

МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РОССИЙСКОЙ
ФЕДЕРАЦИИ
ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ АВТОНОМНОЕ ОБРАЗОВАТЕЛЬНОЕ
УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ
«Национальный исследовательский ядерный университет «МИФИ»
Обнинский институт атомной энергетики
Отделение интеллектуальных кибернетических систем

Лабораторная работа №1

Выполнила студентка
Группы ИС-Б17
Отделения ИКС
Петренко В. Ю.
Проверила:
профессор, д.т.н. Гулина О. М.

Обнинск, 2020

1. Алгоритм

Использовала метод середины квадратов.

Формула заполнения массива псевдослучайными числами:

$$y_{i+1} = 10^{-k} \Pi \left(10^k D \left((1 - y_i)^3 10^k \right) \right)$$

Формула хи-квадрат распределения (критерий Пирсона): $X^2 =$

$$\sum_{i=1}^r \frac{(m - np)^2}{np}$$

(Использовала python 3.6)

1. k=2500 #число элементов
2. r=10 #кол-во интервалов
3. p=0.1 #теоретическая вероятность попадания в каждый интервал
4. array=[] #массив псевдослучайных чисел
5. l_aper=0 #длина апериодичности
6. l_per=0 #длина периода
7. p_i =[] #количество попаданий в каждый интервал
8. X2=0 #хи-вквадрат
- 9.
10. def fraction(x):
11. # функция для расчета дробной части
12. return x - int(x)
- 13.
14. def fillArray():
15. # функция для заполнения массива
16. y0=float(input("Введите гамма-нулевое: "))
17. accrs=int(input("Введите количество знаков после запятой: "))
18. for i in range(k):
19. array.append(y0)
20. y0=(10 ** -accrs)*int((10 ** accrs)*fraction(float(((1-y0) ** 3)*(10 ** accrs)))) #метод середины квадратов
21. print("Массив заполнен псевдослучайными числами.")
- 22.
23. def periodLength():
24. global l_aper, l_per
25. print("Определение длины периода и апериодичности.")
26. flag=True #пока в последовательности будут одинаковые элементы
27. for i in range(k):

```

28.         for j in range(i+1, k):
29.             if(abs(array[i]-array[j])<0.00000001):#сравниваем
30.                 print("Совпадение в ", i, "-ом и ", j, "-ом элементах:
    ", array[i], " и ", array[j])
31.                 l_aper = j
32.                 l_per = j-i
33.                 flag=False
34.                 if not flag:
35.                     break
36.                 if flag:
37.                     #если нет одинаковых элементов, длина
    апериодичности = длине последовательности
38.                     l_aper=k
39.                     l_per=0
40.                 if not flag:
41.                     break
42.         print("Период: ", l_per)
43.         print("Апериодичность: ", l_aper)
44.
45.     def calc_pi():
46.         print("Расчет количества попаданий в каждый
    интервал.")
47.         for i in range(r):
48.             p_i.append(0)
49.         print("[ ", end = ' ')
50.         for i in range(r):
51.             for j in range(l_aper):
52.                 if (array[j]>(i*p) and array[j]<((i+1)*p)):
53.                     p_i[i]+=1
54.         for i in range(r):
55.             print(p_i[i], end = ' ')
56.         print(" ] ")
57.
58.     def calc_X2():
59.         print("Расчет X2.")
60.         X2 = 0
61.         for i in range(r):
62.             X2+=((p_i[i]-(l_aper*p)) ** 2)/(l_aper*p)
63.         print("X2 = ", X2)
64.
65.     def show():

```

```

66.     n=int(input("Вывести последовательность до: "))
67.     if n>k:
68.         n=k
69.     for i in range(n):
70.         print(array[i], end = ', ')
71.
72.     print("Программа генерирует псевдослучайную
последовательность из 2500 элементов на интервале (0;1)\n")
73.     print("Затем рассчитывает длины периода и аperiodичности
и вероятность попадания в каждый интервал (10 интервалов)\n")
74.     print("После чего считает X-квадрат для
последовательности длиной аperiodичности.")
75.     fillArray()
76.     periodLength()
77.     calc_pi()
78.     calc_X2()
79.     show()

```

2. Результаты

k=2

№ эксперимен та	γ_0	P	L
1	0,12	1	22
2	0,34	1	15
3	0,56	1	12
4	0,63	1	6
5	0,88	1	2

k=4

№ эксперимен та	γ_0	P	L

1	0,1234	102	124
2	0,1111	102	118
3	0,1222	102	159
4	0,8765	102	127
5	0,5678	102	130

k=6

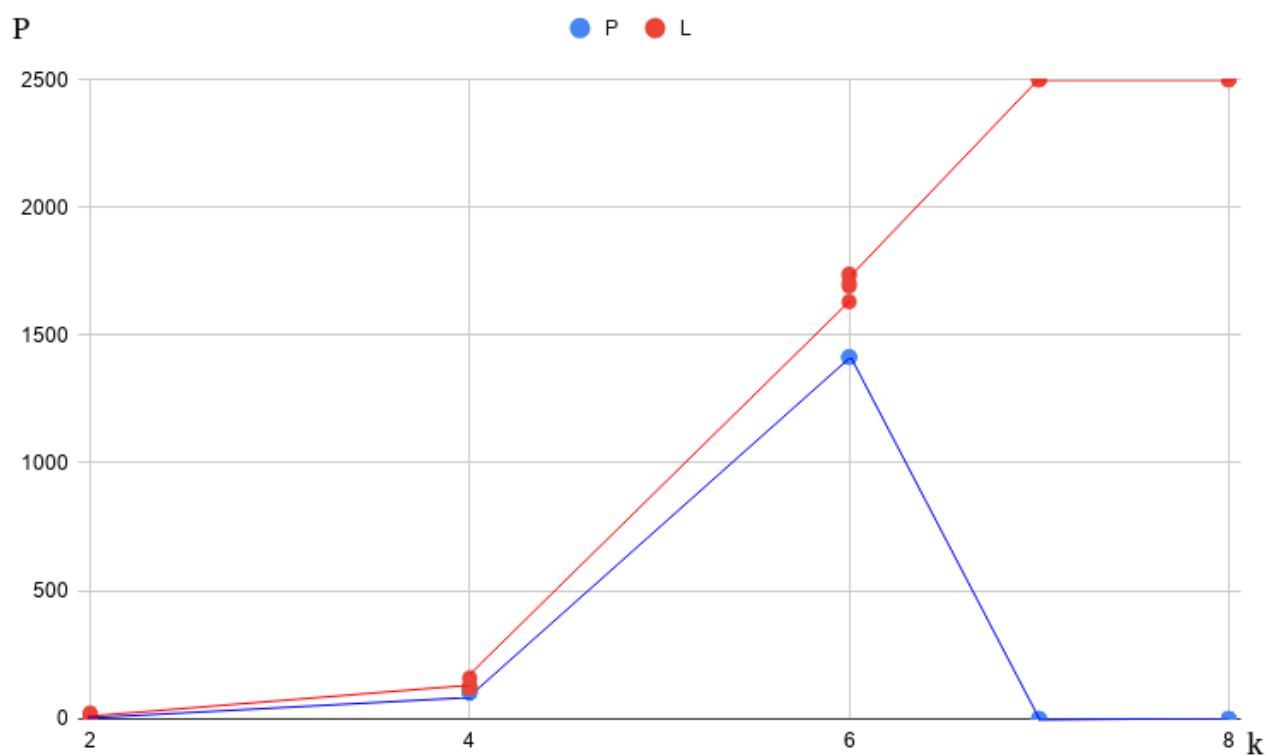
№ эксперимен та	γ_0	P	L
1	0.234567	1414	1739
2	0.342365	1414	1631
3	0.236790	1414	1704
4	0.456456	1414	1734
5	0.512345	1414	1692

k=7

№ эксперимен та	γ_0	P	L
1	0.1234567	0	2500
2	0.2312453	0	2500
3	0.2457654	0	2500
4	0.4562853	0	2500
5	0.8673945	0	2500

k=8

№ эксперимен та	γ_0	P	L
1	0.13243546	0	2500
2	0,45347234	0	2500
3	0,54657629	0	2500
4	0,56428546	0	2500
5	0,96542378	0	2500

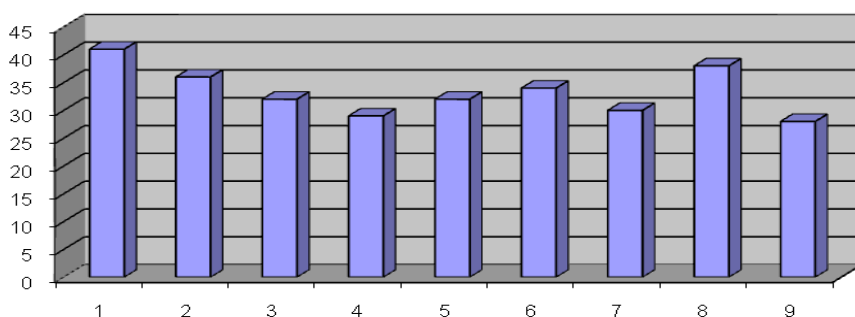


Проверка по критерию Х-квадрат.

γ_{0i}	χ^2	Вывод
0,00012531	4,5	данная последовательность распределена по равномерному закону с точностью примерно 50%
0,23540014	8,52	данная последовательность распределена по равномерному закону с точностью примерно 10%
0,39554201	6,06	данная последовательность распределена по равномерному закону с точностью примерно 40%
0,87126303	13,5	данная последовательность распределена по равномерному закону с точностью примерно 95%
0,91001369	5,16	данная последовательность распределена по равномерному закону с точностью примерно 20%

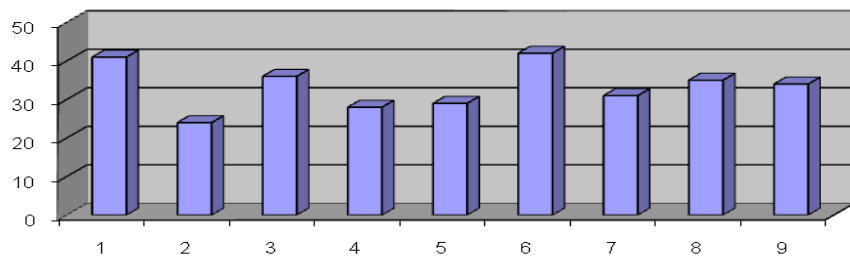
Случай 1. При $\gamma_0 = \mathbf{0,00012531}$

41	36	32	29	32	34	30	38	28
1	2	3	4	5	6	7	8	9



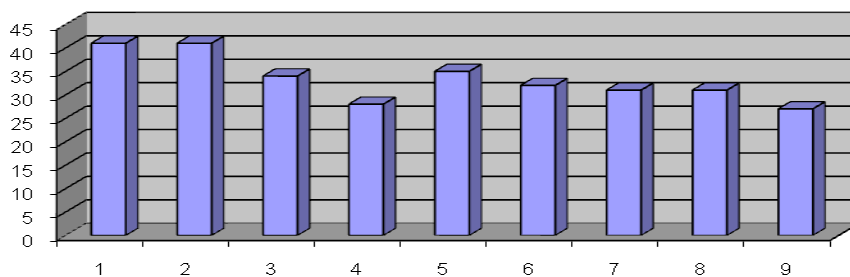
Случай 2. При $\gamma_0 = \mathbf{0,23540014}$

41	24	36	28	29	42	31	35	34
1	2	3	4	5	6	7	8	9



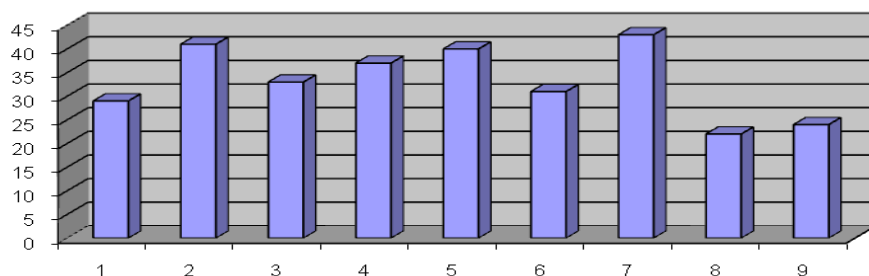
Случай 3. При $\gamma_0 = \mathbf{0,39554201}$

41	41	34	28	35	32	31	31	27
1	2	3	4	5	6	7	8	9



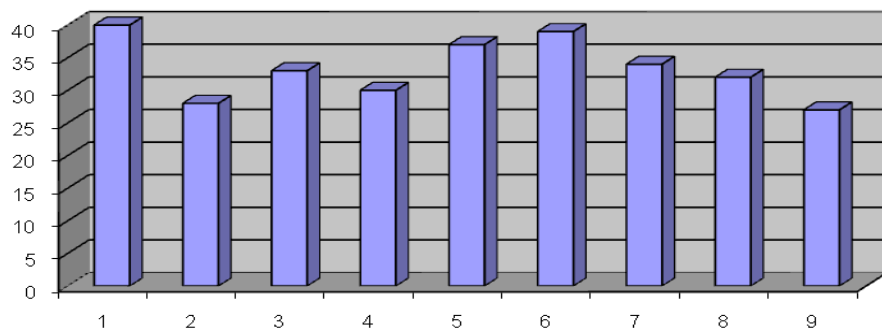
Случай 4. При $\gamma_0 = \mathbf{0,87126303}$

29	41	33	37	40	31	43	22	24
1	2	3	4	5	6	7	8	9



Случай 5. При $\gamma_0 = \mathbf{0,91001369}$

40	28	33	30	37	39	34	32	27
1	2	3	4	5	6	7	8	9



3. Выводы: При увеличении количества знаков после запятой растут длины периода и аperiodичности. Гипотеза о соответствии равномерному закону распределения по значениям χ^2 подтвердилась лишь в некоторых случаях.