.: *R. M. May*, ed., Theoretical Ecology: Principles and Applications, Blackwell, 1981.

J. W. Clark, J. Rafelski, J. V. Winston, Brain without mind: Computer simulation of neural networks with modifiable neuronal interactions, *Phys. Repts.*, **123**, 215 (1985).

M. Creutz, L. Jacobs, C. Rebbi, Monte Carlo computations in lattice gauge theory, *Phys. Repts.*, **95**, 201 (1983).

N. Cusack, The Physics of Structurally Disordered Matter, Taylor & Francis, 1987.

A. K. Dewdney, Computer Recreations, *Sci. Amer.* **251**, 14 (December, 1984). Обсуждается модель Ба—Тора.

Z. Fazarinc, A viewpoint on calculus, *Hewlett-Packard Journal* **38**, 38 (March, 1987).

R. P. Feynman, Robert B. Leighton, Matthew Sands, Lectures on Physics, Vol. II, Addison-Wesley, 1964. [Имеется перевод: Фейнмановские лекции по физике. — М.: Мир, 1967.] Мы позаимствовали название этой главы из подзаголовка гл. 12.1 «Одинаковые уравнения — одинаковые решения».

A. A. Harms, O. E. Hileman, Chemical clocks, feedback, and non-linear behavior, *Am. J. Phys.* **53**, 578 (1985). См. также Irving R. Epstein, *Patterns in Time and Space*, *Chem. & Eng. News* **65**, 24 (1987).

C. Jacobini, L. Reggiani, The Monte Carlo method for the solution of charge transport in semiconductors with applications to covalent materials, *Rev. Mod. Phys.* **55**, 645 (1983).

R. Peyret, T. D. Taylor, *Computational Methods for Fluid Flow*, Springer-Verlag, 1983.

D. E. Rumelhart, J. L. McClelland, *Parallel Distributed Processing: Explorations in the Microstructure of Cognition*, Vol.1: Foundations, MIT Press, 1986. Относительно приложений см. также Vol. 2.

L. S. Schulman, P. E. Seiden, Percolation and Galaxies, *Science* **233**, 425 (1986). Рассматривается также перколит. Авторы отмечают, что для конечных N заболеваемость в конце концов прекращается, но математическое ожидание времени исчезновения эпидемии растет экспоненциально с N .

J. D. Spain, *BASIC Microcomputer Models in Biology*, Addison-Wesley, 1982.

W. Williamson Jr., G. C. Duncan, Monte Carlo simulation of non-relativistic electron scattering, *Am. J. Phys.* **54**, 262 (1986).

ПРИЛОЖЕНИЯ



ПРИЛОЖЕНИЕ Е. УКАЗАТЕЛЬ ПРОГРАММ НА ЯЗЫКЕ TRUE BASIC: ЧАСТЬ 2

Приводится список номеров страниц с программами на языке True BASIC из ч. 2 книги.

Указатель программ моделирования на языке TRUE BASIC

Программа	Страница	Описание
integ	9	Вычисление одномерного интеграла методом прямоугольников
integ2	14	Двумерное численное интегрирование
random	17	Употребление генератора случайных чисел
random_walk	48	Одномерное случайное блуждание
random_walk2	52	Двумерное случайное блуждание
lattice_gas	60	Моделирование двумерного решеточного газа методом Монте-Карло
site	96	Рисование конфигураций ячеечной перколяции
continuum	101	Рисование конфигураций непрерывной перколяции
cluster	107	Маркировка кластеров ячеечной перколяции и расчет характеристик кластеров
rg	128	Визуализация ренорм-группы
single_cluster	144	Рост уединенного перколяционного кластера
Koch	149	Генерирование треугольной кривой Коха
invasion	156	Образование оккупирующего перколяционного кластера
box	180	Моделирование частицы в ящике методом Монте-Карло
entropy	186	Метод подсчета совпадений Ма
ideal_demon	199	Моделирование идеального газа в микроканоническом ансамбле методом Монте-Карло
Ising_demon	207	Моделирование одномерной модели Изинга методом Монте-Карло
conduct	213	Моделирование одномерной модели Изинга с тепловым потоком методом Монте-Карло
Boltzmann	226	Алгоритм Метрополиса для одномерной классической частицы
Ising	235	Алгоритм Метрополиса для двумерной модели Изинга
hard_disk	259	Алгоритм Метрополиса для жестких дисков
eigen	285	Вычисление энергетических уровней и собственных функций в одномерном случае
wavepacket	292	Демонстрация движения волнового пакета
qmwalk	303	Анализ уравнения Шредингера методом случайного блуждания
variation	310	Вариационный расчет основного состояния методом Монте-Карло
galaxy	327	Перколяционная модель формирования галактики

ПРИЛОЖЕНИЕ Ж. РАСПЕЧАТКИ ПРОГРАММ НА ЯЗЫКЕ ФОРТРАН:
ЧАСТЬ 2

Ниже приводятся распечатки перевода восьми программ, приведенных в ч. 2 текста, с языка True BASIC на Фортран-77. Программы пропускались на машине VAX 11/750. Поскольку язык Фортран разрабатывался в те годы, когда машинная графика еще не получила широкого распространения, то стандартный Фортран не содержит никаких графических инструкций и подпрограмм. В качестве примера использования графического пакета в Фортране в наших программах проиллюстрировано применение PLOT 10 (фирма Tektronix)—распространенного пакета, имеющегося на машине VAX и других компьютерах. На большинстве компьютеров PLOT 10 может также вызываться из Паскаля. Краткое описание графических подпрограмм из пакета PLOT 10, задействованных в наших программах, приведено в табл. Г1.

ГЛАВА 10

```
PROGRAM integ
*   вычисляется интеграл от  $f(x)$  в пределах от  $x = a$  до  $x = b$ 
CALL initial(a,b,h,n)
CALL rectangle(a,b,h,n,area)
CALL output(area)
STOP
END

SUBROUTINE initial(a,b,h,n)
*   a - нижний предел интегрирования, b - верхний предел
a = 0.0
b = 0.5*(3.14159)
WRITE(6,*) 'число интервалов = '
READ(5,*) n
h = (b - a)/n
RETURN
END
```



```

SUBROUTINE rectangle(a,b,h,n,area)
  x = a
  sum = 0
  DO 100 i = 0,n - 1
    sum = sum + f(x)
    x = x + h
100 CONTINUE
  area = sum*h
  RETURN
END

SUBROUTINE output(area)
  WRITE(6,13) area
13  FORMAT(2x, 'площадь = ',F12.7)
  RETURN
END

FUNCTION f(x)
  f = cos(x)
  RETURN
END

```

ГЛАВА 11

```

PROGRAM rwalk
*   моделирование Монте-Карло одномерного случайного блуждания
  DIMENSION prob(-64:64)
  CALL start(p,N,ntrial,iseed)
  DO 100 itrial = 1,ntrial
    CALL walk(ix,p,N,iseed)
*   сбор данных после N шагов
    CALL data(ix,xcum,x2cum,prob)
100 CONTINUE
  CALL aver(N,ntrial,xcum,x2cum,prob)
  STOP
END

```

```
SUBROUTINE start(p,N,ntrial,iseed)
WRITE(6,*) 'начальное случайное число (положительное целое) = '
READ(5,*) iseed
*   число испытаний
ntrial = 100
*   вероятность шага вправо
p = 0.5
WRITE(6,*) 'максимальное число шагов = '
READ(5,*) N
RETURN
END

SUBROUTINE walk(ix,p,N,iseed)
*   N-шаговое блуждание
*   начальное положение для каждого испытания
ix = 0
DO 100 istep = 1,N
  IF (ran(iseed).le.p) then
    ix = ix + 1
  ELSE
    ix = ix - 1
  ENDIF
100 CONTINUE
RETURN
END

SUBROUTINE data(ix,xcum,x2cum,prob)
DIMENSION prob(-64:64)
xcum = xcum + ix
x2cum = x2cum + ix*ix
prob(ix) = prob(ix) + 1
RETURN
END
```

```

SUBROUTINE aver(N,ntrial,xcum,x2cum,prob)
*   средние значения для N-шагового блуждания
  DIMENSION prob(-64:64)
  znorm = 1.0/ntrial
  xbar = xcum*znorm
  x2bar = x2cum*znorm
  DO 10 i = -N,N
    prob(i) = prob(i)*znorm
    WRITE(6,13) i,prob(i)
13   FORMAT(2x,i6,f10.5)
10  CONTINUE
  var = x2bar - xbar*xbar
  sigma = sqrt(var)
  WRITE(6,*) 'среднее смещение = ', xbar
  WRITE(6,*) 'sigma = ', sigma
  RETURN
END

```

ГЛАВА 12

```

PROGRAM percl
  DIMENSION lat(100,100),r(100,100),icl(100,100),np(5000)
  DIMENSION mass(5000)
  CALL start(L,p,ntrial,iseed)
  DO 100 itrial = 1,ntrial
35   CALL assign(L,r)
    CALL occupy(L,p,lat,r)
    CALL cluster(L,lat,icl,np)
*   проверка вертикального протекания
    CALL span(L,icl,np,1,ivert)
*   проверка горизонтального протекания
    CALL span(L,icl,np,2,ihori)
    IF ((ivert + ihori).eq.0) then
*   нет протекания ни в одном направлении
      ispan = 0
    ELSEIF ((ivert*ihori).eq.0) then
*   протекание только в одном направлении
      ispan = ivert + ihori

```

```

      IF (ivert.ne.0) spvert = spvert + 1
      IF(ihori.ne.0) sphori = sphori + 1
      ELSE
*       соединение в обоих направлениях
      IF (ivert.eq.ihori) then
        ispan = ivert
        spvert = spvert + 1
        sphori = sphori + 1
      ELSE
*       два соединяющих кластера --> это испытание пропускаем
      GO TO 35
      ENDIF
    ENDIF
*   вычисление массы каждого кластера
    CALL size(L,icl,np,mass,labmax)
*   вычисление среднего размера несоединяющих кластеров
    CALL mean(labmax,mass,ispan,clsize,sum)
    cltot = cltot + clsize
*   вычисление вероятности принадлежности соединяющему кластеру
    CALL Pinf(mass,ispan,sum,zPinf)
    zPtot = zPtot + zPinf
*   вычисление радиуса циркуляции наибольшего несоед. кластера
    CALL radius(L,icl,np,mass,labmax,ispan,radgyr)
    radtot = radtot + radgyr
100  CONTINUE
    CALL output(ntrial,spvert,sphori,cltot,zPtot,radtot)
    STOP
    END

    SUBROUTINE start(L,p,ntrial,iseed)
    WRITE(6,*) 'введите линейный размер решетки'
    READ(5,*) L
    WRITE(6,*) 'введите вероятность занятия ячейки'
    READ(5,*) p
    WRITE(6,*) 'enter seed, positive integer'
    READ(5,*) iseed
    WRITE(6,*) 'введите число испытаний'
    READ(5,*) ntrial
    RETURN
    END

```

```

SUBROUTINE assign(L,r)
*   присваивание каждой ячейке решетки случайного числа
  DIMENSION R(100,100)
  DO 10 i = 1,L
    DO 10 j = 1,L
      r(i,j) = ran(iseed)
10  CONTINUE
  RETURN
  END

SUBROUTINE occupy(L,p,lat,r)
*   занятие ячеек решетки
  DIMENSION lat(100,100),r(100,100)
  DO 10 i = 1,L
    DO 10 j = 1,L
      IF (r(i,j).le.p) then
        lat(i,j) = -1
      ELSE
        lat(i,j) = 0
      ENDIF
10  CONTINUE
  RETURN
  END

SUBROUTINE cluster(L,lat,icl,np)
*   маркировка ячеек
*   метки хранятся в массиве icl, массив правильных меток np
  DIMENSION lat(100,100),icl(100,100),np(5000)
  DO 10 k = 1,1000
    np(k) = 0
10  CONTINUE
  DO 15 i = 1,L
    DO 15 j = 1,L
      icl(i,j) = 0
15  CONTINUE
  nclust = 0
  IF (lat(1,1).lt.0) CALL newcl(nclust,icl,np,1,1)
  DO 20 i = 2,L
    IF (lat(i,1).lt.0) then
      ileft = i - 1

```

```

        IF (lat(ileft,1).lt.0) then
            icl(i,1) = icl(ileft,1)
        ELSE
            CALL newcl(nclust,icl,np,i,1)
        ENDIF
    ENDIF
20    CONTINUE
    DO 30 j = 2,L
        IF (lat(1,j).lt.0) then
            jdown = j - 1
            IF (lat(1,jdown).lt.0) then
                icl(1,j) = icl(1,jdown)
            ELSE
                CALL newcl(nclust,icl,np,1,j)
            ENDIF
        ENDIF
    DO 40 i = 2,L
        IF (lat(i,j).lt.0) then
            jdown = j - 1
            ileft = i - 1
            IF (icl(i,jdown) + icl(ileft,j).eq.0) then
                CALL newcl(nclust,icl,np,i,j)
            ELSE
                CALL neigh(lat,icl,np,i,j)
            ENDIF
        ENDIF
    ENDIF
40    CONTINUE
30    CONTINUE
    RETURN
END

SUBROUTINE newcl(nclust,icl,np,i,j)
*    иницируется новый кластер, указанный в подпрограмме cluster
    DIMENSION icl(100,100),np(5000)
    nclust = nclust + 1
    icl(i,j) = nclust
    np(nclust) = 0
    RETURN
END

```

```

SUBROUTINE neigh(la{,icl,np,i,j)
*   присваивание кластеру меток соседей
DIMENSION la{100,100),icl{100,100),np{5000)
jdown = j - 1
ileft = i - 1
IF (la{(i,jdown)*la{(ileft,j).gt.0) then
    CALL choice(icl,np,i,j,ileft,jdown)
    RETURN
ENDIF
IF (icl(i,jdown).gt.0) then
    icl(i,j) = icl(i,jdown)
    RETURN
ENDIF
icl(i,j) = icl(ileft,j)
RETURN
END

SUBROUTINE choice(icl,np,i,j,ileft,jdown)
*   присваивание кластеру меток при наличии выбора соседей
DIMENSION icl{100,100),np{5000)
IF (icl(ileft,j).eq.icl(i,jdown)) then
    icl(i,j) = icl(ileft,j)
ELSE
*   соседи имеют разные метки
    nleft = icl(ileft,j)
    ndown = icl(i,jdown)
    CALL proper(np,nleft)
    CALL proper(np,ndown)
    kmax = max0(nleft,ndown)
    kmin = min0(nleft,ndown)
    icl(i,j) = kmin
    IF (kmax.ne.kmin)np{kmax) = kmin
ENDIF
RETURN
END

```

```
SUBROUTINE proper(np, label)
*      определяется массив правильных меток np
DIMENSION np(5000)
10  IF (np(label).eq.0) RETURN
    label = np(label)
    GO TO 10
END

SUBROUTINE span(L, icl, np, idir, ispan)
*      выясняется существование соединяющего кластера проверкой
*      наличия ячеек с одинаковой правильной меткой на верхней и
*      нижней (idir = 1 - вертикальное протекание) или левой и
*      правой (idir  $\neq$  1 - горизонтальное протекание) границ
*      решетки. На выходе: ispan = метке соединяющего кластера
DIMENSION icl(100,100), np(5000)
DO 100 i1 = 1, L
    IF (idir.eq.1) then
        iarg = i1
        jarg = 1
    ELSE
        iarg = 1
        jarg = i1
    ENDIF
    IF (icl(iarg, jarg).gt.0) then
        n1 = icl(iarg, jarg)
        CALL proper(np, n1)
        DO 20 i2 = 1, L
            IF (idir.eq.1) then
                iarg = i2
                jarg = L
            ELSE
                iarg = L
                jarg = i2
            ENDIF
```



```

        IF (icl(iarg,jarg).gt.0) then
            n2 = icl(iarg,jarg)
            CALL proper(np,n2)
            IF (n1.eq.n2) then
                ispan = n1
                RETURN
            ENDIF
        ENDIF
20      CONTINUE
    ENDIF
100    CONTINUE
    ispan = 0
    RETURN
END

SUBROUTINE size(L,icl,np,mass,labmax)
DIMENSION icl(100,100),np(5000), mass (5000)
DO 10 k = 1,1000
    mass(k) = 0
10    CONTINUE
    labmax = 0
    DO 100 i = 1,L
        DO 100 j = 1,L
            IF (icl(i,j).gt.0) then
                label = icl(i,j)
                CALL proper(np,label)
                IF (label.gt.labmax) labmax = label
                mass(label) = mass(label) + 1
            ENDIF
        100    CONTINUE
    RETURN
END

```

```
SUBROUTINE mean(labmax, mass, ispan, clsize, sum)
*   вычисление среднего размера несоединяющих кластеров
DIMENSION mass( 5000)
sum = 0
sum2 = 0
DO 10 n = 1, labmax
    IF (n.ne.ispan) then
        sum = sum + mass(n)
        sum2 = sum2 + mass(n)**2
    ENDIF
10 CONTINUE
clsize = sum2/sum
RETURN
END

SUBROUTINE Pinf(mass, ispan, sum, zPinf)
*   вычисление вероятности принадлежности соединяющему кластеру
DIMENSION mass( 5000)
IF (ispan.eq.0) then
*   соединяющего кластера нет
    zPinf = 0.0
    RETURN
ENDIF
zPinf = mass(ispan)/(sum + mass(ispan))
RETURN
END

SUBROUTINE radius(L, icl, np, mass, labmax, ispan, radgyr)
*   расчет радиуса циркуляции наибольшего несоединяющего кластера
*   сначала находится наибольших несоединяющий кластер
DIMENSION icl(100,100), np( 5000), mass( 5000)
maxicl = 0
хсм = 0.0
усм = 0.0
r2 = 0.0
```

```

DO 50 k = 1, labmax
  IF ((mass(k).ne.0).and.(k.ne.ispan)) then
    IF (mass(k).gt.maxcl) then
      maxcl = mass(k)
      maxk = k
    ENDIF
  ENDIF
50 CONTINUE
*   вычисление g**2 и центра масс
DO 100 i = 1, L
  DO 100 j = 1, L
    IF (icl(i,j).gt.0) then
      label = icl(i,j)
      CALL proper(np, label)
      IF (label.eq.maxk) then
        xcm = xcm + i
        ycm = ycm + j
        r2 = r2 + i*i + j*j
      ENDIF
    ENDIF
  ENDIF
100 CONTINUE
*   вычисляем радиус циркуляции (вычитая центр масс)
r2cm = (xcm*xcm + ycm*ycm)/(maxcl*maxcl)
radgyr = sqrt(r2/maxcl - r2cm)
RETURN
END

SUBROUTINE output(ntrial, spvert, sphori, cltot, zPtot, radtot)
spvert = spvert/ntrial
sphori = sphori/ntrial
cltot = cltot/ntrial
zPtot = zPtot/ntrial
radtot = radtot/ntrial
WRITE(6,*) 'доля вертикального протекания = ', spvert
WRITE(6,*) 'доля горизонтального протекания = ', sphori
WRITE(6,*) 'средний размер кластера (несоединяющего) = ', cltot
WRITE(6,*) 'вероятность принадлежности соедин. кластеру = ', zPtot
WRITE(6,*) 'радиус циркуляции наибольшего кластера = ', radtot
RETURN
END

```

ГЛАВА 13

```
PROGRAM invasion
*   расчет оккупирующего перколяционного кластера, рисование
*   кластера, печать относительной занятости и P(r)
DIMENSION r(0:100,0:20), perx(500), perγ(500)
CALL setup(Lx, Ly, iseed)
CALL assign(Lx, Ly, r, perx, perγ, iseed)
CALL invade(perx, perγ, r, Lx, Ly)
CALL aver(Lx, Ly, r)
STOP
END

SUBROUTINE setup(Lx, Ly, iseed)
WRITE(6,*) 'Размер решетки в направлении γ = '
READ(5,*) Ly
WRITE(6,*) 'начальное случайное число = '
READ(5,*) iseed
Lx = 2*Ly
*   задание оконных координат
CALL initf(1200)
CALL dwindo(0.0, float(Lx+1), 0.0, float(Ly+2))
RETURN
END

SUBROUTINE assign(Lx, Ly, r, perx, perγ, iseed)
*   присваивание случайных чисел каждой ячейке
*   занятие первого столбца
*   ячейки второго столбца образуют начальный периметр
DIMENSION r(0:100,0:20), perx(500), perγ(500)
DO 200 j = 1, Ly
*   занимается первый столбец
r(1,j) = 1
CALL box(1,j)
*   присваивание случайных чисел остатку j-й строки
DO 100 i = 2, Lx
r(i,j) = ran(iseed)
100 CONTINUE
```

```

*      для ячеек периметра r(i,j) устанавливается больше 2
      r(2,j) = 2 + r(2,j)
*      sort perimeter sites
      CALL sort(perx,pery,r,2,j,j)
200  CONTINUE
      RETURN
      END

      SUBROUTINE sort(perx,pery,r,i0,j0,nper)
*      двоичная сортировка: список делится пополам; определяется, к
*      какой половине относится новое число; этот список тоже делится
*      пополам и определяется, в какой половине новое число. Процесс
*      продолжается до определения точного места нового числа
      DIMENSION r(0:100,0:20),perx(500),pery(500)
*      если только одна ячейка периметра
      IF (nper.eq.1) then
          perx(nper) = i0
          pery(nper) = j0
          RETURN
      ENDIF
*      если новая ячейка меньше всех предыдущих ячеек периметра
      i = perx(nper-1)
      j = pery(nper-1)
      IF (r(i0,j0).lt.r(i,j)) then
          perx(nper) = i0
          pery(nper) = j0
          RETURN
      ENDIF
*      начало сортировки Шелла, k2 - середина списка, k1,k3 - концы
      k1 = 1
      k3 = nper - 1
      k2 = (k1 + k3)/2
100  CONTINUE
*      цикл до отыскания точного положения
      i = perx(k2)
      j = pery(k2)
*      определение в какой половине списка находится новая ячейка

```

```

      IF (r(i0,j0).gt.r(i,j)) then
        k3 = k2
      ELSE
        k1 = k2
      ENDIF
*      новая середина
      k2 = (k1 + k3)/2
      IF ((k1.eq.k2).or.(k2.eq.k3)) then
*        точное место найдено и равно k3
*        сдвигаем все ячейки выше k3 на одну позицию вверх
        DO 10 kk = nper,k3+1,-1
          perx(kk) = perx(kk-1)
          pery(kk) = pery(kk-1)
10      CONTINUE
          perx(k3) = i0
          pery(k3) = j0
        RETURN
      ENDIF
      GO TO 100
      RETURN
      END

      SUBROUTINE sort2(perx,pery,r,i0,j0,nper)
*      стандартная сортировка методом вставок, элементы perx и pery с
*      наибольшим аргументом соответствуют наименьшему значению
      DIMENSION r(0:100,0:20),perx(500),pery(500)
      DO 200 k = 1,nper-1
        i = perx(k)
        j = pery(k)
        IF (r(i0,j0) .gt. r(i,j)) then
*          вставляем новую ячейку
          DO 100 kk = nper,k+1,-1
            perx(kk) = perx(kk-1)
            pery(kk) = pery(kk-1)
100      CONTINUE

```

```

        perx(k) = i0
        pery(k) = j0
        RETURN
    ENDIF
200  CONTINUE
*   новая ячейка меньше всех предыдущих ячеек периметра
    perx(k) = i0
    pery(k) = j0
    RETURN
    END

SUBROUTINE invade(perx,pery,r,Lx,Ly)
*   массивы nnx and nny указывают положение ближайших соседей
*   относительно каждой ячейки
    DIMENSION r(0:100,0:20),perx(500),pery(500)
    DIMENSION nnx(4),nny(4)
    DATA nnx/1,-1,0,0/
    DATA nny/0,0,1,-1/
    nper = Ly
    DO 1000 idummy = 1,5000
        i = perx(nper)
        j = pery(nper)
        nper = nper - 1
*   занятые ячейки имеют значения между 1 и 2
        r(i,j) = r(i,j) - 1
        CALL box(i,j)
        DO 100 nn = 1,4
*   находим новые ячейки периметра
            nx = i + nnx(nn)
            ny = j + nny(nn)
*   периодические краевые условия по y
            IF (ny .gt. Ly) then
                ny = 1
            ELSEIF (ny.lt.1) then
                ny = Ly
            ENDIF
*   новая ячейка периметра

```

```

        IF (r(nx,ny).lt.1) then
            r(nx,ny) = r(nx,ny) + 2
            nper = nper + 1
            CALL sort(perx,pery,r,nx,ny,nper)
        ENDIF
100    CONTINUE
*      выход по достижении кластером правой границы
        IF (i.gt.Lx) RETURN
1000  CONTINUE
        RETURN
        END

SUBROUTINE aver(Lx,Ly,r)
*      нахождение доли занятых ячеек и распределения вероятности
*      занятия для средней половины решетки
        DIMENSION r(0:100,0:20)
        DIMENSION p(0:20),ns(0:20)
        Lmin = Lx/3
        Lmax = 2*Lmin
*      число ячеек в средней половине
        n = (Lmax - Lmin + 1)*Ly
        DO 200 i = Lmin,Lmax
            DO 100 j = 1,Ly
                iarg = 20*(amod(r(i,j),1.0))
                ns(iarg) = ns(iarg) + 1
                IF ((r(i,j) .gt. 1).and.(r(i,j) .lt. 2) ) then
                    occupied = occupied + 1
*      плотность вероятности P(r)
                    p(iarg) = p(iarg) + 1
                ENDIF
            DO 100 CONTINUE
        DO 200 CONTINUE
*      для вывода результатов в файл по каналу 9 ждем нажатия клавиши
        fraction = occupied/n
        CALL finitt(1,1)
        WRITE(9,*) 'доля оккупированных ячеек = ',fraction
        DO 300 i = 0,20
            rr = i/20.0
            IF (ns(i) .gt. 0) WRITE(9,*) rr, p(i)/ns(i)
        DO 300 CONTINUE

```



```

RETURN
END

SUBROUTINE box(i,j)
*   вычерчивание прямоугольника, состоящего из 11 полос
  x0 = i - 0.5
  y0 = j - 0.5
  DO 10 k = 0,10
    y = y0 + k*0.1
    CALL movea(x0,y)
*   проводим прямую относительно x0,y
    CALL drawr(1.0,0.0)
10  CONTINUE
    CALL tsend
  RETURN
  END

```

ГЛАВА 14

```

PROGRAM entropy
*   вычисляется энтропия методом подсчета совпадений Ma
  DIMENSION mleft(10),mright(10),micro(0:2000)
  CALL start(nl,nr,mleft,mright,micro,nexch,iseed)
*   обмен частиц
  CALL exch(nl,nr,nexch,mleft,mright,micro,iseed)
*   расчет частоты совпадений и энтропии
  CALL output(nexch,micro)
  STOP
  END

```

```
SUBROUTINE start(nl,nr,mleft,mright,micro,nexch,iseed)
*   ввод параметров и выбор начальной конфигурации частиц
  DIMENSION mleft(10),mright(10),micro(0:2000)
  WRITE(6,*) 'полное число частиц = '
  READ(5,*) N
  WRITE(6,*) 'число частиц слева = '
  READ(5,*) nl
  WRITE(6,*) 'начальное случайное число (целое положительное) = '
  READ(5,*) iseed
*   число частиц с правой стороны
  nr = N - nl
  micro(0) = 0
  DO 100 il = 1,nl
*       список номеров частиц с левой стороны
    mleft(il) = il
*       начальное микросостояние
    micro(0) = micro(0)*2 + 2
100  CONTINUE
  DO 200 ir = 1,nr
*       список номеров частиц с правой стороны
    mright(ir) = ir + nl
200  CONTINUE
  WRITE(6,*) 'число обменов = '
  READ(5,*) nexch
  RETURN
END

SUBROUTINE exch(nl,nr,nexch,mleft,mright,micro,iseed)
*   обмен частицы слева с номером ileft
*   с частицей справа с номером iright
  DIMENSION mleft(10),mright(10),micro(0:2000)
  DO 100 iexch = 1,nexch
*       индексы массивов выбираем случайным образом
    ileft = int(ran(iseed)*nl + 1)
    iright = int(ran(iseed)*nr + 1)
    jleft = mleft(ileft)
    jright= mright(iright)
```

```

*      новый номер частицы в левом массиве
      mleft(ileft) = jright
*      новый номер частицы в правом массиве
      mright(iright) = jleft
*      определяем новое микросостояние
      micro(iexch) = micro(iexch - 1) + 2**jright
      micro(iexch) = micro(iexch) - 2**jleft
100  CONTINUE
      RETURN
      END

      SUBROUTINE output(nexch,micro)
*      вычисляются частота совпадений и энтропия
*      полное число сравнений
      DIMENSION micro(0:2000)
      ncomp = nexch*(nexch - 1)/2
*      сравниваем микросостояния
      DO 200 iexch = 1,nexch - 1
          DO 100 jexch = iexch + 1,nexch
              IF (micro(iexch).eq.micro(jexch)) ncoin = ncoin + 1
100      CONTINUE
200  CONTINUE
*      частота совпадений
      rate = float(ncoin)/float(ncomp)
      IF (rate.gt.0) S = alog(1.0/rate)
      WRITE(6,*) 'оценка энтропии = ', S
      RETURN
      END

```

ГЛАВА 15

```

PROGRAM conduct
*   алгоритм демона для одномерной модели Изинга
*   подвод и отвод тепла предусмотрен в концевых узлах
DIMENSION s(1000),edemon(1000),edcum(1000),zmgcum(1000)
CALL start(N,nmcs,s,zl,iseed,nheat)
DO 100 imcs = 1,nmcs
    DO 50 i = 1,N-2
        istep = (imcs - 1)*(N-2) + i
        IF (mod(istep,nheat).eq.0) CALL heat(N,zl,s,edemon)
        CALL change(N,edemon,s,zl,accept,iseed)
50    CONTINUE
        CALL data(N,s,edemon,zmgcum,edcum)
100   CONTINUE
CALL averg(N,nmcs,zmgcum,edcum,accept)
STOP
END

SUBROUTINE start(N,nmcs,s,zl,iseed,nheat)
*   ввод параметров с клавиатуры, запись на устройство 9
DIMENSION s(1000)
WRITE(6,*) 'число спинов = '
READ(5,*) N
WRITE(9,*) 'число спинов = ',N
WRITE(6,*) 'число шагов Монте-Карло на спин = '
READ(5,*) nmcs
WRITE(9,*) 'число шагов Монте-Карло на спин = ',nmcs
WRITE(6,*) 'константа обменного взаимодействия = '
READ(5,*) zl
WRITE(9,*) 'константа обменного взаимодействия = ',zl
WRITE(6,*) 'начальное случайное число = '
READ(5,*) iseed
WRITE(9,*) 'начальное случайное число = ',iseed
WRITE(6,*) 'время между изменением концевых спинов = '
READ(5,*) nheat
WRITE(9,*) 'время между изменением концевых спинов = ',nheat

```

```

DO 100 i = 1,N
*      задание случайной начальной конфигурации
      IF (ran(iseed).gt.0.5) then
          s(i) = 1
      ELSE
          s(i) = -1
      ENDIF
100    CONTINUE
      RETURN
      END

SUBROUTINE change(N,edemon,s,zl,accept,iseed)
DIMENSION s(1000),edemon(1000)
*      вычисляется случайный спин от 2-го до (N-1)-го
ispin = int(ran(iseed)*(N-2) + 2)
*      динамика опрокидывания спина, пробное изменение энергии = de
de = 2*zl*s(ispin)*(s(ispin-1) + s(ispin+1))
IF (de.le.edemon(ispin)) then
    s(ispin) = -s(ispin)
    accept = accept + 1
    edemon(ispin) = edemon(ispin) - de
ENDIF
RETURN
END

SUBROUTINE data(N,s,edemon,zmgcum,edcum)
*      накопление данных
DIMENSION s(1000),edemon(1000),edcum(1000),zmgcum(1000)
DO 10 i = 1,N
    edcum(i) = edcum(i) + edemon(i)
    zmgcum(i) = zmgcum(i) + s(i)
10    CONTINUE
    edemon(1) = 0
    edemon(n) = 0
    RETURN
    END

```

```

SUBROUTINE averg(N,nmcs,zmgcum,edcum,accept)
*   нормирование накопленных данных и печать результатов
DIMENSION zmgcum(1000),edcum(1000)
znorm = 1.0/nmcs
accept = accept*znorm/(N-2)
WRITE(6,*) 'коэффициент принятия = ',accept
WRITE(9,*) 'коэффициент принятия = ',accept
*   величина edcum для спина 1 (или N) равна скорости поступления
*   (отвода) тепла
WRITE(6,*) 'скорость подвода тепла = ',edcum(1)*znorm
WRITE(6,*) 'скорость отвода тепла = ',edcum(N)*znorm
DO 10 i = 2,N-1
    edave = edcum(i)*znorm
    IF (edave.ne.0.0) temp = 4.0/alog(1.0 + 4.0/edave)
    zmag = zmgcum(i)*znorm
    WRITE(6,13) i,edave,temp,zmag
    WRITE(9,13) i,edave,temp,zmag
13   FORMAT(1x,i4,3(2x,f13.5))
10  CONTINUE
RETURN
END

SUBROUTINE heat(N,zJ,s,edemon)
*   производится попытка добавить (отвести) тепло у спина 1 (N)
DIMENSION s(1000),edemon(1000)
IF ((s(1)*s(2).gt.0).and.(s(N)*s(N-1).lt.0)) then
    edemon(1) = edemon(1) + 2*zJ
    s(1) = -s(1)
    edemon(N) = edemon(N) - 2*zJ
    s(N) = -s(N)
ENDIF
RETURN
END

```

ГЛАВА 16

```

PROGRAM Ising2
*   алгоритм Метрополиса для двумерной модели Изинга
DIMENSION spin(32,32),w(-4:4)
CALL start(N,L,T,nmcs,spin,E,M,w,iseed)
DO 100 imcs = 1,nmcs
    CALL Metrop(N,L,spin,E,M,w,ratio,iseed)
    CALL data(E,M,ecum,e2cum,mcum,m2cum)
100 CONTINUE
CALL output(N,nmcs,ecum,e2cum,mcum,m2cum,ratio)
STOP
END

SUBROUTINE start(N,L,T,nmcs,spin,E,M,w,iseed)
DIMENSION spin(32,32),w(-4:4)
WRITE(6,*) 'линейный размер решетки = '
READ(5,*) L
*   число спинов
N = L*L
WRITE(6,*) 'число шагов Монте-Карло на спин = '
READ(5,*) nmcs
WRITE(6,*) 'приведенная температура = '
READ(5,*) T
WRITE(6,*) 'начальное случайное число = '
READ(5,*) iseed
*   случайная начальная конфигурация
DO 200 j = 1,L
    DO 100 i = 1,L
        IF (ran(iseed) .lt. 0.5) then
            spin(i,j) = 1
        ELSE
            spin(i,j) = -1
        ENDIF
*   суммарная намагниченность
M = M + spin(i,j)
100 CONTINUE
200 CONTINUE

```

```

*      вычисление начальной энергии E
DO 300 j = 1,L
    IF (j .eq. L) then
        iup = 1
    ELSE
        iup = j + 1
    ENDIF
    DO 400 i = 1,L
        IF (i .eq. L) then
            irect = 1
        ELSE
            irect = i + 1
        ENDIF
        sum = spin(i,iup) + spin(irect,j)
*      полная энергия
        E = E - spin(i,j)*sum
400    CONTINUE
300    CONTINUE
*      вычисление вероятностей перехода
*      индекс массива w равен сумме спинов четырех соседей
e4 = exp(-4.0/T)
e8 = e4*e4
*      аргумент функции exp = изменению энергии
w(4) = e8
w(-4) = e8
w(2) = e4
w(-2) = e4
RETURN
END

SUBROUTINE Metrop(N,L,spin,E,M,w, ratio, iseed)
DIMENSION spin(32,32),w(-4 : 4)
DO 100 ispin = 1,N
*      случайный разыгрыш координат спина
    i = int(L*ran(iseed) + 1)
    j = int(L*ran(iseed) + 1)
*      находим значения соседних спинов
*      периодические краевые условия
    CALL periodic(i,j,L,spin, isum)

```



```

        IF ( spin(i,j)*isum .le. 0 ) then
            CALL accept(i,j,M,E,isum,spin,ratio)
        ELSEIF ( ran(iseed) .lt. w(isum) ) then
            CALL accept(i,j,M,E,isum,spin,ratio)
        ENDIF
100  CONTINUE
    RETURN
END

SUBROUTINE periodic(i,j,L,spin,isum)
*   находится сумма значений соседних спинов;
*   используются периодические краевые условия
    DIMENSION spin(32,32)
    IF ( i .eq. 1 ) then
        left = spin(L,j)
    ELSE
        left = spin(i - 1,j)
    ENDIF
    IF ( i .eq. L ) then
        ight = spin(1,j)
    ELSE
        ight = spin(i + 1,j)
    ENDIF
    IF ( j .eq. 1 ) then
        idown = spin(i,L)
    ELSE
        idown = spin(i,j - 1)
    ENDIF
    IF ( j .eq. L ) then
        iup = spin(i,1)
    ELSE
        iup = spin(i,j + 1)
    ENDIF
    isum = left + ight + iup + idown
    RETURN
END

```

```
SUBROUTINE accept(i,j,M,E,ism,spin, ratio)
DIMENSION spin(32,32)
spin(i,j) = -spin(i,j)
ratio = ratio + 1
M = M + 2*spin(i,j)
E = E - 2*spin(i,j)*ism
RETURN
END
```

```
SUBROUTINE data(E,M,ecum,e2cum,mcum,m2cum)
*   накапливание данных после каждого шага Монте-Карло на спин
ecum = ecum + E
e2cum = e2cum + E*E
mcum = mcum + M
m2cum = m2cum + M*M
RETURN
END
```

```
SUBROUTINE output(N,nmcs,ecum,e2cum,mcum,m2cum,ratio)
*   средние на спин
znorm = 1/float(nmcs*N)
ratio = ratio*znorm
eave = ecum*znorm
e2ave = e2cum*znorm
save = mcum*znorm
s2ave = m2cum*znorm
WRITE(6,*) 'коэффициент принятия = ',ratio
WRITE(6,*) 'средняя энергия на спин = ',eave
WRITE(6,*) 'средний квадрат энергии на спин = ',e2ave
WRITE(6,*) 'средняя намагниченность = ', save
WRITE(6,*) 'средний квадрат намагниченности = ',s2ave
RETURN
END
```

ГЛАВА 17

```

PROGRAM wp
*   график движения квантовомеханического волнового пакета
  DIMENSION ar(-100:100), ai(-100:100), prob(-100:100)
  DIMENSION prl(-100:100, -100:100), pim(-100:100, -100:100)
  CALL param(zk0, width, x0, xmin, h, dk, dt, tmax, m, N)
  CALL packet(width, x0, m, dk, ar, ai)
*   CALL free(m, N, zk0, dk, h, ar, ai, prl, pim)
  CALL barrie(m, N, zk0, dk, h, ar, ai, prl, pim)
  ntime = tmax/dt
  DO 10 itime = 0, ntime
    t = itime*dt
    CALL move(t, m, N, zk0, dk, h, prl, pim, prob)
    CALL plot(prob, xmin, xmax, N, h)
10  CONTINUE
*   конец рисования, выставление курсора в позицию 1,1
  CALL finitt(1,1)
  STOP
  END

SUBROUTINE param(zk0, width, x0, xmin, h, dk, dt, tmax, m, N)
  WRITE(6,*) 'максимальное значение x = '
  READ(5,*) xmax
  WRITE(6,*) 'минимальное значение x = '
  READ(5,*) xmin
  WRITE(6,*) 'размер шага = '
  READ(5,*) h
*   N = количество вычисляемых координат
  N = (xmax - xmin)/h
  WRITE(6,*) 'центральный волновой вектор = '
  READ(5,*) zk0
  WRITE(6,*) 'число волновых векторов = '
  READ(5,*) m
  WRITE(6,*) 'ширина волнового пакета в x-пространстве = '
  READ(5,*) width
  WRITE(6,*) 'начальная координата пакета = '
  READ(5,*) x0

```

```
      dk = 2.0/(width*m)
      m = m/2
      N = N/2
      WRITE(6,*) 'шаг по времени = '
      READ(5,*) dt
      WRITE(6,*) 'полное время = '
      READ(5,*) tmax
      WRITE(6,*) 'максимальное значение квадрата psi = '
      READ(5,*) ymax
      CALL initt(1200)
      CALL dwindo(xmin,xmax,-0.1,ymax)
      RETURN
      END

      SUBROUTINE packet(width,x0,m,dk,ar,ai)
*      вычисляются коэффициенты Фурье
      DIMENSION ar(-100:100),ai(-100:100)
      sum2 = 0.0
      DO 100 ik = -m,m
          dzk = ik*dk
          a = exp(-dzk*dzk*width*width/4.0)
          ar(ik) = a*cos(-dzk*x0)
          ai(ik) = a*sin(-dzk*x0)
          sum2 = sum2 + ar(ik)*ar(ik) + ai(ik)*ai(ik)
100    CONTINUE
      znorm = dk*sqrt(1.0/(sum2))
      DO 200 ik = -m,m
          ar(ik) = ar(ik)*znorm
          ai(ik) = ai(ik)*znorm
200    CONTINUE
      RETURN
      END
```

```

SUBROUTINE free(m,N,zk0,dk,h,ar,ai,prl,pim)
DIMENSION prl(-100:100,-100:100),pim(-100:100,-100:100)
DIMENSION ar(-100:100),ai(-100:100)
DO 100 ik = -m,m
    zk = zk0 + ik*dk
    DO 50 ix = -N,N
        arg = zk*ix*dx
        fr = cos(arg)
        fi = sin(arg)
        prl(ik,ix) = fr*ar(ik) - fi*ai(ik)
        pim(ik,ix) = fr*ai(ik) + fi*ar(ik)
50    CONTINUE
100  CONTINUE
RETURN
END

SUBROUTINE move(t,m,N,zk0,dk,h,prl,pim,prob)
DIMENSION prl(-100:100,-100:100),pim(-100:100,-100:100)
DIMENSION prob(-100:100)
DO 100 ix = -N,N
    psir = 0.0
    psii = 0.0
    DO 10 ik = -m,m
        zk = zk0 + ik*dk
        arg = -0.5*zk*zk*t
        fr = cos(arg)
        fi = sin(arg)
        *    вещественная и мнимая части psi
        psir = psir + fr*prl(ik,ix) - fi*pim(ik,ix)
        psii = psii + fi*prl(ik,ix) + fr*pim(ik,ix)
10    CONTINUE
    prob(ix) = psir*psir + psii*psii
100  CONTINUE
RETURN
END

```

```

SUBROUTINE barrie(m, N, zk0, dk, h, ar, ai, prl, pim)
DIMENSION ar(-100:100), ai(-100:100)
DIMENSION prl(-100:100, -100:100), pim(-100:100, -100:100)
*   вычисление  $a(k)(\exp(ik1x) + b \exp(-ik1x))$  для  $x > 0$ 
*   и  $a(k)*c*\exp(ik2x)$  для  $x < 0$ 
WRITE(6,*) 'высота барьера = '
READ(5,*) V0
DO 100 ik = -m, m
    zk1 = zk0 + ik*dk
    E = 0.5*zk1*zk1
    IF (E.gt.V0) then
        zk2 = sqrt(2.0*(E - V0))
        br = (zk1 - zk2)/(zk1 + zk2)
        bi = 0.0
        cr = 2.0*zk1/(zk1 + zk2)
        ci = 0.0
    ELSE
        zk2 = sqrt(2.0*(V0 - E))
        denom = zk1*zk1 + zk2*zk2
        br = (zk1*zk1 - zk2*zk2)/denom
        bi = 2.0*zk1*zk2/denom
        cr = 2.0*zk1*zk1/denom
        ci = -2.0*zk1*zk2/denom
    ENDIF
    DO 40 ix = -Nx, 0
        arg = zk1*ix*h
        cs = cos(arg)
        si = sin(arg)
*       далее использовано  $\sin(-kx) = -\sin(kx)$  и  $\cos(-kx) = \cos(kx)$ 
        fr = cs + br*cs + bi*si
        fi = si + bi*cs - br*si
        prl(ik, ix) = fr*ar(ik) - fi*ai(ik)
        pim(ik, ix) = fr*ai(ik) + fi*ar(ik)
40    CONTINUE

```

```

        DO 60 ix = 1,N
            arg = zk2*ix*h
            IF (E.gt.V0) then
                cs = cos(arg)
                si = sin(arg)
            ELSE
                cs = exp(-arg)
                si = 0.0
            ENDIF
            fr = cr*cs - ci*si
            fi = ci*cs + cr*si
            prl(ik,ix) = fr*ar(ik) - fi*ai(ik)
            pim(ik,ix) = fr*ai(ik) + fi*ar(ik)
60      CONTINUE
100     CONTINUE
        RETURN
        END

        SUBROUTINE plot(prob,xmin,N,h)
*       стирание экрана
        CALL newpag
*       нанесение делений
        DO 10 i = -N,N
            x = i*h
            CALL movea(x,0.0)
            CALL drawa(x,0.01*h)
10      CONTINUE
*       проведение горизонтальной прямой, после цикла x = xmax
        CALL movea(xmin,0.0)
        CALL drawa(x,0.0)
*       построение графика квадрата волновой функции
        CALL movea(-N*h,prob(-N))
        DO 20 i = -N,N
            x = i*h
            CALL drawa(x,prob(i))
20      CONTINUE
        RETURN
        END

```

ПРИЛОЖЕНИЕ 3. РАСПЕЧАТКИ ПРОГРАММ НА ЯЗЫКЕ ПАСКАЛЬ:
ЧАСТЬ 2

Ниже приводятся версии перевода восьми программ из ч. 2 данной книги с языка True BASIC на Паскаль. Программы пропускались с использованием компилятора Macintosh Pascal. Для других версий Паскаля необходимо внести минимальные изменения. Применяемые в программах на Паскале графические процедуры входят в состав стандартного программного обеспечения компьютера Macintosh и могут вызываться из других языков программирования, реализованных на этом компьютере. Краткое описание графических процедур для компьютера Macintosh приведено в табл. Д1.

ГЛАВА 10

```
program integ (input, output);
(* вычисляется интеграл от f(x) в пределах от x = a до x = b *)
var
  a, b, h, area : real;
  n : integer;

  procedure initial (var a, b, h : real;
                    var n : integer);
  begin
    a := 0.0;                                (* нижний предел *)
    b := 0.5 * 3.14159;                      (* верхний предел *)
    write('число интервалов = ');
    readln(n);
    h := (b - a) / n                         (* шаг интегрирования *)
  end;

  function f (x : real) : real;
  begin
    f := cos(x)
  end;

  procedure rectangle (a, b, h : real;
                     n : integer;
                     var area : real);
  var
    x, sum : real;
```



```
i : integer;
begin
  x := a;
  sum := 0.0;
  for i := 0 to n - 1 do
    begin
      sum := sum + f(x);
      x := x + h
    end;
  area := sum * h
end;

procedure outdat (area : real);
begin
  writeln('площадь = ', area : 12 : 7)
end;

begin                                     (* основная программа *)
  initial(a, b, h, n);
  rectangle(a, b, h, n, area);
  outdat(area)
end.
```

ГЛАВА 11

```
program random_walk (input, output);
(* моделирование Монте-Карло одномерного случайного блуждания *)
const
  xmax = 64;
type
  list = array[-64..64] of real;
var
  probab : list;
  p : real;                                     (* вероятность шага вправо *)
  x, xcum, x2cum, itrial, ntrial, nstep : integer;
```

```
procedure initial (var p : real;
                  var nstep, ntrial, xcum, x2cum : integer;
                  var prob : list);

var
  i : integer;
begin
  ntrial := 100;                (* число испытаний *)
  p := 0.5;                    (* вероятность шага вправо *)
  write('максимальное число шагов = ');
  readln(nstep);
  write('начальное случайное число (отрицательное целое) = ');
  readln(iseed);
  (* начальное обнуление данных *)
  xcum := 0;
  x2cum := 0;
  for i := -nstep to nstep do
    prob[i] := 0
  end;

function rnd : real;
(* выдается случайное число от 0 до 1 с использованием средств *)
(* компьютера Macintosh. Функция random выдает случайное число *)
(* от -32768 до 32767. Более хорошие генераторы случайных чисел *)
(* можно найти в книге Пресса У. и др., Численные рецепты *)
begin
  rnd := (random + 32768.0)/(32768.0 + 32767.0)
end;

procedure walk (var x : integer;
               p : real;
               nstep : integer);

var
  istep : integer;
begin
  x := 0;    (* начальное положение для каждого испытания *)
  for istep := 1 to nstep do
    if rnd <= p then
      x := x + 1
    else
      x := x - 1
    end;
  end;
```

```

procedure data (var x, xcum, x2cum : integer;
                var prob : list);
begin
    xcum := xcum + x;
    x2cum := x2cum + x * x;
    prob[x] := prob[x] + 1
end;

procedure average (nstep, ntrial, xcum, x2cum : integer;
var prob : list);
(* средние значения для nstep-шагового блуждания *)
var
    norm, xbar, x2bar, sigma, variance, probability : real;
    x : integer;
begin
    norm := 1.0 / ntrial;
    xbar := xcum * norm;
    x2bar := x2cum * norm;
    for x := -nstep to nstep do
        begin
            probability := prob[x] * norm;
            writeln(x : 6, probability : 10 : 5)
        end;
    variance := x2bar - xbar * xbar;
    sigma := sqrt(variance);
    writeln('среднее смещение = ', xbar : 10 : 5);
    writeln('sigma = ', sigma : 10 : 5)
end;

begin
    (* основная программа *)
    initial(p, nstep, ntrial, xcum, x2cum, prob);
    for itrial := 1 to ntrial do
        begin
            walk(x, p, nstep);
            data(x, xcum, x2cum, prob) (* сбор данных после nstep шагов *)
        end;
    average(nstep, ntrial, xcum, x2cum, prob)
end.

```

ГЛАВА 12

```
program site_perc (input, output);
(* рисуются конфигурации ячеечной перколяции *)
const
    boxlength = 180;
    marginy = 10;
    marginx = 100;
type
    list = array[1..50, 1..50] of real;
var
    L : integer;
    r : list;

    procedure initial (var L : integer);
    begin
        write('линейный размер решетки = ');
        readln(L);
        (* рисование ячеек решетки *)
        framerect(margin, margin, boxlength + margin, boxlength + margin)
    end;

    function rnd : real;
    (* выдается случайное число от 0 до 1 с помощью функции *)
    (* random, которая выдает случайное число от -32768 до 32767 *)
    (* Более хорошие генераторы случайных чисел можно найти в *)
    (* книге Пресса У. и др., Численные рецепты *)
    begin
        rnd := (random + 32768.0)/(32768.0 + 32767.0)
    end;

    procedure lattice (L : integer;
                      var r : list);
    (* каждой ячейке придается случайное число и рисуется решетка *)
    var
        col, row, x, y : integer;
        scale : real;
    begin
        scale := boxlength / L;
        for row := 1 to L do
```

```
            (* рисование ячеек решетки *)
```

```

begin
  γ := round((row - 0.5) * scale) + marginγ;
  (* с каждой ячейкой связывается квадрат со стороной 1*)
  for col := 1 to L do
    begin
      x := round(scale * (col - 0.5)) + marginx;
      (* каждой ячейке придается случайное число *)
      r[col, row] := rnd;
      (* в каждой ячейке рисуется точка *)
      moveTo(x, γ);
      lineTo(x, γ)
    end
  end
end;

procedure configuration (L : integer;
                        var r : list);
(* занятие ячеек с данной вероятностью p *)
var
  s : list;
  p, scale : real;
  row, col, x, γ, size : integer;
begin
  for row := 1 to L do
    for col := 1 to L do
      s[row,col] := 0;
    scale := boxlength / L;
    while p <= 1.0 do
      begin
        write(' вероятность p = ');
        readln(p);
        (* половина стороны квадрата для занятых ячеек *)
        size := round(0.5 * scale);
        for row := 1 to L do
          begin
            γ := round(scale * (row - 0.5)) + marginγ;
            for col := 1 to L do
              if (r[col, row] < p) and (s[col, row] <> 1.0) then

```

```
begin      (* новые занятые ячейки *)
  x := round(scale * (col - 0.5)) + marginx;
  paintrect(y - size, x - size, y + size, x + size);
  s[col, row] := 1.0      (* ячейку заняли *)
end
end
end
end;

begin      (* основная программа *)
  initial(L);
  (* каждой ячейке придается случайное число и рисуется решетка *)
  lattice(L, r);
  configuration(L, r)      (* занятие ячеек с данной p *)
end.
```

ГЛАВА 13

```
program single_cluster (input, output);
(* вычисление фрактальной размерности перколяционного кластера *)
(* с помощью алгоритма Хаммерсли-Лиса-Александровица *)
type
  list = array[1..200] of integer;
var
  L : integer;
  p : real;
  s : list;

  procedure parameter (var L : integer;
                      var p : real);
begin
  write('максимальный радиус кластера = ');
  readln(L);
  write('вероятность занятия ячейки = ');
  readln(p)
end;
```

```
function rnd : real;
(* вычисление случайного числа от 0 до 1 *)
begin
    rnd := (random + 32768.0) * (1.5259e-5)
end;

procedure grow (L : integer;
                p : real;
                var s : list);
(* расчет перколяционного кластера *)
var
    i, k, x, y, xn, yn, nper, nn, r : integer;
    r2 : real;
    px, py : array[1..5000] of integer;
    lat : array[-50..50, -50..50] of integer;
    nx, ny : array[1..4] of integer;
(* в массивах nx и ny указаны положения ближайших соседей *)
(* относительно каждой ячейки *)
begin
    for r := 1 to 2*L do
        s[r] := 0;
        for x := -L to L do
            for y := -L to L do
                lat[x,y] := 0;
            lat[0, 0] := 1;
            nx[1] := 1;
            nx[2] := -1;
            nx[3] := 0;
            nx[4] := 0;
            ny[1] := 0;
            ny[2] := 0;
            ny[3] := 1;
            ny[4] := -1;
            for i := 1 to 4 do
                begin
                    px[i] := nx[i];
                    py[i] := ny[i]
                end;
            end;
        end;
```

```

nper := 4;          (* начальное число ячеек периметра *)
repeat
  k := 1 + trunc(rnd * nper); (* случайный выбор ячейки периметра *)
  x := px[k];
  y := py[k];
  if (rnd < p) then
    begin
      (* site tested and occupied *)
      lat[x, y] := 1;
      r2 := x * x + y * y;
      r := round(sqrt(r2)); (* расстояние от начала координат *)
      s[r] := s[r] + 1;      (* число ячеек на расстоянии r *)
      (* вновь занятая ячейка заменяется в массивах px и py *)
      (* последней ячейкой в списке ячеек периметра *)
      px[k] := px[nper];
      py[k] := py[nper];
      nper := nper - 1;
      (* отыскание новых ячеек периметра *)
      for nn := 1 to 4 do
        begin
          xn := x + nx[nn];
          yn := y + ny[nn];
          if ((lat[xn, yn] = 0) and (abs(xn) <= L) and (abs(yn) <= L)) then
            begin
              nper := nper + 1;
              px[nper] := xn;
              py[nper] := yn;
            end
          end
        end
      else
        begin
          (* ячейка проверяется, но не занимает *)
          lat[x, y] := -1;
          px[k] := px[nper];
          py[k] := py[nper];
          nper := nper - 1;
        end
      until (nper < 1) (* пока не будут проверен весь периметр *)
    end;
end;

```



```

procedure plotg (L : integer;
                 s : list);
const
    (* минимальные используемые экранные координаты *)
    y0 = 190;
    x0 = 50;
var
    xmax, ymax, scalex, scaley, hashx, hashy : integer;
    x, y, r, xcoor, ycoor, mass : integer;
begin
    mass := s[1];
    xmax := trunc(ln(L) + 1);
    ymax := trunc(ln(4 * L * L) + 1);
    scalex := 300 div xmax;
    scaley := 190 div ymax;
    hashy := 10;
    hashx := 15;
    (* вычерчивание осей *)
    moveto(x0, y0 - ymax * scaley);
    lineto(x0, y0);
    lineto(x0 + xmax * scalex, y0);
    (* маркировка осей *)
    moveto(250, 270);
    drawstring('ln(r)');
    moveto(3, 80);
    drawstring('ln(M)');
    (* нанесение делений на осях *)
    for x := 1 to xmax do
        begin
            xcoor := x * scalex + x0;
            moveto(xcoor, y0);
            lineto(xcoor, y0 - hashy)
        end;
    for y := 1 to ymax do
        begin
            ycoor := y0 - scaley * y;
            moveto(x0, ycoor);
            lineto(x0 + hashx, ycoor)
        end;
    end;
end;

```

```
(* рисование графика ln(заключенной массы) от ln(r) *)
for r := 2 to L do
  begin
    mass := mass + s[r];
    x := x0 + trunc(scalex * ln(r));
    y := trunc(y0 - scaley * ln(mass));
    paintrect(y - 1, x - 1, y + 1, x + 1)
  end
end;

begin                                     (* основная программа *)
  parameter(L, p);
  grow(L, p, s);
  plot(L, s)
end.
```

ГЛАВА 14

```
program entropy (input, output);
(* вычисляется энтропия методом подсчета совпадений Ma *)
type
  list = array[1..10] of integer;
  state = array[0..2000] of integer;
var
  micro : state;
  right, left : list;
  nl, nr, nexch : integer;
```

```
procedure start (var nl, nr, nexch : integer;
                var left, right : list;
                var micro : state);
(* ввод параметров и выбор начальной конфигурации частиц *)
var
    ir, il, N : integer;
begin
    write('полное число частиц = ');
    readln(N);
    write('число частиц слева = ');
    readln(nl);
    nr := N - nl;          (* число частиц с правой стороны *)
    micro[0] := 0;
    for il := 1 to nl do
        begin
            left[il] := il;  (* список номеров частиц с левой стороны *)
            micro[0] := micro[0] * 2 + 2  (* начальное микросостояние *)
        end;
    for ir := 1 to nr do
        right[ir] := ir + nl;  (* список частиц с правой стороны *)
    end;
    write('число обменов = ');
    readln(nexch);
end;

function rnd : real;
(* формируется случайное число в интервале от 0 до 1 *)
begin
    rnd := (random + 32768.0) / (32768.0 + 32767.0);
end;

function power (n : integer) : integer;
(* выдается 2 в степени n *)
var
    i, p : integer;
begin
    p := 1;
    for i := 1 to n do
        p := p * 2;
    end;
    power := p;
end;
```

```
procedure exch (var nl, nr, nexch : integer;
               var left, right : list;
               var micro : state);
(* обмен частицы слева с номером ileft *)
(* с частицей справа с номером iright *)
var
  iexch, ileft, iright, jleft, jright : integer;
begin
  for iexch := 1 to nexch do
    begin
      (* индексы массивов выбираем случайным образом *)
      ileft := trunc(rnd * nl + 1);
      iright := trunc(rnd * nr + 1);
      jleft := left[ileft];
      jright := right[iright];
      left[ileft] := jright; (* новый номер частицы слева *)
      right[iright] := jleft; (* новый номер частицы справа *)
      (* определяем новое микросостояние *)
      micro[iexch] := micro[iexch - 1] + power(jright);
      micro[iexch] := micro[iexch] - power(jleft)
    end
  end;
end;
```

```

procedure outdat (nexch : integer;
                  var micro : state);
(* вычисляются частота совпадений и энтропия *)
(* полное число сравнений *)
var
  iexch, jexch, ncomp, ncoin : integer;
  rate, S : real;
begin
  ncomp := nexch * (nexch - 1) div 2;
  ncoin := 0;
  (* сравниваем микросостояния *)
  for iexch := 1 to nexch - 1 do
    for jexch := iexch + 1 to nexch do
      if (micro[iexch] = micro[jexch]) then
        ncoin := ncoin + 1;
    rate := ncoin / ncomp;          (* частота совпадений *)
    if (rate > 0) then
      S := ln(1.0 / rate);
    writeln('оценка энтропии = ', S)
  end;

begin
  (* основная программа *)
  start(nl, nr, nexch, left, right, micro);
  exch(nl, nr, nexch, left, right, micro);      (* обмен частиц *)
  outdat(nexch, micro)  (* расчет частоты совпадений и энтропии *)
end.

```

ГЛАВА 15

```

program ideal_demon (input, output);
(* алгоритм демона для одномерного идеального классического газа *)
type
  list = array[1..100] of real;
var
  vel : list;
  N, nmcs, imcs, i : integer;
  esystem, edemon, vtot, dvmax, escum, vcum, edcum, accept : real;

```

```
procedure initial (var N, nmcs : integer;
                  var esystem, edemon, vtot, dvmax : real;
                  var vel : list);
var
  i : integer;
  vinitial : real;
begin
  write('число частиц = ');
  readln(N);
  write('число шагов Монте-Карло = ');
  readln(nmcs);
  write('начальная энергия системы = ');
  readln(esystem);
  edemon := 0; (* энергия демона *)
  write('максимальное изменение скорости = ');
  readln(dvmax);
  vinitial := sqr(2 * esystem / N);
  (* у всех частиц одинаковые начальные скорости *)
  vtot := 0.0;
  for i := 1 to N do
    begin
      vel[i] := vinitial;
      vtot := vtot + vinitial (* полная скорость системы *)
    end
  end;

function rnd : real;
begin
  rnd := (random + 32768.0) / (32768.0 + 32767.0)
end;

procedure changes (N : integer;
                  var esystem, edemon, vtot, dvmax, accept : real;
                  var vel : list);
var
  dv, vtrial, de : real;
  ip : integer;
```

```

begin
  dv := (2 * rnd - 1) * dvmax;      (* пробное изменение скорости *)
  ip := trunc(rnd * N + 1);         (* берем случайную частицу *)
  vtrial := vel[ip] + dv;           (* пробная скорость *)
  de := 0.5 * (vtrial * vtrial - vel[ip] * vel[ip]); (* изменение энергии *)
  if (de <= edemon) or (de <= 0) then
    begin
      vel[ip] := vtrial;
      vtot := vtot + dv;             (* полная скорость системы *)
      accept := accept + 1;
      edemon := edemon - de;
      esystem := esystem + de
    end
  end;

procedure data (var esystem, edemon, vtot, escum, vcum, edcum : real);
begin
  edcum := edcum + edemon;
  escum := escum + esystem;
  vcum := vcum + vtot
end;

procedure averages (N, nmcs : integer;
                    escum, vcum, edcum, accept : real);
var
  norm, edave, vave, esave : real;
begin
  norm := 1.0 / (nmcs * N);
  edave := edcum * norm;             (* средняя энергия демона *)
  accept := accept * norm;           (* коэффициент принятия *)
  (* средние системы на частицу *)
  norm := norm / N;
  esave := escum * norm;             (* средняя энергия на частицу системы *)
  vave := vcum * norm;              (* средняя скорость на частицу системы *)
  writeln('средняя энергия демона = ', edave : 10 : 5);
  writeln('средняя энергия системы на частицу = ', esave : 10 : 5);
  writeln('средняя скорость на частицу = ', vave : 10 : 5);
  writeln('коэффициент принятия = ', accept : 10 : 5)
end;

```

```
begin                                     (* основная программа *)
  initial(N, nmcs, esystem, edemon, vtot, dvmax, vel);
  accept := 0.0;
  escum := 0.0;
  vcum := 0.0;
  edcum := 0.0;
  for imcs := 1 to nmcs do
    for i := 1 to N do
      begin
        changes(N, esystem, edemon, vtot, dvmax, accept, vel);
        (* накапливаем данные после каждого испытания *)
        data(esystem, edemon, vtot, escum, vcum, edcum)
      end;
    averages(N, nmcs, escum, vcum, edcum, accept)
  end.
```

ГЛАВА 16

```
program Ising2 (input, output);
(* алгоритм Метрополиса для двумерной модели Изинга *)
type
  lattice = array[1..32, 1..32] of integer;
  list = array[-4..4] of real;
var
  spin : lattice;
  w : list;
  N, L, imcs, nmcs : integer;
  T, E, M, ratio, ecum, e2cum, mcum, m2cum : real;

  function rnd : real;
  (* формируется случайное число в интервале от 0 до 1 *)
  begin
    rnd := (random + 32768.0) / (32768.0 + 32767.0)
  end;

  procedure initial (var N, L, nmcs : integer;
    var spin : lattice;
    var T, E, M : real;
    var w : list);
```



```

var
  i, j, up, right, sum : integer;
  e4, e8 : real;
begin
  write('линейный размер решетки = ');
  readln(L);
  N := L * L;           (* число спинов *)
  write('число шагов Монте-Карло на спин = ');
  readln(nmcs);
  write('приведенная температура = ');
  readln(T);
  (* случайная начальная конфигурация *)
  for j := 1 to L do
    for i := 1 to L do
      begin
        if (rnd < 0.5) then
          spin[i, j] := 1
        else
          spin[i, j] := -1;
        M := M + spin[i, j]  (* суммарная намагниченность *)
      end;
    (* вычисление начальной энергии E *)
    for j := 1 to L do
      begin
        if (j = L) then
          up := 1
        else
          up := j + 1;
        for i := 1 to L do
          begin
            if (i = L) then
              right := 1
            else
              right := i + 1;
            sum := spin[i, up] + spin[right, j];
            E := E - spin[i, j] * sum  (* полная энергия *)
          end
        end;
      end;
    (* вычисление вероятностей перехода *)
    (* индекс массива w равен сумме спинов четырех соседей *)

```

```
e4 := exp(-4.0 / T);
e8 := e4 * e4;
w[4] := e8;
w[-4] := e8;
w[2] := e4;
w[-2] := e4
end;

procedure pbc (var i, j, L, sum : integer;
               var spin : lattice);
(* находится сумма значений соседних спинов; *)
(* используются периодические краевые условия *)
var
    left, right, up, down : integer;
begin
    if (i = 1) then
        left := spin[L, j]
    else
        left := spin[i - 1, j];
    if (i = L) then
        right := spin[1, j]
    else
        right := spin[i + 1, j];
    if (j = 1) then
        down := spin[i, L]
    else
        down := spin[i, j - 1];
    if (j = L) then
        up := spin[i, 1]
    else
        up := spin[i, j + 1];
    sum := left + right + up + down
end;
```

```

procedure accept (i, j, sum : integer;
                 var spin : lattice;
                 var ratio, M, E : real);

begin
    spin[i, j] := -spin[i, j];
    ratio := ratio + 1;
    M := M + 2 * spin[i, j];
    E := E - 2 * spin[i, j] * sum
end;

procedure Metropolis (N, L : integer;
                    var spin : lattice;
                    var E, M, ratio : real;
                    var w : list);

var
    ispin, i, j, sum : integer;
begin
    for ispin := 1 to N do
        begin
            (* случайный разыгрыш координат спина *)
            i := trunc(L * rnd + 1);
            j := trunc(L * rnd + 1);
            (* находим значения ближайших соседних спинов *)
            (* периодические краевые условия *)
            pbc(i, j, L, sum, spin);
            if (spin[i, j] * sum <= 0) then
                accept(i, j, sum, spin, ratio, M, E)
            else if (rnd < w[sum]) then
                accept(i, j, sum, spin, ratio, M, E)
            end
        end
    end;

procedure data (var E, M, ecum, e2cum, mcum, m2cum : real);
(* накапливание данных после каждого шага Монте-Карло на спин *)
begin
    ecum := ecum + E;
    e2cum := e2cum + E * E;
    mcum := mcum + M;
    m2cum := m2cum + M * M
end;

```

```

procedure outdat (N, nmcs : integer;
                  ecum, e2cum, mcum, m2cum, ratio : real);
var
    norm, eave, e2ave, save, s2ave : real;
begin
    norm := 1.0 / (nmcs * N);
    (* средние на спин *)
    ratio := ratio * norm;
    eave := ecum * norm;
    e2ave := e2cum * norm;
    save := mcum * norm;
    s2ave := m2cum * norm;
    writeln('коэффициент принятия := ', ratio);
    writeln('средняя энергия на спин := ', eave);
    writeln('средний квадрат энергии на спин := ', e2ave);
    writeln('средняя намагниченность := ', save);
    writeln('средний квадрат намагниченности := ', s2ave)
end;

begin
    (* основная программа *)
    initial(N, L, nmcs, spin, T, E, M, w);
    for imcs := 1 to nmcs do
        begin
            Metropolis(N, L, spin, E, M, ratio, w);
            data(E, M, ecum, e2cum, mcum, m2cum)
        end;
        outdat(N, nmcs, ecum, e2cum, mcum, m2cum, ratio)
    end.

```

ГЛАВА 17

```

program eigen (input, output);
var
    V0, a, xmax, dx, x0, y0, xscale, yscale : real;

    procedure parameter (var V0, a, xmax, dx : real);
    begin
        write('глубина ямы = ');
        readln(V0);
        write('полуширина ямы = ');

```

```

    readln(a);
    write('максимальное выводимое на график значение x = ');
    readln(xmax);
    write('размер шара dx = ');
    readln(dx);
end;

procedure plot_potential (V0, a, xmax : real;
                           var xscale, yscale, x0, y0 : real);
(* задание масштаба графика и построение потенциала*)
begin
    xscale := 512 / (2.0 * xmax);
    yscale := 300 / (2.0 * 10.0);
    x0 := xscale * xmax;
    y0 := yscale * 10.0;
    moveto(trunc(x0 - xmax * xscale, trunc(y0 + 6 * yscale)));
    lineto(trunc(x0 - a * xscale, trunc(y0 + 6 * yscale)));
    lineto(trunc(x0 - a * xscale, trunc(y0 + 9 * yscale)));
    lineto(trunc(x0 + a * xscale, trunc(y0 + 9 * yscale)));
    lineto(trunc(x0 + a * xscale, trunc(y0 + 6 * yscale)));
    lineto(trunc(x0 + xmax * xscale, trunc(y0 + 6 * yscale)));
end;

function V (x, V0, a : real) : real;
(* вычисляется функция потенциала V(x) *)
begin
    if x > a then
        V := V0
    else
        V := 0.0
    end;
end;

procedure Euler (V0, a, dx, xmax, xscale, yscale, x0, y0 : real);
var
    i, j, parity : integer;
    x, phi, dphi, d2phi, E : real;
begin
    write('четная или нечетная функция (1 или -1) ');
    readln(parity);

```

```
repeat
  write('E = ');
  readln(E);
  (* задание начальных значений при x = 0 *)
  if parity = -1 then
    begin
      phi := 0.0;
      dphi := 0.0
    end
  else
    begin
      phi := 1.0;
      dphi := 0.0
    end;
  x := 0.0;
  repeat (* вычисление волновой функции *)
    x := x + dx;
    d2phi := 2.0 * phi * (V(x, V0, a) - E);
    dphi := dphi + d2phi * dx;
    phi := phi + dphi * dx;
    (* plot wave function *)
    i := trunc(x0 + xscale * x);
    j := trunc(y0 + yscale * phi);
    moveto(i,j);
    lineto(i,j);
    i := trunc(x0 - xscale * x);
    j := trunc(y0 + yscale * phi * parity);
    moveto(i,j);
    lineto(i,j);
  until (x > xmax) or (abs(phi) > 10);
until E = 0 (* для выхода из программы введите E = 0 *)
end;

begin (* основная программа *)
  parameter(V0, a, xmax, dx);
  plot_potential(V0, a, xmax, xscale, yscale, x0, y0);
  Euler(V0, a, dx, xmax, xscale, yscale, x0, y0)
end.
```

ПРЕДМЕТНЫЙ УКАЗАТЕЛЬ: ЧАСТИ I И 2¹⁾

Автокорреляционная функция скорости 171, 172, 187

Автомат клеточный 167—172

Алгоритм 16, 29

— устойчивость 39—41, 79, 131; 288

Анализ численный 14; 80, 330

Ансамбль канонический 201—204, 222—278

— микроканонический 194—217

Антиферромагнетизм 205, 254—256

Аппроксимации погрешность 40

Астрономические единицы 75, 76, 78, 92

Аттрактор 188, 189, 196, 206, 208

Баланс детальный 224, 233

Баня тепловая 221

Биения 220

Бимана алгоритм, см. Интегрирование численное

Био—Савара закон 263, 264

Бокса—Мюллера метод 30

Больцмана распределение 203, 204, 211, 222—230

Бифуркация 188, 193, 198

Ван-дер-Ваальса уравнение 167; 264

Вариационный принцип 308—318, см. *Ферма* принцип

Ва—Тор модель 324

Верле алгоритм, см. Интегрирование численное

Вероятностей распределение

— гауссово, см. Гауссово распределение

— дисперсия 22, 30

— интегральное 27, 28

— момент 64

— неоднородное 26—29

— плотность 26

— равномерно отклоняющееся 27

— стандартное отклонение 22—25

Вириал 161, 162, 180

Волновая функция 281

Волновое уравнение 227

Волновой пакет 290—300

Волны свойства

— групповая скорость 230

— движение 226—231

— дисперсия 230, 231

— период 229

— фазовая скорость 230

Восприимчивость магнитная 234, 243, 254, 255, 276, 278

Выборка по значимости 26, 31, 32, 34

¹⁾Номера страниц, набранные жирным шрифтом, относятся к ч. 1, а набранные светлым шрифтом — к ч. 2. — *Прим. перев.*

Газ 165, 167; 256, 271

— решеточный 60—64, 252, 253

Гауссово распределение 28, 29—34, 51, 66, 67, 80, 230, 291

Геометрическая оптика 257—265

График в двойном логарифмическом масштабе 81, 96—98, 104

Графика 18, 19, 42—47, 109—112, 146

Групповая скорость, см. Волновые свойства

Гюйгенса принцип 235

Давление 160—162, 180; 257, 258, 264

Динамика молекулярная 144—179; 281

Дисперсия, см. Вероятностей распределение, Волны свойства

Дифференциальные уравнения

— — дифракции 231—236

— — диффузии 171—174, 275; 45, 55, 60, 66, 160—164, 253, 281, 300, 301

— — порядок 132—135

— — метод решения Бимана 135, 136, 140

— — — — Верле 133, 135, 140, 146, 147, 157

— — — — полушага 132, 133

— — — — предиктор-корректора 136

— — — — Рунге — Кутты 136, 137

— — — — самостартующие 133

— — — — средней точки 132

— — — — Эйлера 27—29, 36, 41, 55, 59, 103, 131, 132, 146

— — — — Кромера 55, 59, 65, 102, 103, 132, 146, 205, 216, 218; 283

Диэлектрика пробой 165

Длина волны 229

Жидкость 168—170; 265, 266, 269, 270

Жесткие сферы 258—267

Задача о коробейнике 271—274

— — коммивояжере, см. Задача о коробейнике

— — разорении игрока 58, 59

Игра «Жизнь» 170

Идеальный газ 198—201, 230, 231

Изинга модель 204—217, 231—256

— — динамика спинного обмена 231

— — — опрокидывания спина 205, 234

Интегрирование численное

— — аппроксимация в средней точке 11, 12

— — Бимана 135, 136, 140

— — Верле 133, 134, 140, 146, 147, 153, 157

— — многомерное 12—15

— — Монте-Карло 15—26

— — оценки погрешности 37, 38

— — полушага 132, 133

— — порядок 132—135

— — предиктор-корректора 136

— — Рунге — Кутты 136, 137

— — самостартующее 133

— — средней точки 132

— — формула прямоугольников 7, 9—11, 11, 13

— — — Симпсона 8—12, 20

— — — трапеций 7—12

— — *Эйлера* 27—29, 37, 41, 54, 59, 103, 131, 132, 146
— — *Эйлера—Кромера* 54, 59, 65, 102, 103, 132, 146, 205, 216, 218; 283
Интерференция 231—236

Кеплера законы 70, 80—82

Кирхгофа правило 120

Ковер, *Серпинского* прокладка 151, 152

Конденсатор 120—129, 276, 277

Корреляции время 188, 240, 241

— длина 243

Корреляционная функция парная 258, 259, 263

Коха кривая 148—152

Краевые условия периодические 147—151, 168; 63, 154, 238, 261

Лапласа уравнение 269—279; 166, 167

—фрактальные ковры 172

Леннарда—Джонса потенциал 146, 147, 175; 267, 268

Лоренца сила 252

Лотки—Вольтерра уравнение 321, 322

Лучепреломление двойное 240

Магнитная (электронная) линза 268, 269

Магнитное поле 263—269

Макросостояние 156; 176, 194

Максвелла—Больцмана распределение 166, 175

Масса приведенная 70, 71

Масштабирование 198—201; 51, 125, 148

—конечномерное 122, 123, 247—250, 255

Маятник 105, 108, 202—207

Метрополиса алгоритм 30—35, 222—242, 245—255, 258—271, 273, 274

Микросостояние 176, 194

Мираж 248

Моделирование 14—20; 80, 280, 330

Моды нормальные 218—223

Молния 165, 166

Момент импульса (момент количества движения) 70

Монте-Карло метод

— — — анализ погрешности 20—26, 65

— — — для интегрирования 15—20

— — — канонического ансамбля 201—204, 206—217

— — — микроканонического ансамбля 197—204, 206—217

— — — полимеров 67—76

— — — случайного блуждания, см. Ренормгруппа

— — — частицы в ящике 178—191

Морза осциллятор 138

—потенциал 137, 138; 315

Наименьших квадратов метод 86—90

Начальные условия 152, 153, 156, 157, 166

Ньютона закон всемирного тяготения 52, 63, 71, 72, 75—78

— второй 50, 64, 72, 82, 91, 100, 104, 202, 215; 280

— остывания 26, 39

— третий 153, 154

Обратного преобразования (обратных функций) метод 27—30, 66

— — —необратимый 157

Округления погрешность 38, 135, 188

- Осциллятор гармонический
 - —затухающий 112, 113
 - —квантовый 288, 307, 312
 - —математический маятник 100—105
 - —нелинейный 105—108
 - —связанный 215—223
 - —совершающий вынужденные колебания 113—120
- Отбора-отказа метод 41
- Отжиг модельный 272, 273
- Отклонение стандартное 21—25
 - —среднего 21—25, 39, 40
- Отношение характеристическое 78
- Отображения нелинейные 188, 195—217
 - —графический анализ 191—195
 - —двумерные 207, 208
 - —орбиты 184, 188
 - —показатели 198—201
 - —порядок 196
 - —стандартное квадратичное отображение 183—201
 - —универсальные свойства 199—202
- Переноса свойства 183—187
- Периметра ячейки 143, 153, 154, 162, 165
- Периода удвоение 188, 189—195, 206, 207
- Перколяция 92—137, 324—327
 - в непрерывном случае 100, 101
 - длина связности 119—124, 131, 132
 - кластер 93, 97, 98, 103—118, 142—148
 - конечномерное масштабирование 123, 124
 - оккупирующая 154—161
 - перколит 324
 - показатели 121—124, 131—137, 142, 143
 - порог 93, 97, 98, 124, 126, 127, см. также Фазовый переход
 - протекание 93, 94, 98, 131, 134, 147
 - ренормгруппа 198; 125—137, 143
 - структура галактики 324—330
 - цепная 99
- Пластика в четверть длины волны 241
- Поле электрическое 236, 237, 252—263, 277
 - —силовые линии 252—257, 262
- Полимеры 67—75
- Поляризация 236—241
- Потенциал электрический 260—263, 269—279
- Предиктор-корректора метод, см. Интегрирование численное
- Преобразования символьные 14
- Приближение к равновесию 155—158
- Прицельный параметр 259, 260
- Программирования языки 19, 20
- Пуассона уравнение 276—279
- Пуанкаре отображение 203—206
- Равновесие 157, 165, 168
- Рассеяние 259, 260
- Резонанс 117—119, 126—129, 219
- Рекурсия 193, 194; 149
- Релаксации метод 272, 273; 165
- Ренормгруппа 198; 125—137, 143
- Рефракция 257—265

Решетка квадратная 53, 277
RC-цепь 42, 132—137
RCL-цепь 130, 131, 134, 135, 201

Системы нелинейные 17, 105—108, 120, 182—208

Скорость установившаяся 54, 60

Случайное блуждание 44—90, 300—308

- без самопересечений 68—73
- время первого прохода 58
- диффузия 45, 60
- истинное без самопересечений 73—75
- метод репаций 71—73
- перебор 46, 47, 163, 164
- персистентное 57
- с ловушками 58, 76—78
- среднеквадратичное смещение 54, 67

Случайные последовательности 152, 153; 17, 18, 26, 79—84

- линейный конгруэнтный генератор 80

Снелла закон 248

Собственная функция 282

Собственное значение 282

Сортировка 157, 158, 160

Состояние метастабильное 252

Спад критический 250

Сплав бинарный 278

Суперпозиция 119, 120, 219—221, 230

Температура 26, 159, 160; 201—204, 219, 222

Теплоемкость удельная 160, 167

Теплопроводность удельная 216

Тормозящая сила 52, 53, 62—67, 83, 112

Точка неподвижная 188, 191—193, 196—201, 210; 126—132

Точность 39—41

Треугольная решетка 168, 169; 54, 55, 99, 216, 255

True BASIC 20, 21

- BOX AREA 43, 44
- BOX CIRCLE 77, 78, 111, 153
- BOX CLEAR 43, 44
- BOX ELLIPSE 43
- BOX KEEP 109—111, 153
- BOX LINES 43, 44
- BOX SHOW 109—111, 153
- CALL 30
- CLEAR 43, 44
- CLOSE 109
- DATA 163
- DECLARE DEF 45
- DEF 45
- DIM 75
- DO-LOOP 56
- END 31
- END IF 46
- END SUB 30
- FLOOD 43, 79
- FOR-NEXT 30, 31
- GET KEY 85, 86
- IF-THEN-ELSE 46
- INPUT 163; 240

- —INPUT PROMPT 36, 37
- —INT 18
- —KEY INPUT 85, 86
- —LET 30
- —массивы 75 —78
- —OPEN 108, 109; 240
- —ORD 85, 86
- —PAUSE 43
- —PLOT 45
- —PLOT LINES 43
- —PLOT POINTS 42, 43
- —PLOT TEXT 43, 46
- —Подпрограмма 30 —35
- —PRINT 30, 109, 162; 264
- —PRINT USING 46, 109
- —PROGRAM 31
- —RANDOMIZE 17
- —READ 163
- —RND 151; 17, 18
- —SELECT CASE 243, 244
- —SET BACK 43, 44
- —SET COLOR 43, 44
- —SET WINDOW 42, 43, 78
- —SGN 62
- —SUB 30
- —TRUNCATE 190; 261
- —UNTIL 56
- —WHILE 56

Универсальность 199 —201; 68, 125

Управление в реальном времени 14, 15

Уравнение логистическое 183

—состояния 167, 168

Фазовая скорость, см. Волны свойства

Фазовое пространство 104, 105, 203

Фазовый переход 118, 119, 243, 244, 250, 251, 255, 256, 267

— —геометрический 94

— —магнитный 242 —256

— —параметр порядка 243

Ферма принцип 241 —248

Ферромагнетизм 204, 233, 243

Фоккера —Планка уравнение 78

Фурье-анализ 223 —226

Хаос 189, 190, 197, 202, 203; 190, 191

Центральная предельная теорема 64, 66

—сила 73, 74

Цепи электрические 120 —129

Частиц орбиты

— —в поле, не пропорциональном обратному квадрату 82, 83, 84, 90

— — — —электрическом 258 —260

— — —пространстве скоростей 88 —90

— —возмущенные 83 —88, 90, 93, 94

- —круговые 73, 80
- —прецессирующие 82
- —эллиптические 73, 74, 81

Шредингера уравнение 280—308

- —нестационарное 281, 289—300
- —стационарное 283—288, 300, 301

Эдена модель 153—158

Эйлера—Кромера алгоритм, см. Интегрирование численное

Эйнштейна соотношение 171; 161

Эксцентриситет 74, 75

Энергии сохранение 71, 103, 131, 135, 157

Энтропия 157; 182—190

Эргодическая гипотеза 198

Яма потенциальная прямоугольная 285—289, 307, 313

ОГЛАВЛЕНИЕ

ГЛАВА 10.	Численное интегрирование	5
10.1.	Простые одномерные методы численного интегрирования	6
10.2.	Числовой пример	8
10.3.	Численное интегрирование многих интегралов	12
10.4.	Вычисление интегралов простейшим методом Монте-Карло	15
10.5.	Вычисление многомерных интегралов методом Монте-Карло	18
10.6.	Анализ погрешности метода Монте-Карло	20
10.7.	Неравномерные распределения вероятностей	26
10.8.	Выборка по значимости	30
10.9.	Методы случайного блуждания	32
	Литература	35
	Дополнительная литература	36
	Приложение 10А. Оценки погрешностей численного интегрирования	37
	Приложение 10Б. Аналитический вывод стандартного отклонения от среднего	39
	Приложение 10В. Метод отбора-отказа	41
ГЛАВА 11.	Случайное блуждание	43
11.1.	Введение	44
11.2.	Одномерное случайное блуждание	44
11.3.	Обобщения метода случайных блужданий	51
11.4.	Приложения в физике полимеров	67
11.5.	Непрерывный предел	78
11.6.	Случайные числа	80
	Литература	84
	Дополнительная литература	85
	Приложение 11А. Метод наименьших квадратов	86
	Литература к приложению	90
ГЛАВА 12.	Задача о перколяции	91
12.1.	Введение	92
12.2.	Порог перколяции	95
12.3.	Маркировка кластеров	103

12.4.	Критические показатели и конечномерное масштабирование	118
12.5.	Ренорм-группа	125
	Литература	137
	Дополнительная литература	138
ГЛАВА 13.	Фракталы, модели кинетического роста и клеточные автоматы	139
13.1.	Фрактальная размерность	140
13.2.	Регулярные фракталы и самоподобие	148
13.3.	Процессы роста фракталов	152
13.4.	Клеточные автоматы	167
13.5.	Заключение	172
	Литература	172
	Дополнительная литература	174
ГЛАВА 14.	Приближение к равновесию	175
14.1.	Введение	176
14.2.	Простая модель	177
14.3.	Точный перебор	178
14.4.	Метод Монте-Карло	179
14.5.	Энтропия	182
14.6.	Влияние корреляций	188
14.7.	Равновесная энтропия	189
14.8.	Энтропия и хаос	190
	Литература	191
	Дополнительная литература	192
ГЛАВА 15.	Микроканонический ансамбль	193
15.1.	Введение	194
15.2.	Микроканонический ансамбль	194
15.3.	Моделирование методом Монте-Карло	197
15.4.	Одномерный классический идеальный газ	198
15.5.	Температура и канонический ансамбль	201
15.6.	Модель Изинга	204
15.7.	Поток тепла	211
15.8.	Замечания	217
	Литература	218
	Дополнительная литература	218
	Приложение 15А. Связь средней энергии демона с температурой	219

ГЛАВА 16.	Моделирование канонического ансамбля методом Монте-Карло	221
16.1.	Канонический ансамбль	222
16.2.	Алгоритм Метрополиса	223
16.3.	Проверка распределения Больцмана	225
16.4.	Моделирование двумерной модели Изинга	233
16.5.	Фазовый переход Изинга	243
16.6.	Другие применения модели Изинга	250
16.7.	Моделирование классических жидкостей	256
16.8.	Другие приложения	271
	Литература	274
	Дополнительная литература	275
	Приложение 16А. Флуктуации в каноническом ансамбле	276
	Приложение 16Б. Точный расчет модели Изинга для решетки 2×2	277
ГЛАВА 17.	Квантовые системы	279
17.1.	Введение	280
17.2.	Обзор квантовой теории	281
17.3.	Стационарное уравнение Шредингера	283
17.4.	Нестационарное уравнение Шредингера	289
17.5.	Анализ квантовых систем с помощью метода случайных блужданий	300
17.6.	Вариационные методы Монте-Карло для квантовомеханических систем	308
	Литература	318
	Дополнительная литература	318
ГЛАВА 18.	Эпилог: одинаковые программы — одинаковые решения	321
18.1.	Единство физики	322
18.2.	Перколяция и галактики	324
18.3.	Как компьютеры влияют сегодня на физику?	330
	Литература	331
ПРИЛОЖЕНИЕ Е.	Указатель программ на языке TRUE BASIC:	
	Часть 2	334
ПРИЛОЖЕНИЕ Ж.	Распечатки программ на языке Фортран: Часть 2	335
ПРИЛОЖЕНИЕ З.	Распечатки программ на языке Паскаль: Часть 2	367
ПРЕДМЕТНЫЙ УКАЗАТЕЛЬ		390

УВАЖАЕМЫЙ ЧИТАТЕЛЫ

Ваши замечания о содержании книги, ее оформлении, качестве перевода и другие просим присылать по адресу: 129820, ГСП, Москва, 1-й Рижский пер., д. 2, изд-во «Мир».

Учебное издание

Харви Гулд, Ян Тобочник

КОМПЬЮТЕРНОЕ МОДЕЛИРОВАНИЕ В ФИЗИКЕ

В двух частях

ч. 2

Заведующий редакцией проф. А.Н.Матвеев

Зам. зав. редакцией С.М.Жебровский

Ст. научный редактор В.И.Самсонова

Мл. научный редактор И.В.Терешкина

Художественный редактор К.В.Радченко

Корректор С.С.Суставова

ИБ № 7283

Оригинал-макет подготовлен на персональном компьютере и отпечатан на лазерном принтере в издательстве «Мир»

Подписано к печати 25.05.90. Формат 60 × 90/16.

Бумага офсетная N 1. Гарнитура таймс. Объем 12,5 бум.л.

Усл.печ.л. 25,00. Усл.кр.-отт. 48,38. Уч.-изд.л. 23,70.

Изд. № 2/6898. Тираж 20000 экз. Зак. 635.

Цена 2 р. 50 к.

Издательство «Мир» В/О «Совэкспорткнига» Государственного комитета СССР по печати.

129820, ГСП, Москва, 1-й Рижский пер., 2.

Можайский полиграфкомбинат В/О «Совэкспорткнига» Государственного комитета СССР по печати.

г. Можайск, ул. Мира, 93

