Министерство науки и высшего образования Российской Федерации Федеральное государственное бюджетное образовательное учреждение высшего образования «Тульский государственный университет»

КАФЕДРА ИНФОРМАЦИОННОЙ БЕЗОПАСНОСТИ

**СТАТИСТИЧЕСКОЕ МОДЕЛИРОВАНИЕ.   
МЕТОД МОНТЕ-КАРЛО**

отчет о практической работе №4

по дисциплине

*ТЕОРИЯ СИСТЕМ И СИСТЕМНЫЙ АНАЛИЗ*

**Вариант №14**

Выполнила \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

ст. гр. №230711, Павлова В.С.

Проверила \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

к. т. н, доцент Грачева И.А.

Тула, 2023

## **ЦЕЛЬ И ЗАДАЧА РАБОТЫ**

**Цель работы**: знакомство с основами статистического моделирования систем различного типа.

**Задание на работу**:

1. Построить на плоскости фигуру с 7-мью углами.

2. Найти площадь фигуры методом Монте-Карло. Точность расчета должна составлять более 90%.

## **ХОД РАБОТЫ**

Пусть семиугольная фигура задаётся следующими координатами вершин: (0, 0), (2, 2), (4, 3), (5, 1), (4, 0), (3, 0), (2, 1). Полученный семиугольник представлен на рисунке 1.

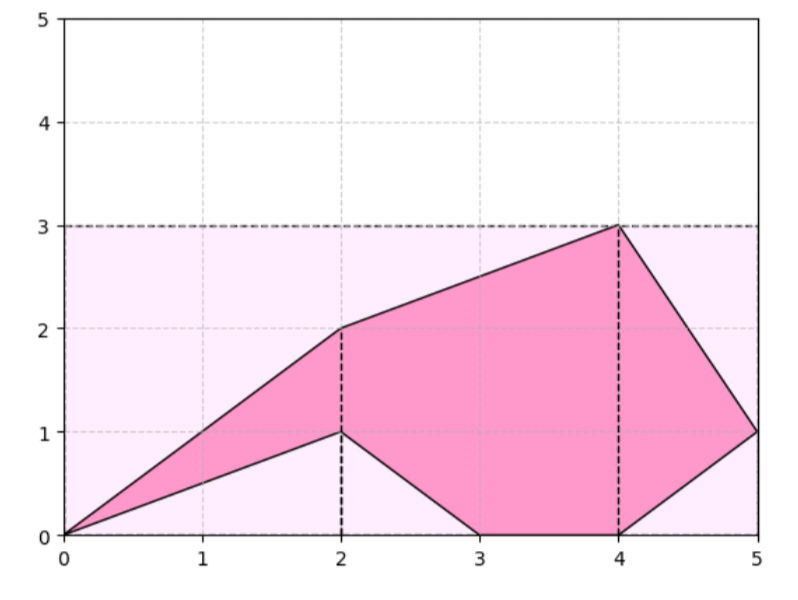


Рисунок 1 – Фигура на плоскости

Если рассчитать площадь фигуры вручную, разбив её на треугольники и прямоугольники, получим, что точное значение площади . Прямоугольник , в который вписана фигура , имеет координаты вершин (0,0), (0,3), (5,0), (5,3), а площадь его составляет .

Для решения задачи методом Монте-Карло нам необходимо сгенерировать пары чисел (*R*, *G)*, равномерно распределенных в интервале от 0 до 1. Здесь число *R* имитирует координату (0 ≤ ≤ 5), поэтому = 5\**R*. Число *G*, в свою очередь, имитирует координату (0 ≤ ≤ 3), следовательно, = 3\**G*. Сгенерируем по 10 чисел *R* и *G* и отобразим 10 точек на графике и занесём их в таблицу 1.

Таблица 1 – Решение задачи методом Монте-Карло

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **№п/п** | ***R*** | **G** | **X** | **Y** |  |  |
| 1 | 0.699393 | 0.171636 | 3.496965 | 0.514908 | yes | yes |
| 2 | 0.0619526 | 0.0404065 | 0.309763 | 0.12122 | yes | no |
| 3 | 0.44145 | 0.510758 | 2.20725 | 1.532274 | yes | yes |
| 4 | 0.314219 | 0.686575 | 1.571095 | 2.059725 | yes | no |
| 5 | 0.977111 | 0.49263 | 4.885555 | 1.47789 | yes | no |
| 6 | 0.435652 | 0.406629 | 2.17826 | 1.219887 | yes | yes |
| 7 | 0.00177007 | 0.36726 | 0.00885 | 1.10178 | yes | no |
| 8 | 0.0847804 | 0.830409 | 0.423902 | 2.491227 | yes | no |
| 9 | 0.0375378 | 0.220985 | 0.187689 | 0.662955 | yes | no |
| 10 | 0.312265 | 0.505814 | 1.561325 | 1.517442 | yes | yes |
| **Total:** | | | | | 10 | 4 |

Статистическая гипотеза заключается в том, что количество точек, попавших в контур фигуры, пропорционально площади фигуры: . То есть, по формуле метода Монте-Карло, получаем, что площадь семиугольника . Это значение разнится с точным значением площади на величину .

Для оценки точности исследуем, как менялась величина от опыта к опыту (таблица 2). В ней p – вероятность попадания случайной точки в испытуемую область.

Таблица 2 – Оценка точности ответа

|  |  |  |
| --- | --- | --- |
| **Кол-во испытаний** | **Оценка вероятности p** | **Оценка площади S методом Монте-Карло** |
| 1 |  |  |
| 2 |  |  |
| 3 |  |  |
| 4 |  |  |
| 5 |  |  |
| 6 |  |  |
| 7 |  |  |
| 8 |  |  |
| 9 |  |  |
| 10 |  |  |

Как видно по таблице, необходимая точность не достигается: результат должен совпадать с точным значением до первой цифры после запятой включительно, то есть колебаться не больше, чем 6,7 . Для достижения большей точности увеличим количество испытаний до 100.

С помощью программы на языке программирования С++, код которой приведён в листинге 1, сгенерируем ещё 90 точек и вычислим для них аналогичным образом оценку точности. Вычисления приведены в таблице 3. В ней S – площадь фигуры, p – вероятность попадания случайной точки в испытуемую область.

Таблица 3 – Оценка точности для решения с числом испытаний до N = 100

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **N** | ***x*** | ***y*** | **Оценка площади S** | **Оценка вероятности p** |
| 11 | 0.276803 | 0.846248 | 5.45455 | 0.363636 |
| 12 | 2.06824 | 1.39814 | 6.25 | 0.416667 |
| 13 | 4.10367 | 0.486343 | 6.92308 | 0.461538 |
| 14 | 2.99371 | 1.12357 | 7.5 | 0.5 |
| 15 | 4.42778 | 0.155095 | 7 | 0.466667 |
| 16 | 1.56285 | 0.543657 | 6.5625 | 0.4375 |
| 17 | 1.66906 | 2.88949 | 6.17647 | 0.411765 |
| 18 | 1.12766 | 0.757073 | 6.66667 | 0.444444 |
| 19 | 2.17246 | 1.4594 | 7.10526 | 0.473684 |
| 20 | 4.38383 | 1.62987 | 7.5 | 0.5 |
| 21 | 0.557726 | 0.84817 | 7.14286 | 0.47619 |
| 22 | 4.45128 | 1.92523 | 7.5 | 0.5 |
| 23 | 1.77145 | 2.24604 | 7.17391 | 0.478261 |
| 24 | 0.84994 | 2.97162 | 6.875 | 0.458333 |
| 25 | 0.408032 | 0.694998 | 6.6 | 0.44 |
| 26 | 3.83419 | 2.81808 | 6.92308 | 0.461538 |
| 27 | 1.93121 | 2.27131 | 6.66667 | 0.444444 |
| 28 | 0.975524 | 2.02228 | 6.42857 | 0.428571 |
| 29 | 0.117344 | 1.85949 | 6.2069 | 0.413793 |
| 30 | 1.93472 | 1.34367 | 6.5 | 0.433333 |
| 31 | 0.707724 | 0.0460524 | 6.29032 | 0.317073 |
| 32 | 4.93393 | 2.04691 | 6.09375 | 0.309524 |
| 33 | 0.508896 | 2.45643 | 5.90909 | 0.302326 |
| 34 | 3.80154 | 0.303507 | 6.17647 | 0.318182 |
| 35 | 3.76827 | 2.27049 | 6.42857 | 0.333333 |
| 36 | 2.04093 | 1.03192 | 6.66667 | 0.347826 |
| 37 | 3.7466 | 2.24458 | 6.89189 | 0.361702 |
| 38 | 0.23835 | 2.8146 | 6.71053 | 0.354167 |
| 39 | 1.83981 | 2.61034 | 6.53846 | 0.346939 |
| 40 | 1.01764 | 1.62181 | 6.375 | 0.34 |
| 41 | 0.986358 | 1.12778 | 6.21951 | 0.414634 |
| 42 | 3.68755 | 0.55858 | 6.42857 | 0.428571 |
| 43 | 0.307474 | 2.41853 | 6.27907 | 0.418605 |

Таблица 3 – Оценка точности для решения с числом испытаний до N = 100 (продолжение)

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **N** | ***x*** | ***y*** | **Оценка площади S** | **Оценка вероятности p** |
| 44 | 4.28312 | 0.491562 | 6.47727 | 0.431818 |
| 45 | 2.15125 | 2.62481 | 6.33333 | 0.422222 |
| 46 | 4.42503 | 0.286111 | 6.19565 | 0.413043 |
| 47 | 2.11814 | 0.0520035 | 6.06383 | 0.404255 |
| 48 | 3.15516 | 2.14295 | 6.25 | 0.416667 |
| 49 | 4.32432 | 0.575976 | 6.42857 | 0.428571 |
| 50 | 2.172 | 2.56694 | 6.3 | 0.42 |
| 51 | 1.11133 | 2.12336 | 6.17647 | 0.411765 |
| 52 | 2.87744 | 1.52724 | 6.34615 | 0.423077 |
| 53 | 0.351115 | 2.65374 | 6.22642 | 0.415094 |
| 54 | 3.48872 | 2.22361 | 6.38889 | 0.425926 |
| 55 | 4.53795 | 2.34382 | 6.27273 | 0.418182 |
| 56 | 3.76034 | 1.29826 | 6.42857 | 0.428571 |
| 57 | 0.0273141 | 2.2823 | 6.31579 | 0.421053 |
| 58 | 3.10526 | 1.56771 | 6.46552 | 0.431034 |
| 59 | 1.44154 | 0.982757 | 6.61017 | 0.440678 |
| 60 | 2.28767 | 0.157567 | 6.5 | 0.433333 |
| 61 | 1.25141 | 2.64028 | 6.39344 | 0.42623 |
| 62 | 3.8432 | 2.58773 | 6.53226 | 0.435484 |
| 63 | 2.61116 | 1.674 | 6.66667 | 0.444444 |
| 64 | 1.94174 | 1.91562 | 6.79688 | 0.453125 |
| 65 | 2.50862 | 0.176611 | 6.69231 | 0.446154 |
| 66 | 2.41844 | 2.43144 | 6.59091 | 0.439394 |
| 67 | 4.28037 | 1.95837 | 6.71642 | 0.447761 |
| 68 | 4.64309 | 0.612964 | 6.61765 | 0.441176 |
| 69 | 1.32206 | 0.995941 | 6.73913 | 0.449275 |
| 70 | 0.783715 | 2.36076 | 6.64286 | 0.442857 |
| 71 | 1.32511 | 1.27903 | 6.76056 | 0.450704 |
| 72 | 2.26981 | 1.64095 | 6.875 | 0.458333 |
| 73 | 0.545213 | 0.880398 | 6.78082 | 0.452055 |
| 74 | 2.23029 | 1.25156 | 6.89189 | 0.459459 |
| 75 | 0.389874 | 0.871517 | 6.8 | 0.453333 |
| 76 | 4.07834 | 0.0281991 | 6.71053 | 0.447368 |

Таблица 3 – Оценка точности для решения с числом испытаний до N = 100 (продолжение)

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **N** | ***x*** | ***y*** | **Оценка площади S** | **Оценка вероятности p** |
| 77 | 4.24268 | 1.15781 | 6.81818 | 0.454545 |
| 78 | 0.808741 | 1.5407 | 6.73077 | 0.448718 |
| 79 | 1.95303 | 1.65413 | 6.83544 | 0.455696 |
| 80 | 2.12699 | 0.115452 | 6.75 | 0.45 |
| 81 | 1.62023 | 1.83432 | 6.66667 | 0.444444 |
| 82 | 0.976592 | 0.853664 | 6.76829 | 0.45122 |
| 83 | 2.28172 | 0.706076 | 6.68675 | 0.445783 |
| 84 | 3.38435 | 1.59554 | 6.78571 | 0.452381 |
| 85 | 1.91549 | 0.651051 | 6.70588 | 0.447059 |
| 86 | 0.717643 | 2.4156 | 6.62791 | 0.44186 |
| 87 | 4.09238 | 0.955748 | 6.72414 | 0.448276 |
| 88 | 0.471816 | 2.25172 | 6.64773 | 0.443182 |
| **89** | 4.47676 | 1.28681 | **6.74157** | 0.449438 |
| **90** | 2.49962 | 0.134129 | **6.76667** | 0.444444 |
| **91** | 1.71087 | 0.930662 | **6.75824** | 0.450549 |
| **92** | 2.08258 | 2.81112 | **6.78478** | 0.452319 |
| **93** | 3.72784 | 0.135594 | **6.77419** | 0.451613 |
| **94** | 4.82025 | 0.819514 | **6.70213** | 0.446809 |
| **95** | 0.458693 | 0.208289 | **6.73158** | 0.448772 |
| **96** | 1.61016 | 0.149419 | **6.7625** | 0.450833 |
| **97** | 3.91095 | 0.660482 | **6.74948** | 0.449965 |
| **98** | 1.77023 | 0.149236 | **6.78163** | 0.452109 |
| **99** | 4.23917 | 2.29896 | **6.76667** | 0.451111 |
| **100** | 3.3404 | 2.11631 | **6.75** | 0.45 |

Как видно по таблице, первый знак после запятой в оценке площади перестал колебаться, а значит нужная точность достигнута после 89 испытаний, что примерно соответствует теоретической оценке того, что .

Таким образом, можно утверждать, что площадь фигуры приблизительно равна , что соотносится с точным расчётом, согласно которому .

Листинг 1 – Код программы на языке программирования C++, реализующей вычисление оценки площади методом Монте-Карло

#include <iostream>

#include <cstdlib>

#include <ctime>

#include <fstream>

#include <vector>

void Randomizer()

{

std::ofstream inputFile("input.txt");

std::srand(static\_cast<unsigned int>(std::time(nullptr)));

int numPairs = 10;

for (int i = 0; i < numPairs; ++i)

{

double R = static\_cast<double>(std::rand()) / RAND\_MAX;

double G = static\_cast<double>(std::rand()) / RAND\_MAX;

inputFile << R \* 5 << " " << G \* 3 << "\n";

}

inputFile.close();

}

bool isInsideRectangle(double x, double y)

{

return (x >= 0.0 && x <= 5.0 && y >= 0.0 && y <= 3.0);

}

bool isInsideHexagon(double x, double y)

{

return (

(y >= 0.5 \* x) && (y <= x) && (x >= 0) && (x <= 2)

|| (y >= -x+3) && (y <= 0.5\*x+1) && (x >= 2) && (x <= 3)

|| (y >= 0) && (y <= 0.5 \* x + 1) && (x >= 3) && (x <= 4)

|| (y >= x - 4) && (y <= -2\*x+11) && (x >= 4) && (x <= 5)

);

}

int main(void)

{

Randomizer();

std::ifstream inputFile("input.txt");

std::vector<std::pair<double, double>> points;

double x, y;

int ctrY = 40, ctrTotal = 90;

while (inputFile >> x >> y)

{

points.emplace\_back(x, y);

}

inputFile.close();

Листинг 1 – Код программы на языке программирования C++, реализующей вычисление оценки площади методом Монте-Карло (продолжение)

for (const auto& point : points)

{

if (isInsideHexagon(point.first, point.second))

{

ctrY++;

}

ctrTotal++;

std::cout << point.first << "\t" << point.second;

std::cout << "S: " << (((double)ctrY \* 15) / ctrTotal) << "\n";

std::cout << "P: " << (((double)ctrY) / ctrTotal) << "\n";

}

return EXIT\_SUCCESS;

}

## **ВЫВОД**

В рамках данной практической работы я ознакомилась с основами статистического моделирования систем различного типа.