Министерство образования и науки Российской Федерации

Федеральное государственное бюджетное образовательное учреждение высшего образования

«Пермский национальный исследовательский

политехнический университет»

(ПНИПУ)

Кафедра вычислительной математики и механики

Лабораторная работа

по дисциплине: «Моделирование процессов и систем»

по теме: «Пушка Монте- Карло»

Выполнил:

Студент группы

Соболь Евгений Валерьевич

Проверил:

доцент кафедры ВМиМ

Максимов Петр Викторович

Пермь

2021 г

**Цель работы:** разработать программный алгоритм, позволяющий проводить моделирование стрельбы из пушки методом Монте-Карло, проанализировать результаты работы, составить графики и гистограммы.

**Задание к лабораторной работе:**

1. Выбрать угол стрельбы α.

2. Определить начальную скорость  для этого угла , чтобы дальность стрельбы L была равна 1 км.

3. Задать параметры нормального закона распределения для скорости  и угла : математическое ожидание для скорости , среднеквадратическое отклонение для скорости , а также математическое ожидание для угла , и среднеквадратическое отклонения для угла .

4. Методом Монте-Карло определить среднюю дальность выстрела и вероятность поражения мишени размером ∆; определить математическое ожидание M и дисперсию D.

5. Построить графики зависимости M и D от числа испытаний.

6. Построить гистограмму распределения случайной величины X.

7. Повторить моделирование для других параметров  и , но таких же ,  и .

8. Повторить моделирование для всех других параметров.

**Основная часть:**

1. Из формулировки задания определим ***входные данные*** алгоритма моделирования стрельбы из пушки методом Монте-Карло:

* Дальность установки мишени (стрельбы) L
* Размер мишени 
* Математическое ожидание для угла (в градусах) m\_alpha
* Математическое ожидание для скорости m\_v
* Среднеквадратичное отклонение для скорости sigma\_v
* Среднеквадратичное отклонение для угла sigma\_alpha
* Количество выстрелов пушки n\_shots
* Размер деления для вычисления средних и дисперсии для диаграмм del\_n\_shots
* Значение (в метрах) от которого строить гистограмму частоты попаданий graph\_start
* Значение до которого строить гистограмму частоты попаданий graph\_stop
* Значений размера деления для гистограммы частоты попаданий graph\_del

1. Определим ***выходные данные*** алгоритма моделирования стрельбы из пушки методом Монте-Карло:

* Средняя дальность выстрела
* Вероятность попадания в мишень размером 
* Математическое ожидание
* Дисперсия
* График зависимости математического ожидания от числа испытаний
* График зависимости дисперсии от количества испытаний
* Гистограмма распределения случайной величины X

**Описание алгоритма работы программы:**

1. Вводим входные значения.
2. Генерируем случайное число от 0 до 1 по равномерному закону распределения, после чего генерируем случайное число по нормальному закону распределения с математическим ожиданием 0 и среднеквадратичным отклонением 1, как сумму двенадцати случайных чисел от 0 до 1 по равномерному закону распределения минус 6.
3. Генерируем случайное число по нормальному закону распределения с заданными математическим ожиданием и среднеквадратичным отклонением по формуле mu+sigma\*n01, где n01 – число, сгенерированное в шаге B.
4. Подставляем в формулу в шаге C значения математического ожидания и среднеквадратичного отклонения для скорости, получаем значение скорости для расчёта дальности выстрела.
5. Подставляем в формулу в шаге C значения математического ожидания и среднеквадратичного отклонения для угла, получаем значение угла для расчёта дальности выстрела.
6. Производим расчёт дальности выстрела  , записываем значение в список.
7. Повторяем шаги b, c, d, e, f до тех пор, пока не выполним заданное количество испытаний.
8. Считаем математическое ожидание и дисперсию для определённых интервалов испытаний, записываем значения в список.
9. Считаем среднее значение математического ожидания  и дисперсии  для всех испытаний.
10. Определяем среднюю дальность стрельбы и вероятность попадания в мишень  .
11. Производим расчёты частот попадания снарядов в заданные интервалы в метрах, записываем значения частот и сами интервалы в списки.
12. Строим график зависимости математического ожидания от числа испытаний.
13. Строим график зависимости дисперсии от числа испытаний.
14. Строим гистограмму частот попадания в заданные интервалы.
15. Проверяем полученные результаты с помощью правила трёх  .

**Реализуем программу моделирования стрельбы из пушки методом Монте-Карло на Python:**

1. Импортируем всё необходимое для написания алгоритма: функцию для генерации числа по равномерному закону распределения, функцию квадратного корня, синуса, преобразования градусов в радианы, функцию для построения диаграмм и гистограмм.

from random import uniform

from math import sqrt, sin, radians

from matplotlib import pyplot as plt

1. Создадим функцию r\_0\_1 для генерации числа от 0 до 1 по равномерному закону распределения с помощью библиотеки random:

def r\_0\_1() -> float:

"""Случайное число по равн. закону распределения от 0 до 1"""

return uniform(0, 1)

1. Создадим функцию n\_0\_1 для генерации числа с математическим ожиданием 0 и среднеквадратическим отклонением 1 по нормальному закону распределения:

def n\_0\_1() -> float:

"""Случайное число по норм. закону распределения мат. ожид. = 0, сред. квадр. откл. = 1"""

s = 0

for i in range(1, 13):

s += r\_0\_1()

return s – 6

1. Создадим функцию n\_m\_sigma для генерации числа с заданным математическим ожиданием и среднеквадратическим отклонением:

def n\_m\_sigma(m: float, sigma: float) -> float:

"""Случайное число по норм. закону распределения при заданном мат. ожид. и сред. квадр. откл."""

return m + sigma \* n\_0\_1()

1. Создадим cannon\_shot для расчёта длины выстрела по формуле  :

def cannon\_shot(v: float, alpha: float) -> float:

"""Расчёт расстояния выстрела пушки"""

print(f"V: {v}")

print(f"Alpha: {alpha}")

result = ((v\*\*2)\*sin(2 \* alpha)) / 9.8

print(f"-> ВЫСТРЕЛ:{result} (M)")

print("---------------------------")

return result

1. Создадим функцию tank, отвечающую за генерацию заданного количества выстрелов и генерацию списка с результатами экспериментов :

def tank(n: int, m\_v: float, m\_alpha: float, sigma\_v:float, sigma\_alpha: float) -> list:

"""Функция генерирующая заданное количество выстрелов"""

result\_shots = []

k = 1

while k <= n:

v\_shot = n\_m\_sigma(m\_v, sigma\_v)

alpha\_shot = n\_m\_sigma(m\_alpha, sigma\_alpha)

result\_shots.append(cannon\_shot(v\_shot, alpha\_shot))

k += 1

return result\_shots

1. Создадим функцию calc\_mu\_D, которая отвечает за генерацию списков с математическим ожиданием и дисперсией по заданным интервалам испытаний:

def calc\_mu\_D(list\_shots: list, d: int):

"""Вычисление средних мат. ожидание и дисперсии"""

mu = []

D = []

S\_xi2 = 0

S\_xi = 0

for i in range(len(list\_shots)):

S\_xi += list\_shots[i]

S\_xi2 += (list\_shots[i]) \*\* 2

if ((i + 1) % d) == 0:

mu.append(S\_xi / (i + 1))

D.append((1 / i) \* (S\_xi2 - (1 / (i + 1)) \* (S\_xi \*\* 2)))

return mu, D

1. Вычислим среднее математическое ожидание и дисперсию по всем испытаниям:

m\_mu, m\_D = calc\_mu\_D(list\_result\_shots, del\_n\_shots)

average\_mu = sum(m\_mu)/len(m\_mu)

average\_D = sum(m\_D)/len(m\_D)

1. Создадим функцию target\_hit\_probability для расчёта средней дальности выстрела и вероятности попадания в заданную мишень размером :

def target\_hit\_probability(result\_shots: list, n\_shots: int, d: int, L: int):

"""Расчёт попадания в мешень на расстоянии L, размером дельта"""

k = 0

S = 0

for i in result\_shots:

S += i

if (i >= (L - d / 2)) and (i <= (L + d / 2)):

k += 1

print("Средняя дальность выстрела:", S / n\_shots)

print("Вероятность попадания в заданную мишень: ", k / n\_shots)

1. Создадим функцию для вычисления частот попадания величины  в определённые интервалы и сохранения этих частот и интервалов в списки:

def calc\_p(start: int, end: int, d: int, n\_shots: int, result\_shots: list):

"""Расчёты частоты попадания для гистограммы"""

p = []

n = []

S\_P = 0

print(f"Вероятность попадения в интервал [{start}:{end}] = ", end="")

start += d

while start <= end:

k = 0

for j in result\_shots:

if (j >= (start - d)) and (j <= start):

k += 1

p.append(k / n\_shots)

S\_P += (k / n\_shots)

n.append(start - d)

start += d

n.append(end)

print(S\_P)

return p, n

1. Создадим функцию graph\_mu\_d для создания графиков зависимости математического ожидания и дисперсии от количества испытаний по уже сформированным спискам значений, выполненных на 7 шаге:

def graph\_mu\_d(x: int, separation: int, m: list, flag: int):

"""График мат. ожидания и дисперсии"""

plt.plot([i for i in range(1, x, separation)], m)

if flag == 1:

plt.ylabel('Математическое ожидание')

else:

plt.ylabel('Дисперсия')

plt.xlabel('Количество выстрелов')

plt.xlim(1, x)

plt.show()

1. Создадим функцию graph\_p для создания гистограммы частот распределения случайной величины  по уже сформированным спискам значений:

def graph\_p(p: list, m\_n: list):

"""Функция построения гистограммы частоты попаданий"""

n, bins, patches = plt.hist(m\_n[:-1], m\_n, weights=p)

plt.title("Частоты попаданий в интервалы")

plt.ylabel("Частота (%)")

plt.xlabel("Интервал (метрах)")

plt.show()

1. Создадим функцию check\_result, которая выводит среднее значение математического ожидания и дисперсии по всем испытаниям и проверяет правильность результатов работы программы правилом “Трёх  “:

def check\_result(result\_shot: list, mu: float, D: float):

"""Функция проверки, правилом 3 сигма """

check\_start = mu - 3 \* sqrt(D)

check\_stop = mu + 3 \* sqrt(D)

k\_check = 0

for i in result\_shot:

if (i >= check\_start) and (i <= check\_stop):

k\_check += 1

print("---------------ПРОВЕРКА РЕЗУЛЬТАТОВ---------------")

print(f"При MU={mu}, D={D}, sigma={sqrt(D)} попадание в [{check\_start}:{check\_stop}] = {(k\_check/len(result\_shot))\*100}(%)")

1. Создадим и вызовем основную функцию main, в которой принимаются на вход все необходимые для моделирования значения и последовательно выполняются действия, описанные в разделе описания алгоритма программы:

def main():

"""Основная функция, c вводом начальных значений и последовытельным выполнением действий лабораторныой работы"""

# L = float(input("Введите дальность стрельбы (в метрах): "))

L = 1000

# delta = float(input("Введите размер мишени (в метрах): "))

delta = 10

m\_alpha = radians(float(input("Введите мат. ожидание для угла (в градусах): ")))

print(f"Математическое ожидание (в радианах): {m\_alpha}")

# m\_v = float(input("Введите мат. ожид. для скорости: "))

m\_v = sqrt(L \* 9.8/(sin(2 \* m\_alpha)))

print(f"Вычислим мат. ожидание для скорости корень({L}\*9.8)/sin(2\*{m\_alpha}) = {m\_v}")

sigma\_v = float(input("Введите среднеквадр. откл для скорости: "))

sigma\_alpha = float(input("Введите среднеквадр. откл для угла: "))

n\_shots = int(input("Введите количество выстрелов пушки: "))

del\_n\_shots = int(input("Введите размер деления выстрелов для диаграмм: "))

graph\_start = int(input("Введите точку начала для гистограммы: "))

graph\_end = int(input("Введите точку конца для гистограммы: "))

graph\_del = int(input("Введите размер деления для гистограммы: "))

list\_result\_shots = tank(n\_shots, m\_v, m\_alpha, sigma\_v, sigma\_alpha)

m\_mu, m\_D = calc\_mu\_D(list\_result\_shots, del\_n\_shots)

average\_mu = sum(m\_mu)/len(m\_mu)

average\_D = sum(m\_D)/len(m\_D)

# print(m\_mu)

# print(m\_D)

target\_hit\_probability(list\_result\_shots, n\_shots, delta, L)

m\_p, m\_n = calc\_p(graph\_start, graph\_end, graph\_del, n\_shots, list\_result\_shots)

# print(m\_n)

# print(m\_p)

graph\_mu\_d(n\_shots, del\_n\_shots, m\_mu, 1)

graph\_mu\_d(n\_shots, del\_n\_shots, m\_D, 2)

graph\_p(m\_p, m\_n)

check\_result(list\_result\_shots, average\_mu, average\_D)

main()

**КОНЕЦ ПРОГРАММЫ**

**Моделирование ситуаций:**

Центр мишени установили на расстоянии 1000 метров, мишень размером 10 метров, среднее значение угла подъёма пушки (математическое ожидание) равняется 45 градусам. Среднее значение скорости выстрела (математическое ожидание рассчитывается по формуле , среднеквадратичное отклонение для скорости (разброс скорости выстрела) равняется 5, а среднеквадратическое отклонение угла стрельбы равно 0,04. Пушка выстреливает 50 тысяч раз. При заданных исходных получаем среднюю дальность выстрела равную 999.4910618439328, вероятность попадания в мишень 0.04008 , математическое ожидание = 999.179989356 и дисперсию = 10214.3291466.

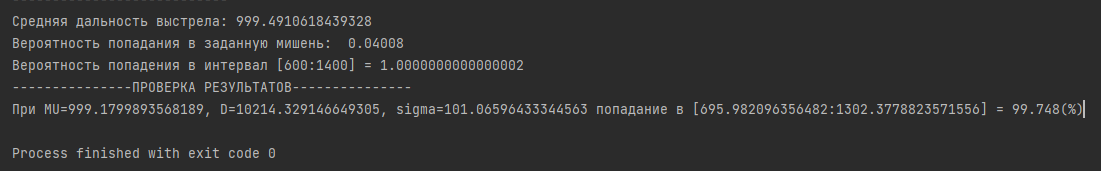


Рисунок 1 – Результаты моделирования

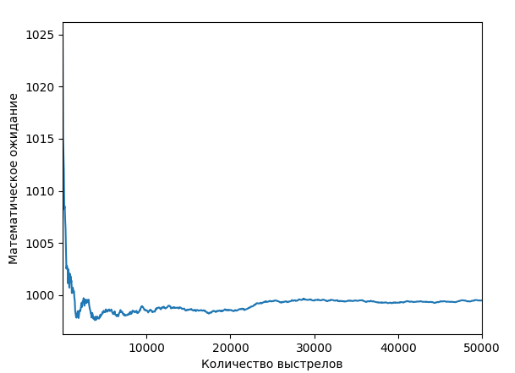


Рисунок 2 - График зависимости математического ожидания от числа выстрелов

Из него можно видеть, что оптимальное число экспериментов для достижения оптимальной точности находится примерно на 25 тысячах испытаний.

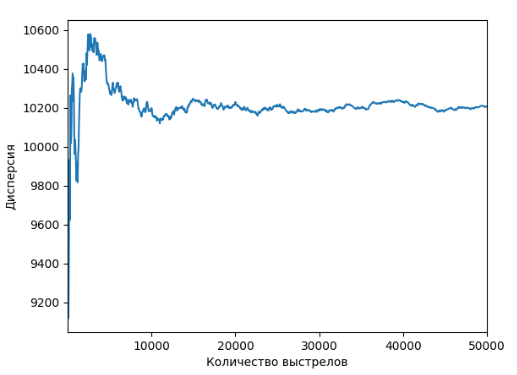


Рисунок 3 - График зависимости дисперсии от числа выстрелов

Из графика видно, что оптимальное количество экспериментов для достижения оптимальной точности находится примерно 23 тысячах экспериментов.

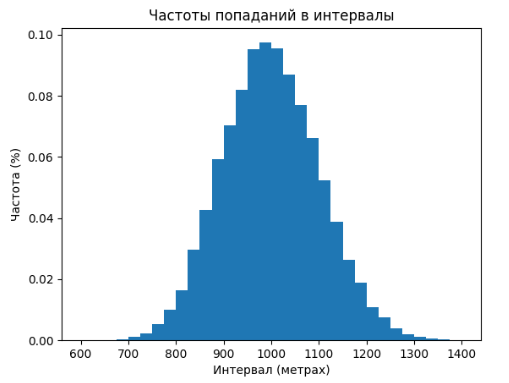


Рисунок 4 - Гистограммы распределения случайной величины  (дальность выстрела)

За деления брались интервалы в 25 метров, начиная с 600 метров и заканчивая 1400 метрами. Из гистограммы видно, что наибольшее количество снарядов упало в интервал [1000, 1025) метров. Чем дальше от центра математического ожидания, тем меньше вероятность попадания снаряда в интервал. Проверив результаты правилом трёх sigma, можно увидеть, что 99.748% снарядов попали в интервал [695.982096356482, 1302.3778823571556], что подтверждает правильность результатов работы алгоритма моделирования.

Изменим значения среднеквадратического отклонения для скорости и угла, и оставим без изменения другие параметры, как указано в 7 пункте задания к лабораторной работе и повторим моделирование. Среднеквадратичное отклонение уменьшим до 2.5, и среднеквадратичное отклонение для угла тоже до 0.03.

Получаем среднюю дальность выстрела равную 998.17731560, вероятность попадания в мишень значительно увеличилась и составила 0.07734, математическое ожидание осталось примерно на прежнем уровне равное 998.5646008, при этом дисперсия уменьшилась примерно в 4 раза и составила 2558.2729868. Из этого можно сделать вывод, что с уменьшением или увеличением среднеквадратического отклонения для скорости и угла пропорционально уменьшается или увеличивается разброс дальности стрельбы.

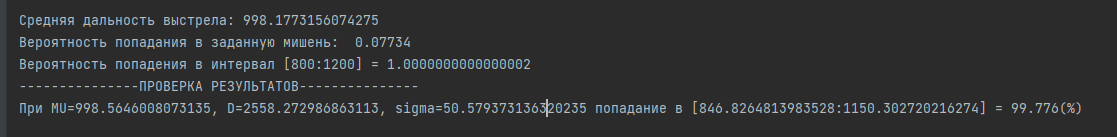


Рисунок 5 – Результаты моделирования

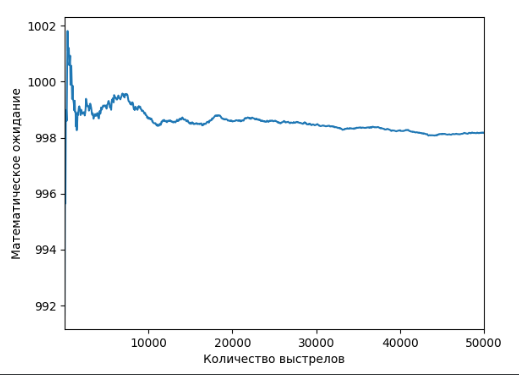


Рисунок 6- График зависимости математического ожидания от числа выстрелов

Значения оптимальной точности математического ожидания от количества экспериментов остались примерно на прежнем уровне.

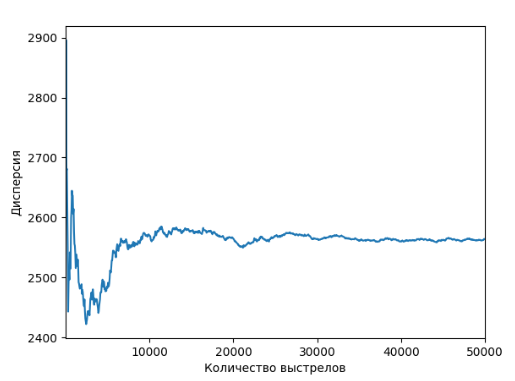


Рисунок 7 - График зависимости дисперсии от числа выстрелов

Значения оптимальной точности дисперсии от количества экспериментов остались примерно на прежнем уровне, хотя сам график по оси находится значительно ниже.

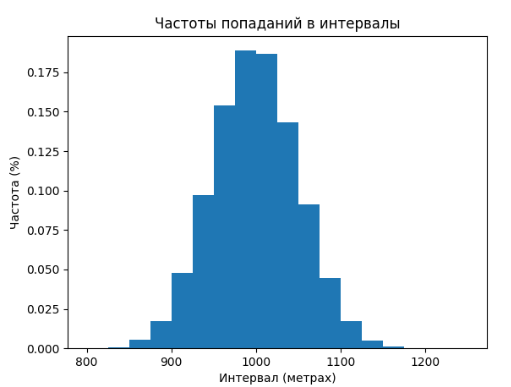


Рисунок 8 - Гистограммы распределения случайной величины  (дальность выстрела)

По данной гистограмме можно видеть, что разброс дальности выстрелов значительно уменьшился, а также увеличилось количество попаданий, приближенных к математическому ожиданию.

Изменим значения среднеквадратического отклонения для скорости и угла, и оставим без изменения другие параметры, как указано в 7 пункте задания к лабораторной работе и повторим моделирование. Среднеквадратичное отклонение увеличим до 10 и среднеквадратичное отклонение для угла тоже до 0.05.

Получаем среднюю дальность выстрела равную 1004.3966324, вероятность попадания в мишень значительно уменьшилась по сравнению с первоначальным результатом и составила 0.01924, математическое ожидание незначительно увеличилось и составляет 1003.4385767769767 метра, при этом дисперсия увеличилась примерно в 4 раза и составила 40313.2729868. Из этого можно сделать вывод, что с увеличением среднеквадратичного отклонения скорости в 2 раза в 4 раза увеличивается разброс дальности стрельбы (дисперсии). Среднеквадратичное отклонение угла за счёт меньшего изменения влияет гораздо меньше на точность стрельбы.

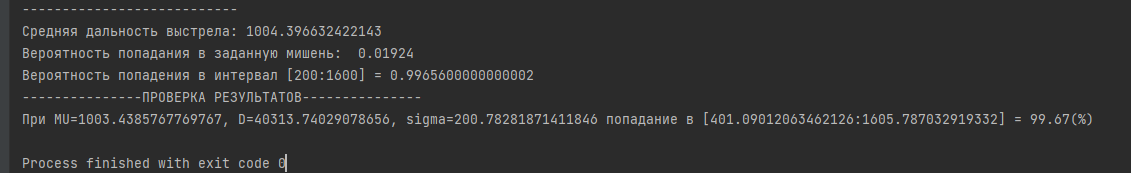


Рисунок 9 – Результаты моделирования

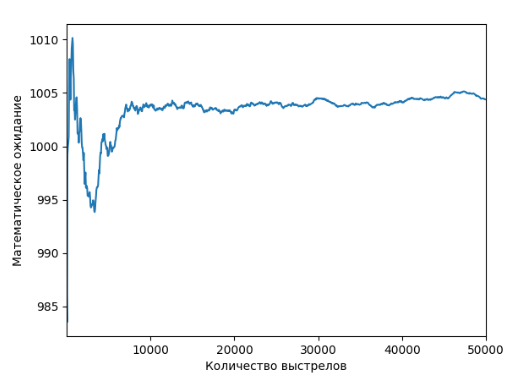


Рисунок 10 - График зависимости математического ожидания от числа выстрелов

Значения оптимальной точности математического ожидания от количества экспериментов уменьшилось, как и уменьшилась точность значения математического ожидания. Если необходима большая точность математического ожидания, то необходимо увеличить число экспериментов до достижения более ровного горизонтального графика.

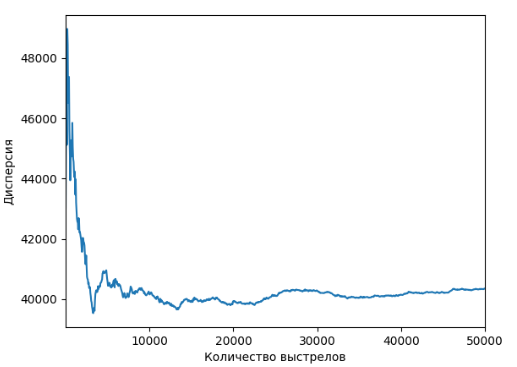


Рисунок 11 - График зависимости дисперсии от числа выстрелов

Значения оптимальной точности дисперсии от количества экспериментов сместилось примерно на отметку в 30 тысяч испытаний.

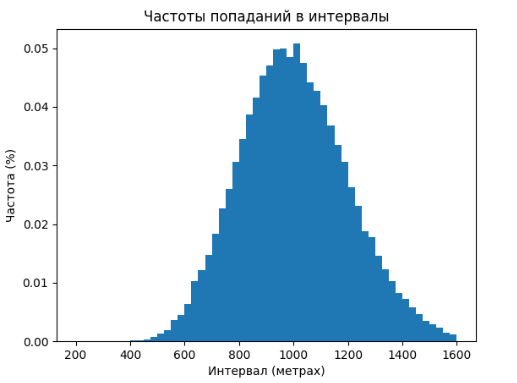


Рисунок 12 – Гистограмма распределения случайной величины Xi

По данной гистограмме можно видеть, что разброс дальности выстрелов значительно увеличился, а также уменьшилось количество попаданий, приближенных к математическому ожиданию.

Повторим моделирование для других параметров, как сказано в 8 пункте задания к лабораторной работе.

Центр мишени установили на расстоянии 5000 метров, мишень размером 20 метров, среднее значение угла подъёма пушки (математическое ожидание) равняется 60 градусам. Среднее значение скорости выстрела (математическое ожидание) рассчитывается по формуле , можно заметить, что скорость значительно увеличилась, вместе с тем как увеличилась дистанция до цели. Среднеквадратичное отклонение для скорости (разброс скорости выстрела) равняется 10, а среднеквадратическое отклонение угла стрельбы равно 0,05. Пушка выстреливает 100 тысяч раз. При заданных исходных получаем среднюю дальность выстрела равную 4984.6425517, вероятность попадания в мишень 0.01532, математическое ожидание = 4987.3078685 и дисперсию = 260715.046407.

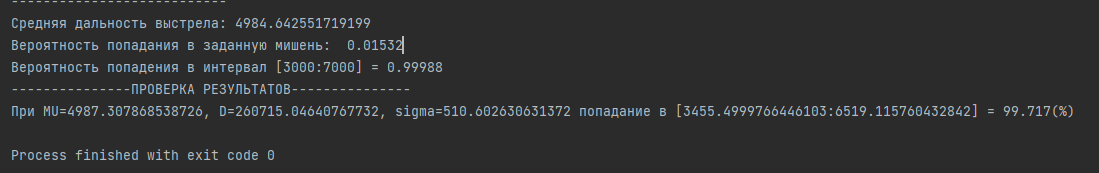


Рисунок 13 – Результаты моделирования

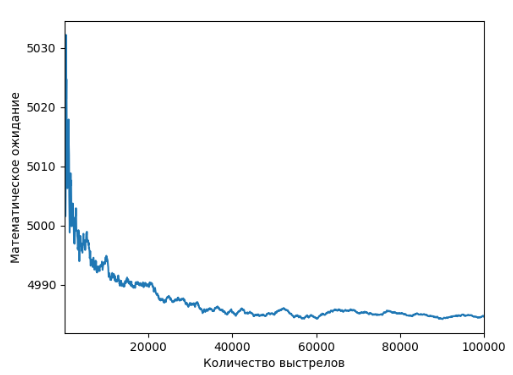


Рисунок 14 – График зависимости математического ожидания от числа испытаний

Значения оптимальной точности математического ожидания от количества экспериментов находится в районе 55 тысяч испытаний.

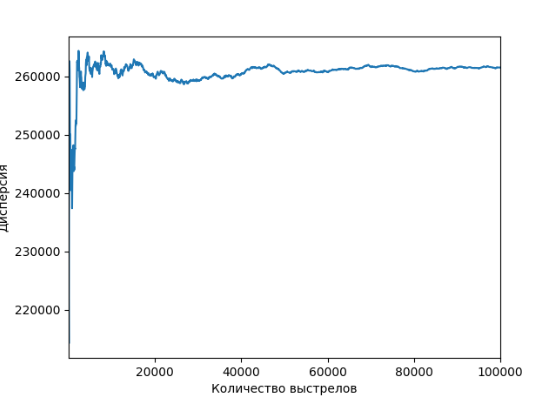


Рисунок 15 - График зависимости дисперсии от числа выстрелов

Значение оптимальной точности дисперсии от количества экспериментов достигается примерно на 55 тысячах экспериментах.

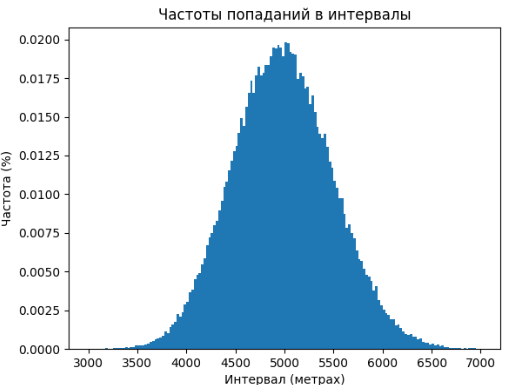


Рисунок 16 – Гистограмма распределения случайной величины Xi

За деления брались интервалы в 25 метров, начиная с 3000 метров и заканчивая 7000 метрами. Из гистограммы видно, что наибольшее количество снарядов упало в интервал [5050, 5075) метров. Чем дальше от центра математического ожидания, тем меньше вероятность попадания снаряда в интервал. Проверив результаты правилом трёх sigma, можно увидеть, что 99.717% снарядов попали в интервал [3455.4999766446103:6519.115760432842], что подтверждает правильность результатов работы алгоритма моделирования.

**Заключение:**

В данной лабораторной работе были достигнуты следующие результаты:

* разработан программный алгоритм, позволяющий проводить моделирование стрельбы из пушки методом Монте-Карло;
* данный алгоритм реализован на языке программирования Python;
* смоделированы различные ситуации, задавая соответствующим образом входные параметры написанной программе;
* проанализированы результаты моделирования;
* составлены графики и гистограммы зависимостей величин от числа испытаний;
* проверена правильность работы алгоритма.