Санкт-Петербургский политехнический университет Петра Великого  
Институт компьютерных наук и технологий  
Высшая школа программной инженерии

Лабораторная работа №1

По дисциплине «Статистическое моделирование»

Выполнил студент гр 33534/5  Донцов А. Д.

Руководитель В. В. Чуркин

Санкт-Петербург  
2019 г.

# Цель работы

1. Получение на ЭВМ с помощью программного датчика базовой последовательности псевдослучайных чисел, имеющих равномерное распределение.
2. Освоение методов статистической оценки полученного распределения: вычисление эмпирических значений для математического ожидания и дисперсии.
3. Освоение методов оценки статистики связи: вычисление значений автокорреляционной функции и построение коррелограммы.
4. Освоение методов графического представления законов распределения: построение функции плотности распределения и интегральной функции распределения.

# Результаты работы программы

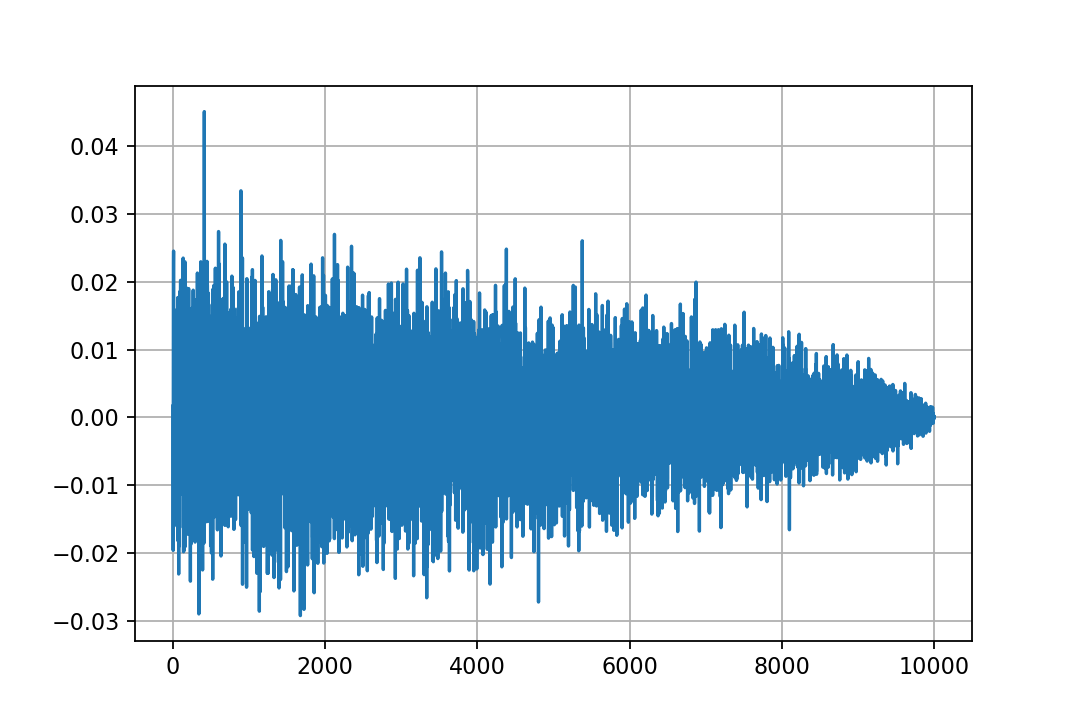
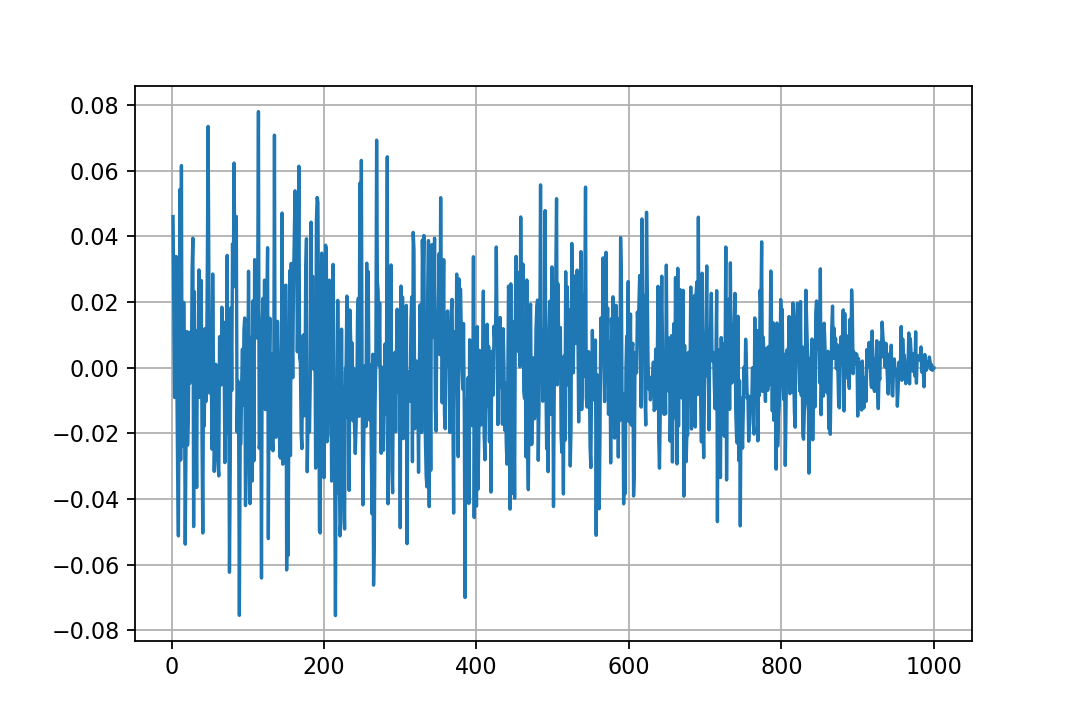
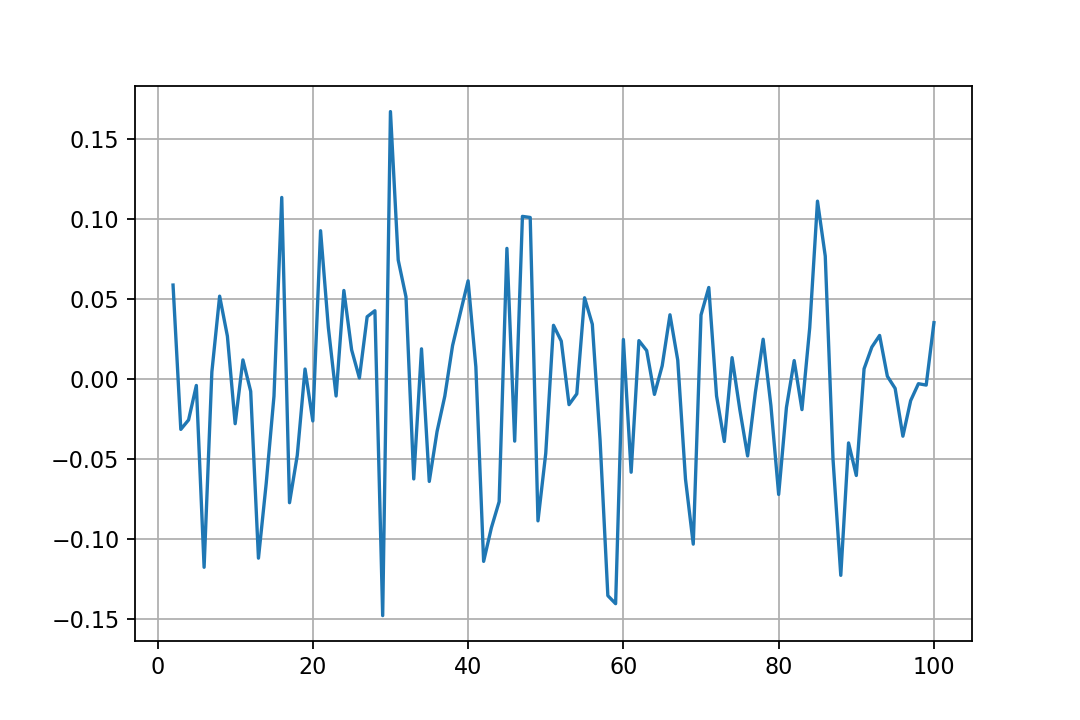
Код программы приведен в приложении 1.

Были вычислены эмирические значения математического ожидания и дисперсии для сгенерированных распределений величиной 100, 1000, 10000 чисел, получены следующие результаты:

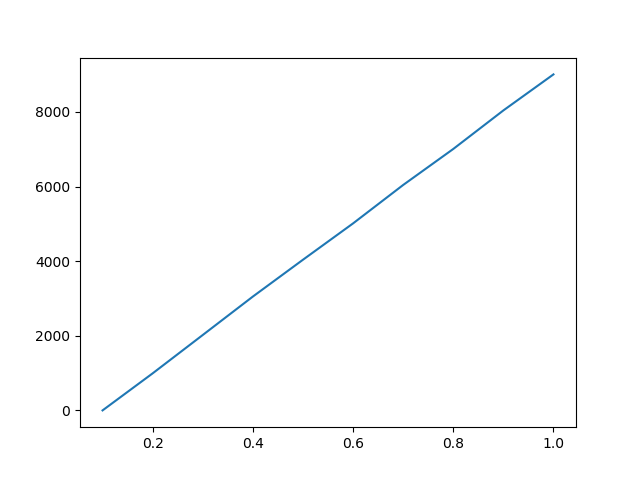
|  |  |  |  |
| --- | --- | --- | --- |
| N | Оценка распределения | Экспериментальные данные | Теоретические данные |
| 100 | M | 0.5033546875750039 | 0.5 |
| D | 0.08240335390263516 | 0.083333 |
| 1000 | M | 0.4943025338106951 | 0.5 |
| D | 0.08612892709387415 | 0.083333 |
| 10000 | M | 0.501566281006014 | 0.5 |
| D | 0.08300784397318597 | 0.083333 |

Из данного представления видно, что эмпирические данные имеют небольшое отклонение от теоретических.

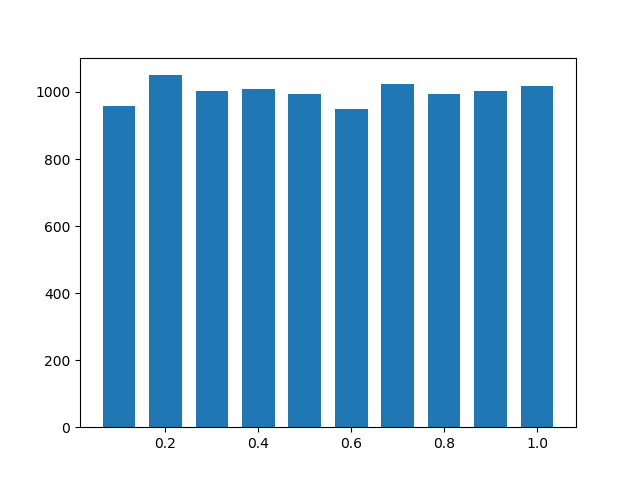
Были получены коррелограммы для распределений:



Был построен график интегральной функции плотности:



Была построена гистограмма плотности распределения:



# Вывод

В ходе работы были получены псевдослучайные равномерные последовательности различной длины. Для данных последовательностей были вычислены математическое ожидание, дисперсия, значение функции автокорреляции.

Были построены корреллограммы для полученных распределений, график интегральной функции плотности и гистограмма плотности распределения.

Из полученных результатов можно сделать вывод, что генератор псевдослучайных чисел numpy.random.uniform() подходит для решения задач.

# Текст программы:

from matplotlib import pyplot as plt  
import numpy as np  
  
  
def dens\_dst(dst, step, pk\_id):  
 pr = [0 for i in range(int(1/step))]  
 steps = []  
 steps = np.arange(0.0, 1.1, step)  
 for p in range(len(steps)-1):  
 for i in dst:  
 if p==0:  
 if i >= steps[p] and i<steps[p+1]:  
 pr[p]+=1  
 else:  
 if i > steps[p] and i <= steps[p+1]:  
 pr[p]+=1  
 cn = 1  
 ff = []  
 for i in pr:  
 ff.append(sum(pr[1:cn]))  
 cn += 1  
 plt.figure()  
 plt.plot(steps[1::], ff)  
 plt.savefig('integral\_function.png')  
 ### histogram  
 plt.figure()  
 plt.bar(steps[1::], pr, width=0.07)  
 plt.savefig('hist.png')  
 vizual\_rez(pr, pk\_id, steps[0:len(steps)-1])  
  
  
def math\_waiting(dst):  
 summ = 0.0  
 for i in dst:  
 summ += i  
 return summ/len(dst)  
  
  
def dispersion(dst, m\_w):  
 summ = 0.0  
 for i in dst:  
 summ += (i-m\_w)\*\*2  
 return summ/len(dst)  
  
  
def korrel(dst, m\_w, graph\_counter):  
 f = 1  
 k = []  
 summ2 = 0.0  
 summ1 = 0.0  
 for i in dst:  
 summ2 += (i-m\_w)\*\*2  
 for f in range(len(dst)):  
 summ1 = 0.0  
 for i in range(len(dst)-f):  
 summ1 += (dst[i]-m\_w)\*(dst[i+f]-m\_w)  
 k.append(summ1/summ2)  
 vizual\_rez(k, graph\_counter)  
   
  
def vizual\_rez(sol\_array, graph\_counter , plt\_list = []):  
 counter = [i+1 for i in range(len(sol\_array))]  
 fig = plt.figure(dpi=160, figsize=(1080/160, 720/160))  
 if graph\_counter > 0:   
 plt.grid()  
 plt.plot(counter[1::], sol\_array[1::])  
 filename = 'Кореллограмма' + str(graph\_counter) + '.png'  
 fig.savefig(filename)  
 else:  
 plt.grid()  
 plt.plot(plt\_list, sol\_array)  
 filename = 'плотность' + str(graph\_counter) + '.png'  
 fig.savefig(filename)  
   
  
distr\_cnt = [100, 1000, 10000]  
gk = 1  
pk = -1  
for i in distr\_cnt:  
 distr = np.random.uniform(0, 1, i)  
 print('Мат ожидание '+str(math\_waiting(distr)))  
 print('Дисперсия '+str(dispersion(distr, math\_waiting(distr))))  
 dens\_dst(distr, 0.1, pk)  
 korrel(distr, math\_waiting(distr), gk)  
 distr.fill(0)  
 gk += 1  
 pk -= 1  
   
#plt.show()