

Министерство науки и высшего образования РФ
ФГБОУ ВО «Тверской государственный университет»
Математический факультет
Направление 02.03.01 Математика и компьютерные науки
Профиль «Математическое и компьютерное моделирование»

ВЫПУСКНАЯ КВАЛИФИКАЦИОННАЯ РАБОТА
(ВЫПУСКНАЯ РАБОТА БАКАЛАВРА)

Метод Монте-Карло для уравнения Фоккера-Планка

Автор:
Крамарь
Владислав Олегович

Научный руководитель:
д. ф.-м. н. Цирулёв А.Н.

Допущен к защите:
Руководитель ООП:

_____ Цветков В. П

Тверь 2021

Оглавление

Введение	3
1 Общая схема метода Монте-Карло	4
1.1 Моделирование случайных блужданий	4
1.2 Метод Монте-Карло в математическом представлении . . .	6
2 Уравнение Фоккера-Планка	8
2.1 Вывод уравнения Фоккера-Планка методом Монте-Карло .	8
2.2 Модельная задача	11
2.3 Компьютерное моделирование реальных решений уравне- ния Фоккера-Планка	12
Заключение	15
Приложение	16
Литература	19

Введение

Данная работа посвящена методу Монте-Карло для решения уравнения Фоккера-Планка. Метод был разработан двумя математиками: Джоном фон Нейманом и Станиславом Уламом и впервые представлен в научном журнале в 1949 году. Свое название метод приобрел благодаря коммуне в княжестве Монако, из-за большого количества казино находящихся в Монте-Карло, поскольку именно рулетка казино - самое наглядное представление генератора случайных чисел.

Актуальность работы объясняется востребованностью метода Монте-Карло в вычислениях и компьютерном моделировании различных решений и процессов. Цель дипломной работы состоит в изучении теории случайных блужданий и метода Монте-Карло, а также в применении математического моделирования случайных блужданий к решению уравнения Фоккера-Планка. Для достижения этой цели были поставлены задачи, которые заключаются в изучении учебной и научной литературы, получении решения уравнения Фоккера-Планка методом Монте-Карло на примере составления программы на языке $C\#$. Объектом исследования являются случайные блуждания генерируемые методом Монте-Карло. Предметом исследования является математическое моделирование решения уравнения Фоккера-Планка методом Монте-Карло.

Работа состоит из введения, двух глав, заключения, приложения и списка литературы. В первой главе представлен теоретический материал по случайным блужданиям и методу Монте-Карло. Вторая глава является заключительной в дипломной работе, она посвящена теории об уравнении Фоккера-Планка, представлению этого уравнения методом Монте-Карло, и математическому моделированию точного решения и решения методом Монте-Карло. В приложении приведен исходный код реализации метода Монте-Карло для уравнения Фоккера-Планка на языке $C\#$. В заключении подведены итоги работы.

Глава 1

Общая схема метода Монте-Карло

1.1 Моделирование случайных блужданий

Рассмотрим простейшую математическую модель случайного блуждания. Пусть точечная частица может совершать только один тип движений: в дискретные моменты времени t_0, t_1, \dots частица совершает скачок вдоль прямой так, что в момент времени t_{n+1} она оказывается в точке, отстоящей на единичное расстояние влево или вправо от точки, где она находилась в момент времени t_n . Без ограничения общности можно считать, что координата частицы в любой момент времени есть целое число. Введем на прямой некоторое начало отсчета и будем писать $\xi_j = m$, если в момент времени t_j частица находилась в точке m , здесь $j = 0, 1, 2, \dots$ и $m = 0, \pm 1, \pm 2, \dots$

Положим, что блуждание имеет случайный характер поведения: прыжок вправо точечная частица совершает с вероятностью p , прыжок влево – с вероятностью $q = 1 - p$. Ограничимся двумя данными возможными перемещениями. Стоит отметить, что вероятность следующего скачка никак не зависит от вероятности предыдущих сценариев прыжков.

При анализе случайных блужданий частицы удобно пользоваться понятием случайной траектории её движения за n шагов. Она представляет собой набор точек (j, ξ_j) , $j = 0, 1, \dots, n$, на двумерной координатной

плоскости, в котором первая координата – это номер члена последовательности, т.е. по сути момент времени $t = j$, а вторая – (случайная) величина, значение которой равно координате частицы в момент времени $t = j$. Для наглядности удобно соединить точки траектории отрезками прямых, на графике получится непрерывная ломаная из n звеньев, координаты узлов которой суть (j, ξ_j) , $j = 0, 1, \dots, n$. При этом возможное смещение частицы за один прыжок, $\xi_j - \xi_{j-1}$, $j = 0, 1, \dots, n$, также является независимой случайной величиной, принимающей значение 1 с вероятностью p и -1 с вероятностью q .

Заметим, что можно рассматривать общий случай, когда частица с некоторыми вероятностями (их сумма равна единице) совершает скачки не только в соседние точки, но и в более отдаленные. Однако в этом общем подходе методы математического моделирования случайных блужданий принципиально не меняются, поэтому в данной работе практически рассматриваются (исключительно для наглядности и экономии вычислительных ресурсов) случайные блуждания только при сделанных выше предположениях.

Итак, математической моделью движения частицы является последовательность случайных величин. Эта модель применяется в математической физике двояким образом. В следующем разделе данной главы показано, как случайные блуждания позволяют найти решения уравнения Фоккера-Планка, которое возникает во многих приложениях: статистической механике, теории диффузии и т.д. С другой стороны, широкий класс цепей Маркова с дискретным временем моделируется случайными блужданиями и наоборот. В следующей главе этот вопрос рассмотрен подробнее вместе с конкретными примерами, допускающими численное моделирование.

1.2 Метод Монте-Карло в математическом представлении

Основная суть метода Монте-Карло заключается в генерации случайных чисел в сочетании с функцией распределения вероятностей для исследуемого процесса. Если рассматривать этот процесс с математической точки зрения, то метод стоит описать так. Требуется найти значение m некоторой изучаемой величины. Для этого выбирают такую случайную величину X , математическое ожидание которой равно m : $M[X] = m$. На практике же разыгрывают N возможных значений x_i случайной величины X , после чего находят их среднее арифметическое

$$\bar{x} = \frac{1}{N} \sum_{i=1}^N x_i. \quad (1.1)$$

Так как последовательность одинаково распределенных случайных величин, у которых существуют математические ожидания, подчиняется закону больших чисел, то при $N \rightarrow \infty$ среднее арифметическое этих величин сходится по вероятности к математическому ожиданию. Таким образом, при больших N величина $\bar{x} \approx m$. Для применения метода Монте-Карло необходимо уметь разыгрывать случайную величину (генерировать случайные числа).

Всего есть **3 основных правила**, которыми следует руководствоваться при розыгрыше случайной величины. Для начала введем еще одну непрерывную величину R , равномерно распределенную в интервале $(0, 1)$, и $r_i (i = 1, 2, \dots)$ в свою очередь, как возможные значения R .

Правило 1. Для того, чтобы разыграть дискретную случайную величину X , заданную законом распределения надо:

X	x_1	x_2	\dots	x_n
p	p_1	p_2	\dots	p_n

1. Разбить интервал $(0, 1)$ оси Or на n частичных интервалов; 2. Выбрать (например, из таблицы случайных чисел) случайное число r_i . Если r_i попало в частичный интервал, то разыгрываемая величина приняла возможное значение x_i .

Правило 2. Для того чтобы разыграть возможное значение x_i непрерывной случайной величины X , зная ее функцию распределения $F(x)$, надо выбрать случайное число x_i , приравнять его функции распределения и решить относительно x_i полученное уравнение $F(x_i) = r_i$.

Правило 3. Для того, чтобы разыграть возможное значение x_i непрерывной случайной величины X , зная ее плотность вероятности $f(x)$, надо выбрать случайное число r_i и решить относительно x_i уравнение

$$\int_{-\infty}^{x_i} f(x)dx = r_i,$$

или уравнение

$$\int_a^{x_i} f(x)dx = r_i,$$

где a — наименьшее конечное возможное значение X .

Глава 2

Уравнение Фоккера-Планка

2.1 Вывод уравнения Фоккера-Планка методом Монте-Карло

Положим, что точечная частица передвигается по прямой, с шагом $h = 1$ вправо с вероятностью p и влево с вероятностью $q = 1 - p$. Если частица начала свое движение из начального положения в точке 0, тем самым выполняя начальное условие, то после прохождения n шагов, мы имеем вероятность нахождения частицы в точке $m \in \mathbb{Z}$ равной

$$P(n, m) = \binom{n}{\frac{n+m}{2}} p^{\frac{n+m}{2}} q^{\frac{n-m}{2}},$$

Давайте рассмотрим график возможных движений частицы по точкам m на прямой Рис. 2.1. Красным цветом отмечены достижимые точки при условии $P(n, m) > 0$ (точка m достижима на шаге n). Это условие эквивалентно условию одинаковой четности n и m (т. е. $n = m \pmod{2}$). Синим цветом отмечены достижимые точки при условии $P(n+1, m) > 0$.

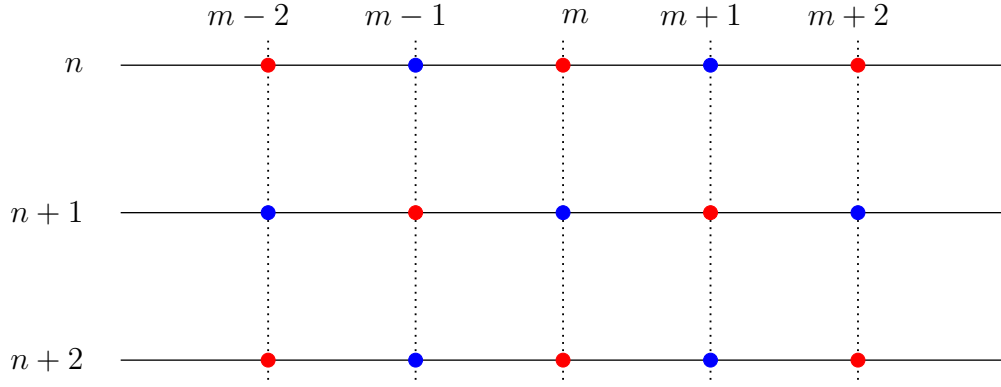


Рис. 2.1: Движение частицы по прямой

Исходя из Рис. 2.1 выведем рекуррентные соотношения:

$$\begin{aligned} P(n+1, m) &= pP(n, m-1) + qP(n, m+1), \\ P(n+2, m) &= p^2P(n, m-2) + 2pqP(n, m) + q^2P(n, m+2). \end{aligned}$$

Далее выведем разностное уравнение исходя из последнего соотношения

$$\begin{aligned} P(n+2, m) - P(n, m) &= p^2P(n, m-2) + (2pq-1)P(n, m) + q^2P(n, m+2) = \\ &= \alpha P(n, m) - \beta [P(n, m+2) - P(n, m-2)] \\ &\quad + \gamma [P(n, m+2) - 2P(n, m) + P(n, m-2)], \end{aligned}$$

в котором коэффициенты должны удовлетворять системе уравнений

$$\begin{cases} \gamma - \beta = q^2 \\ \alpha - 2\gamma = 2pq - 1 \\ \gamma + \beta = p^2 \end{cases}$$

Далее найдем α, β, γ

$$\begin{cases} \alpha = 0 \\ \beta = \frac{(p-q)}{2} \\ \gamma = \frac{(p^2+q^2)}{2} \end{cases}$$

После чего разностное уравнение примет вид:

$$P(n+2, m) - P(n, m) = -\frac{p-q}{2} [P(n, m+2) - P(n, m-2)] + \frac{p^2+q^2}{2} [P(n, m+2) - 2P(n, m) + P(n, m-2)]. \quad (2.1)$$

$P(n, m)$ — функция двух независимых целочисленных переменных. Положим

$$t_n = \tau n, \quad x_m = hm, \quad f(t_n, x_m) = P(n, m).$$

Функция $f(t_n, x_m)$ задана на решетке координатной плоскости (t, x) и может рассматриваться как ограничение на решетку некоторой дважды непрерывно дифференцируемой функции $f(t, x)$, для которой разностное уравнение (2.1) можно записать в виде

$$\begin{aligned} \frac{f(t+2\tau, x) - f(t, x)}{2\tau} = & -h \frac{p-q}{\tau} \frac{f(t, x+2h) - f(t, x-2h)}{4h} \\ & + h^2 \frac{p^2+q^2}{\tau} \frac{f(t, x+2h) - 2f(t, x) + f(t, x-2h)}{4h^2}. \end{aligned}$$

При малых значениях временного шага τ и пространственного шага h последнее уравнение является конечно-разностной аппроксимацией уравнения Фоккера-Планка

$$\frac{\partial f}{\partial t} = -k \frac{\partial f}{\partial x} + D \frac{\partial^2 f}{\partial x^2}, \quad (2.2)$$

в котором коэффициенты скорости процесса (k) и диффузии (D) выражаются через вероятности перехода и величины шагов по формулам

$$k = h \frac{p-q}{\tau}, \quad D = h^2 \frac{p^2+q^2}{\tau}.$$

Шаги τ и h на самом деле связаны ($\tau = ah$) совместным выбором точности аппроксимации для производных по времени и производных по координатам. Поэтому, если k и D заданы, то величины p и h выражаются через них однозначно (при выборе коэффициента пропорциональности a в соответствии с характерными величинами производных в конкретной задаче).

2.2 Модельная задача

Уравнение Фоккера-Планка для плотности функции распределения вероятностей $f = f(t, x)$ с постоянными коэффициентом переноса k и коэффициентом диффузии D имеет вид

$$\frac{\partial f}{\partial t} = -k \frac{\partial f}{\partial x} + D \frac{\partial^2 f}{\partial x^2}. \quad (2.3)$$

Преобразованием переменных

$$t = \frac{D}{k^2} \tilde{t}, \quad x = \frac{D}{k} \tilde{x}$$

уравнение (2.3) приводится к стандартной форме

$$\frac{\partial \tilde{f}}{\partial \tilde{t}} = -\frac{\partial \tilde{f}}{\partial \tilde{x}} + \frac{\partial^2 \tilde{f}}{\partial \tilde{x}^2}, \quad \tilde{f}(\tilde{t}, \tilde{x}) = f(t(\tilde{t}), x(\tilde{x})). \quad (2.4)$$

Фундаментальное решение для (2.4), т. е. решение начально-краевой задачи с условиями

$$\tilde{f}(0, \tilde{x}) = \delta(\tilde{x}), \quad \tilde{f}(\tilde{t}, -\infty) = 0, \quad \tilde{f}(\tilde{t}, +\infty) = 0,$$

дается формулой [1]

$$\tilde{f}(\tilde{t}, \tilde{x}) = \frac{1}{\sqrt{4\pi\tilde{t}}} \exp \left\{ -\frac{(x-t)^2}{4t} \right\}.$$

После перехода к исходным переменным, решение уравнения (2.3) с начально-краевыми условиями

$$f(0, x) = \delta(x), \quad f(t, -\infty) = 0, \quad f(t, +\infty) = 0,$$

примет вид

$$f(t, x) = \frac{1}{\sqrt{4\pi Dt}} \exp \left\{ -\frac{(x-kt)^2}{4Dt} \right\}. \quad (2.5)$$

Прямым интегрированием можно проверить, что условие нормировки

$$\int_{-\infty}^{+\infty} f(t, x) dx = 1$$

выполнено.

2.3 Компьютерное моделирование реальных решений уравнения Фоккера-Планка

Давайте определим, насколько метод Монте-Карло точен при решении уравнения Фоккера-Планка. Для этого смоделируем точное решение уравнения Фоккера-Планка, а после - решение методом Монте-Карло, далее сосчитаем статистическую ошибку и выведем графики, для визуального сравнения решений. Моделирование будет реализовано с помощью собственноручно написанной программы на языке $C^\#$. На графиках изображена плотность распределения вероятностей.

Ниже представлен первый график для точного решения уравнения Фоккера-Планка, при параметрах вероятности шага вправо равной $p = 0,6$ и количестве испытаний в $n = 40$.

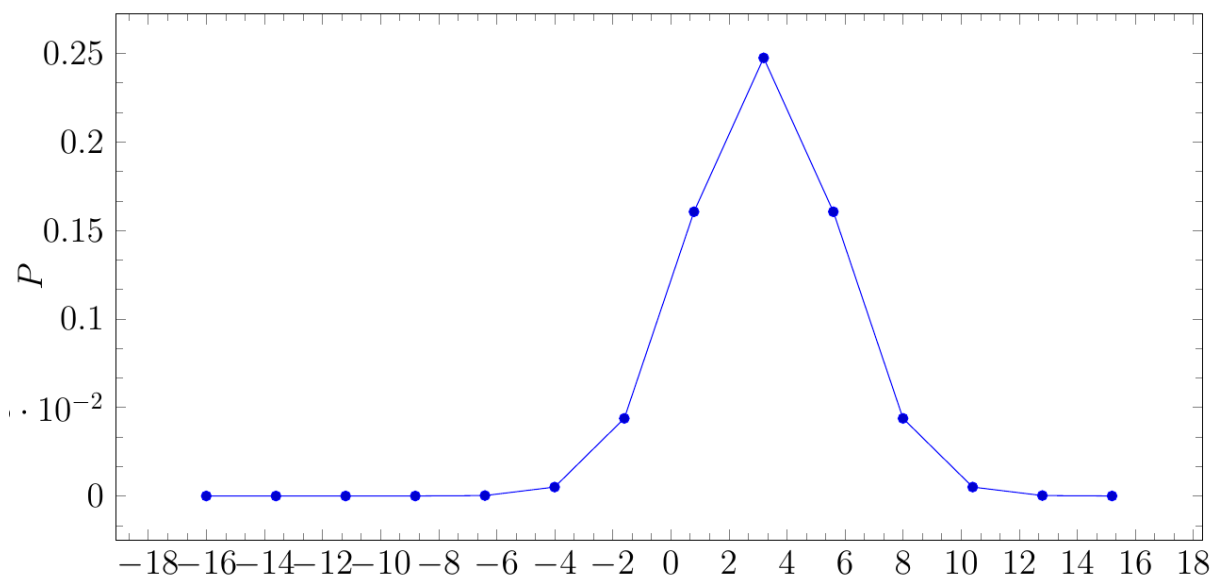


Рис. 2.2: График плотности распределения вероятностей для точного решения

Из Рис. 2.2 можно заметить что, график принимает форму функции (2.5), что свойственно для уравнения Фоккера-Планка. Следующий график отображает плотность распределения для решения уравнения Фоккера-Планка методом Монте-Карло, параметры заданы аналогичное прошлому графику, вероятность шага вправо $p = 0.6$, а количество испытаний $n = 40$.

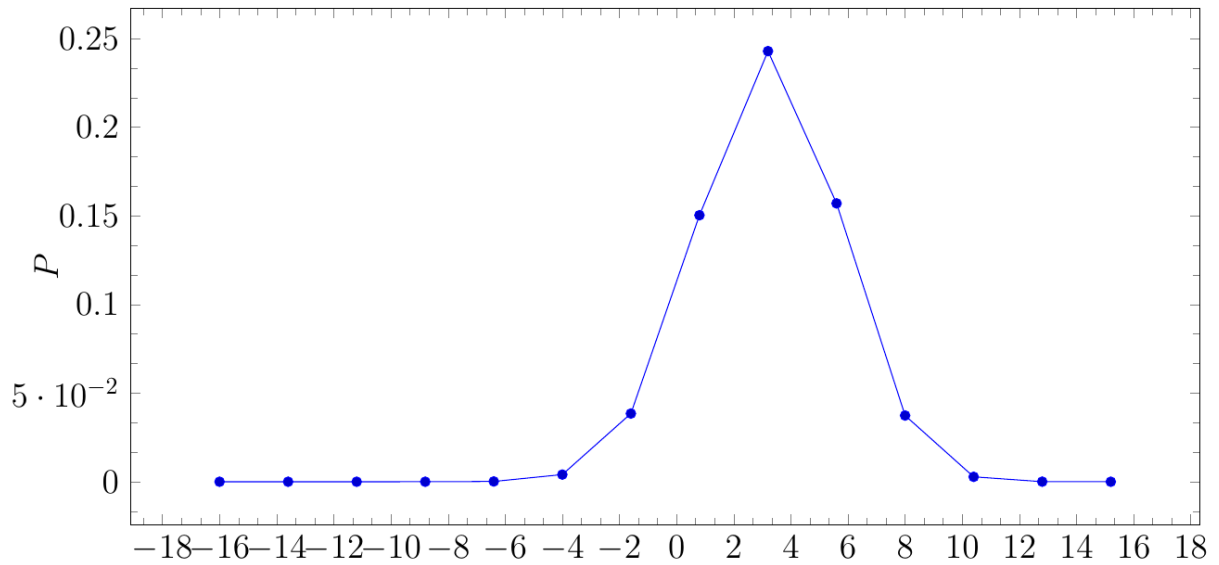


Рис. 2.3: График плотности распределения вероятностей методом Монте-Карло

На Рис. 2.3 график сильно схож с предыдущим. Для большей наглядности наложим два графика друг на друга и увидим насколько велика разница в результатах моделирования точным решением и методом Монте-Карло.

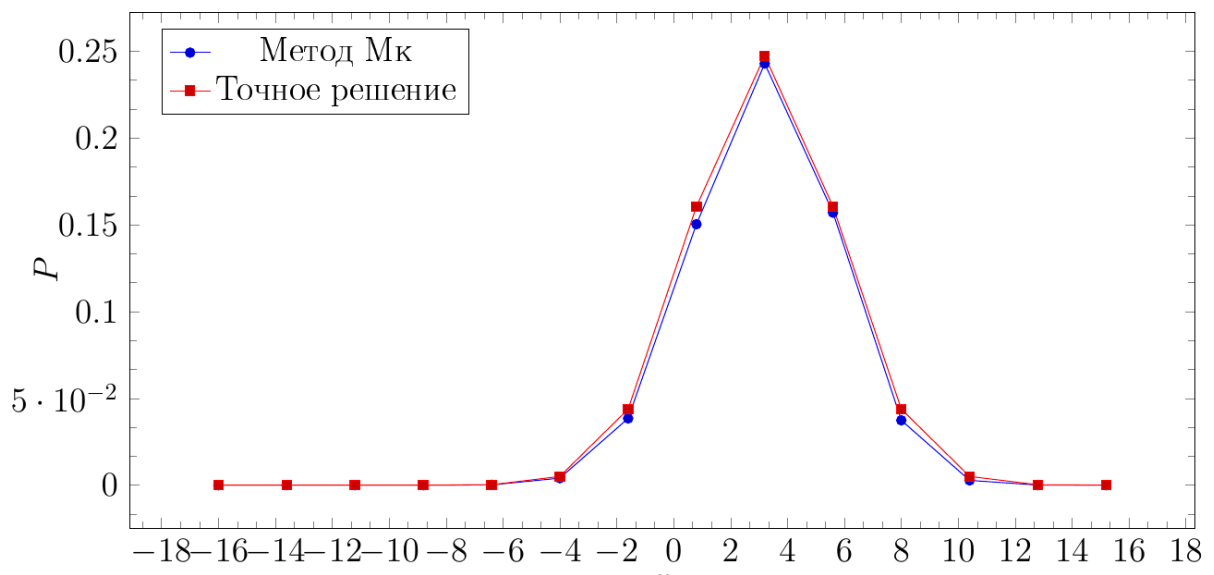


Рис. 2.4: Наложенные друг на друга графики двух решений

На Рис. 2.4 наглядно видны минимальные расхождения между двумя графиками, это говорит о близости двух решений. Стоит сосчитать статистическую ошибку, чтобы убедиться в точности метода. Для этого воспользуемся формулой

$$\frac{1}{n} \sqrt{\sum_{i=0}^n (P[i] - f[i])^2} = EPS \quad (2.6)$$

При расчете по данной формуле, статистическая ошибка выходит совершенно небольшая $EPS = 0,009$ - что окончательно говорит о большой точности метода Монте-Карло при моделировании решения уравнения Фоккера-Планка. Из чего мы можем сделать окончательный вывод о пригодности метода и достаточной точности при решении методом Монте-Карло.

Заключение

В работе получены следующие основные результаты:

1. Методом случайных блужданий построен оригинальный вывод уравнения Фоккера-Планка с постоянными коэффициентами переноса и диффузии (для таких уравнений имеется известное точное решение).
2. Алгоритм случайных блужданий с дискретным временем использован для построения решений уравнения Фоккера-Планка методом Монте-Карло. Проведено сравнение точного решения с результатом математического моделирования.
3. Алгоритм метода Монте-Карло реализован на языке $C\#$.

Приложение

Программная реализация метода Монте-Карло для Уравнения Фоккера-Планка

```
1
using System;
3 using System.Text;
using System.IO;
5
7 namespace RandomWalks
{
9     class Program
    {
11         static void Main(string[] args)
            {
13             Console.Write("Введите число шагов n = ");
                int n = Convert.ToInt32(Console.ReadLine());
15             Console.WriteLine();
                Console.Write("Введите вероятность перехода направо p = ");
17             double p = Convert.ToDouble(Console.ReadLine());
                Console.WriteLine();
19
                double Xmax = 16.0, T = 8.0, h, tau, k, D, EPS = 0;
21             h = Xmax / n; tau = T / n; k = Xmax / T * (2 * p - 1); D = h *
                Xmax / T * (2 * p * p - 2 * p + 1);
23
                int dim = 2 * n + 1, N = 100000000, scale;
                scale = 2 * Convert.ToInt32(Convert.ToDouble(n) / Xmax + 0.01);
25
                Console.WriteLine
27             (" h={0,5:F3}, tau={1,5:F3}, scale={2,2}, k={3,3:F2}, D={4,3:F2}"
                , h, tau, scale, k, D);
                Console.WriteLine();
29
                string path1 = @"C:\Users\at-te\Desktop\ДИПЛОМТех\\1337.txt";
31             string path2 = @"C:\Users\at-te\Desktop\ДИПЛОМТех\\228.txt";
                string format = " {0,4} {1,12:F9}";
33
                Frog frog = new Frog(n, p, N);
35
                //Запись значений точного решения уравнения ФоккераПланка- в массив FP
37
                double[] FP = new double[dim];
39             for (int m = 0; m < dim; m++)
                {
41                 double x = (m - n) * h - k * T;
                    FP[m] = 1 / Math.Sqrt(4 * Math.PI * D * T) * Math.Exp(-x * x
                        / (4 * D * T));
43             }
```



```

45         //Вывод результатов моделирования и точного решения из массивов P и FP
46         //на экран и в файлы Model.txt и Exact.txt
47
48         for (int i = 0; i < dim; i++)
49         {
50             if (i % scale == 0)
51                 Console.WriteLine(" {0,4} {1,12:F9} {2,12:F9}", h * (i
- n), 1.9 * frog.P[i], 1.6 * FP[i]);
52         }
53
54         using (StreamWriter MC = new StreamWriter(path1, false, Encoding.
Default))
55         {
56             for (int i = 0; i < dim; i++)
57             {
58                 if (i % scale == 0)
59                 {
60                     MC.WriteLine(format, h * (i - n), 1.9 * frog.P[i]);
61                 }
62             }
63         }
64
65         using (StreamWriter E = new StreamWriter(path2, false, Encoding.
Default))
66         {
67             for (int i = 0; i < dim; i++)
68             {
69                 EPS+= Math.Pow((frog.P[i] - FP[i]), 2);
70
71                 if (i % scale == 0)
72                 {
73                     E.WriteLine(format, h * (i - n), 1.6 * FP[i]);
74                 }
75             }
76             Console.WriteLine(((Math.Sqrt(EPS)) / 40));
77         }
78
79         Console.ReadKey();
80     }
81 }
82
83
84
85 class Frog
86 {
87     public int n, N;
88     public double p;
89     public double[] P;
90
91     public Frog(int n, double p, int N)
92     {
93         this.n = n; this.p = p; this.N = N;
94         int dim = 2 * n + 1;
95         this.P = new double[dim];

```

```

97         Random rnd = new Random();

99         for (int i = 0; i < N; i++)
100         {
101             int k = n;
102             for (int j = 0; j < n; j++)
103             {
104                 if (rnd.NextDouble() < p) { k = k + 1; }
105                 else { k = k - 1; }
106             }
107             P[k] += 1;
108         }
109
110         for (int i = 0; i < dim; i++) P[i] = P[i] / N;
111     }
112 }
113 }

```

Литература

- [1] Тихонов А.Н., Самарский А.А. *Уравнения математической физики* Изд-во МГУ, 6-е изд., 1999.
- [2] Carlsson J. A backward *Monte-Carlo method for solving parabolic partial differential equations* <http://arxiv.org/abs/math/0010118>
- [3] Кирякин К.С. *Метод Монте-Карло для решения уравнения Фоккера-Планка-Колмогорова* Сборник научных трудов НГТУ. – 2007
- [4] Ф. Спицер. *Принципы случайного блуждания*. М: Мир, 1969, 292с.
- [5] Б.В. Гнеденко. *Курс теории вероятностей*. М: Едиториал УРСС, 2005, 448с.