**Федеральное государственное бюджетное образовательное учреждение высшего образования**

**"Уфимский государственный авиационный технический университет"**

**Кафедра** Высокопроизводительных вычислительных технологий и систем

**Дисциплина:** Математическая статистика

**Отчет по лабораторной работе № 4**

**Тема:** «Применение метода статистических испытаний к моделированию винеровского процесса»

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Группа ПМ-453 | Фамилия И.О. | Подпись | Дата | Оценка |
| Студент | Шамаев И.Р. |  |  |  |
| Принял | Маякова С.А. |  |  |  |

**Уфа 2022**

**Практическая часть**

Работа выполнена согласно варианту №13.

**Задание**

Реализовать модель аддитивных случайных блужданий , где с начальным приближением *x*0**=***N* для моментов времени t=1, 2, 3, …15+*N,* где *N –* номер по списку.

Длягенерации нормально распределенных случайных величин в модели (1.38) использовать: а) метод сумм, б) метод Бокса-Миллера.

Провести 100 испытаний, в каждом из которых рассчитывается последовательность *x*1, *x*2, … *x*15+*N*. По результатам испытанийна каждом временном шаге *t*=1, 2, 3, …15+*N* вычислить выборочное среднее значение

и волатильность

Построить графики 5 произвольных выборочных траекторий случайного блуждания, сгенерированных по модели. Рядом с ними построить график одной траектории для модели с подставленной винеровской переменной.

Построить уравнения регрессии выборочного среднего и волатильности как функций от времени. Для выборочного среднего уравнение выбирать из класса линейных функций, а для волатильности – из класса функций . Насколько точно экспериментальные точки ложатся на графики полученных уравнений регрессии?

**Решение:**

Задание 1

Генерируем выборку из n=1000 равномерно распределённых случайных чисел. Затем строим её гистограмму и поверх неё строим теоретическую плотность распределения

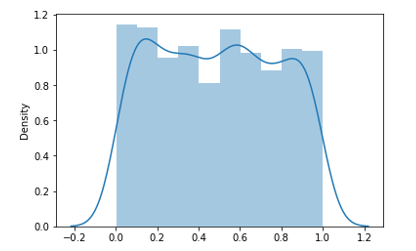
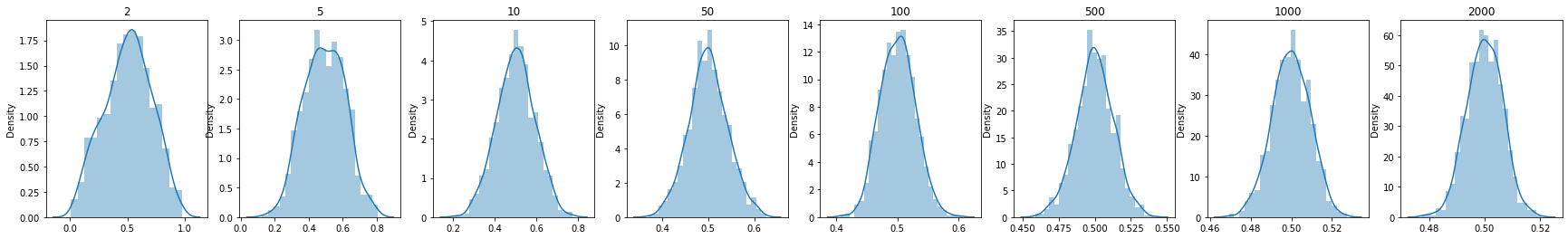


Рисунок 1. Гистограмма выборки из 1000 равномерно распределённых случайных чисел и теоретическая плотность распределения.

Таким образом, из полученного изображения следует, что числа подчиняются равномерному закону распределения.

Далее при четырех и более значениях n (n = 2, 5, 10, 50, 100, 500, 1000, 2000) сгенерируем 1000 выборок объёма n и построим гистограммы распределений их выборочных средних.

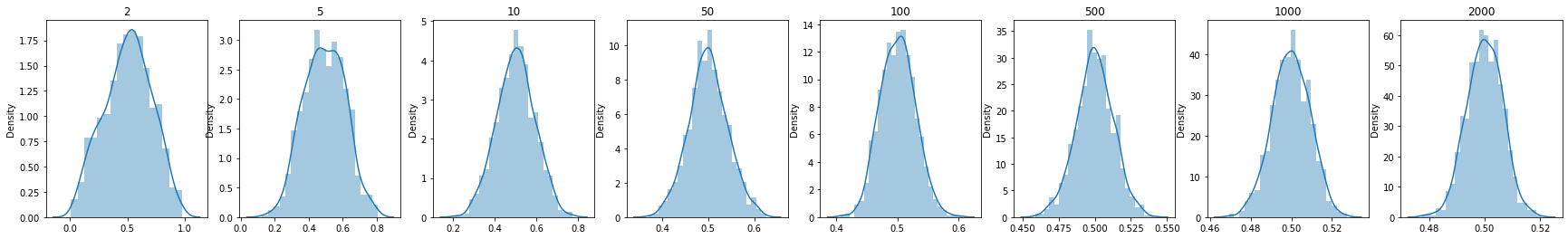


Рисунок 2. Гистограммы распределений выборочных средних выборок объёма

Таким образом, из полученного изображения следует, что у выборочных средних выборок нормальное распределение.

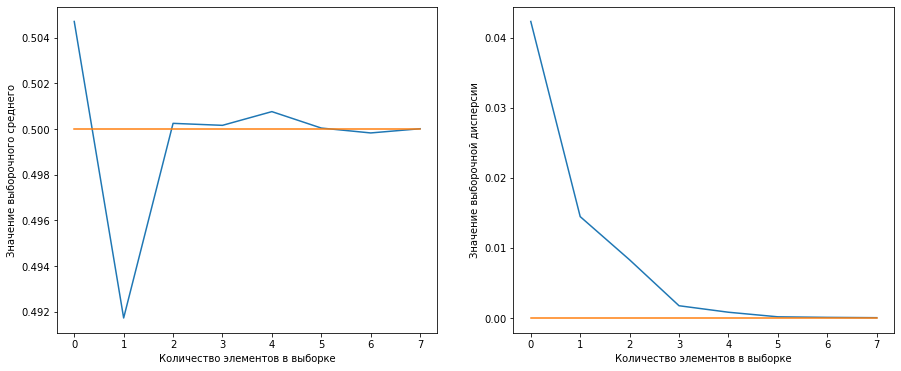
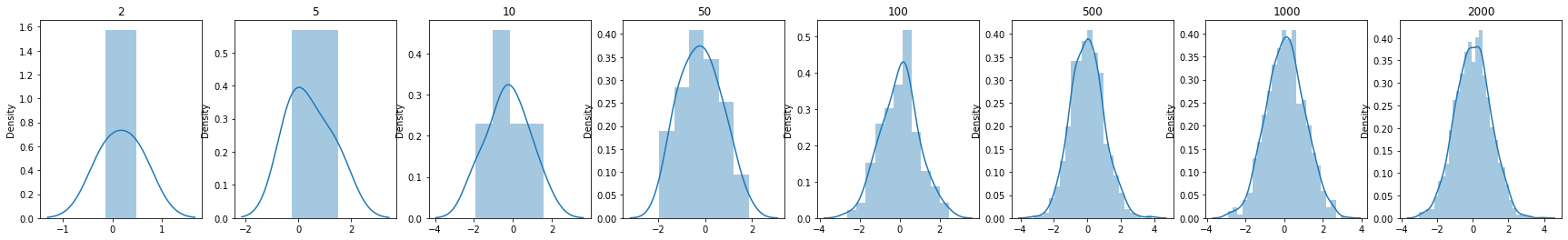


Рисунок 3- Графики изменения выборочного среднего и дисперсии

Далее реализуем метод Бокса-Мюллера в полярной форме. Сгенерируем несколько выборок различного объёма (100, 500, 1000) по методу Бокса-Мюллера, построим гистограмму выборки и нарисуем поверх неё теоретическую плотность распределения нашей случайной величины.

****

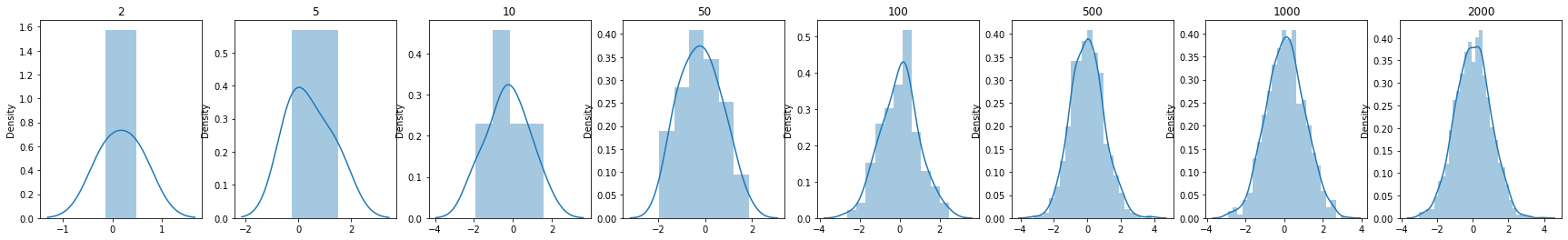
****

Рисунок 4. Гистограммы выборок различного объёма и теоретическая плотность распределения.

Таким образом, из полученного изображения следует, что при увеличении объёма выборки гистограмма всё больше напоминает график теоретической плотности нормального распределения.

Задание 2

Теперь реализуем модель аддитивных случайных блужданий с начальным приближением для моментов времени , где . Параметр .

Далее были проведены 100 испытаний и для каждого момента времени рассчитаны выборочное среднее и волатильность. Построим их графики и гистограммы:

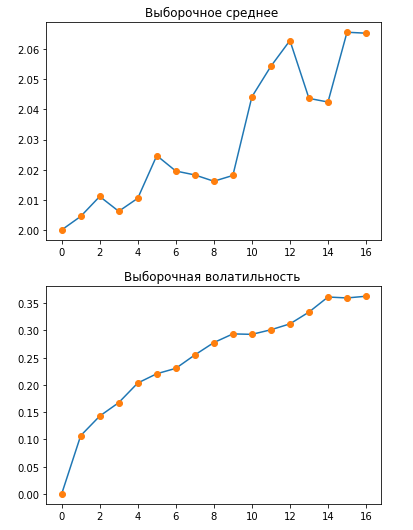


Рисунок 5. Графики выборочного среднего и волатильности аддитивных случайных блужданий

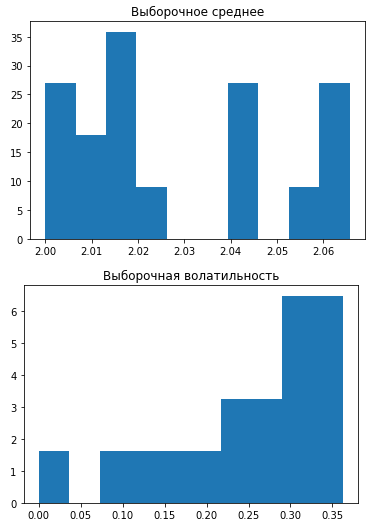


Рисунок 6. Гистограммы выборочного среднего и выборочной волатильности аддитивных блужданий.

Таким образом, из полученного изображения следует, что у выборочного среднего аддитивных блужданий гистограмма напоминает график плотности нормального распределения, а у выборочной волатильности аддитивных блужданий гистограмма напоминает график плотности экспоненциального распределения.

Теперь для случая винеровской переменной.

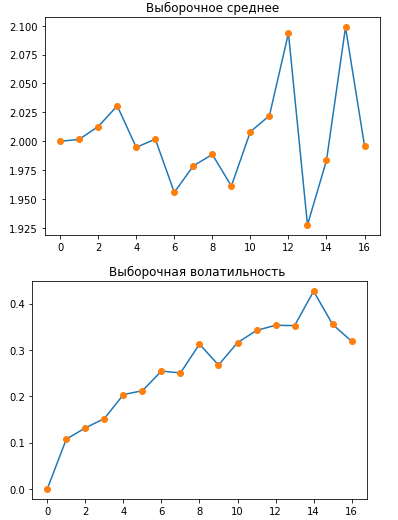


Рисунок 7. Графики выборочного среднего и выборочной волатильности винеровских аддитивных блужданий.

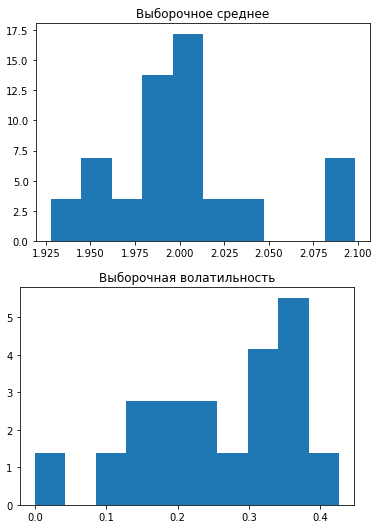


Рисунок 8. Гистограммы выборочного среднего и выборочной волатильности винеровских аддитивных блужданий.

Таким образом, из полученных изображений следует, что гистограмма выборочных средних винеровских аддитивных блужданий напоминает график плотности нормального распределения, а гистограмма волатильности винеровских аддитивных блужданий напоминает график плотности экспоненциального распределения.

Далее построим уравнения регрессии для выборочного среднего и волатильности как функций от времени. Для выборочного среднего уравнение выберем из класса линейных функций, а для волатильности – из класса функций .

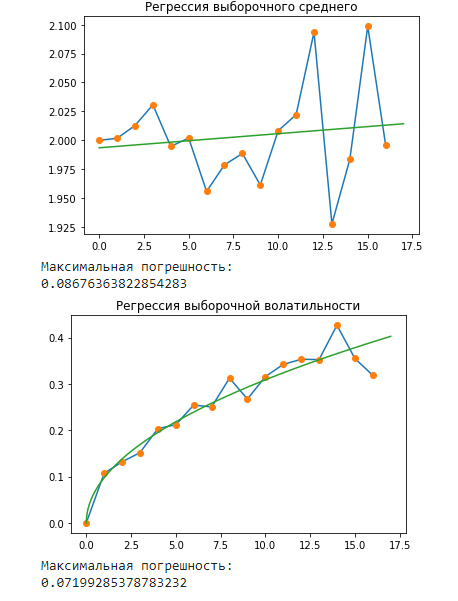


Рисунок 9. Графики регрессий выборочного среднего и выборочной волатильности винеровских аддитивных блужданий.

Максимальная по модулю ошибка в случае выборочной средней винеровских аддитивных блужданий = 0.087, а в случае выборочной волатильности – 0.072.

**Задание 3**

Была реализована модель аддитивных случайных блужданий с начальным приближением для моментов времени , где -номер по списку.

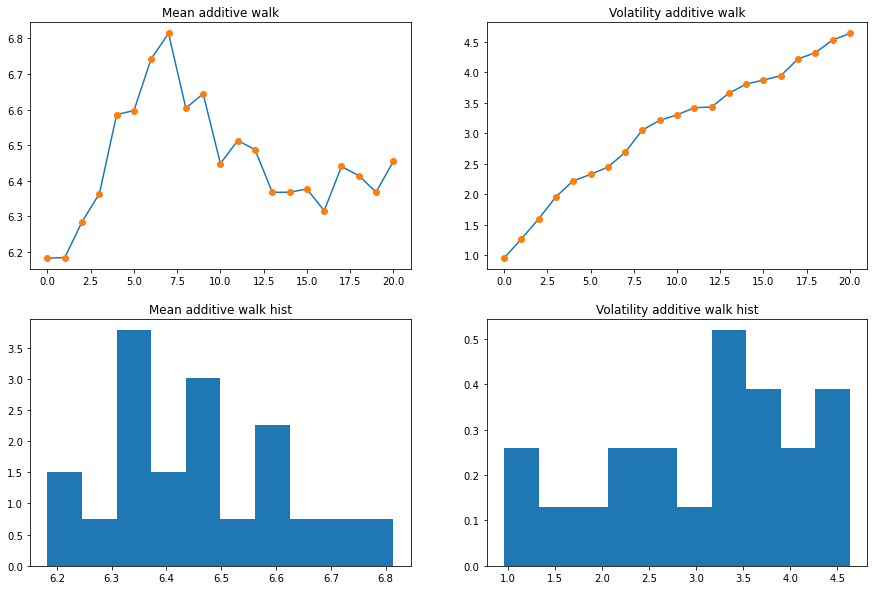
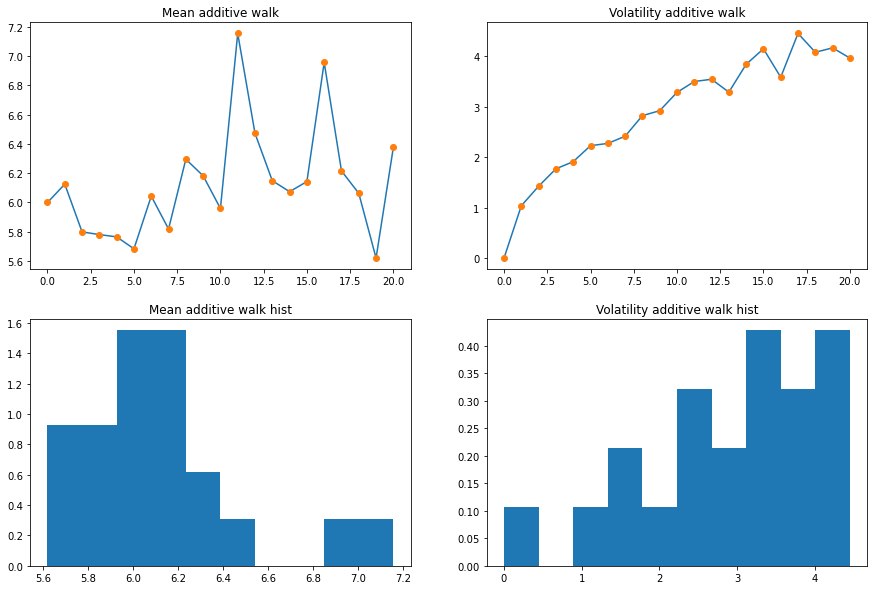
Для графиков среднего и дисперсии случая с винеровской переменной были построены уравнения линейной и из класса регрессий.

Рисунок 5 - Графики среднего, дисперсии, а также гистограммы распределения среднего и дисперсии для случая с винеровской переменной

Рисунок 6 - Графики среднего, дисперсии, а также гистограммы распределения среднего и дисперсии для модели аддитивных случайных блужданий

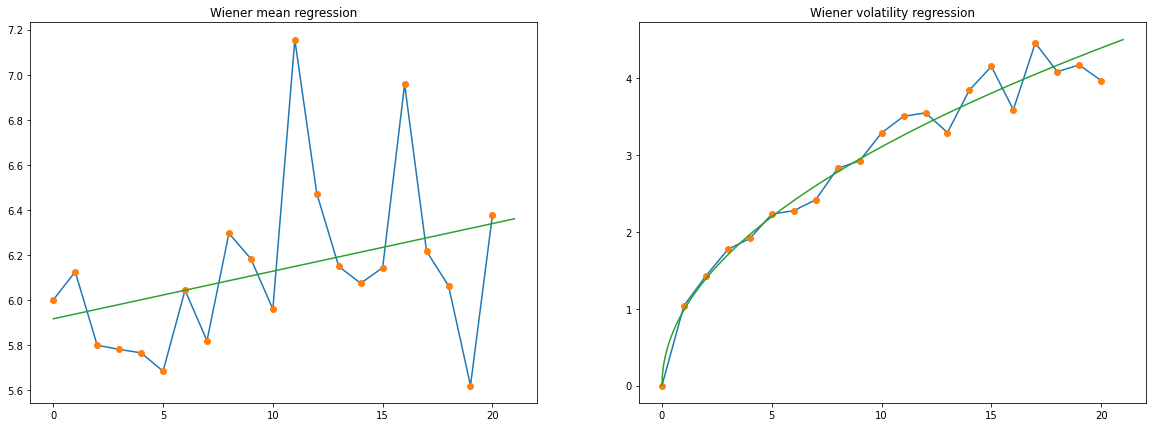


Рисунок 7 - Графики регрессий для значений среднего и волатильности (дисперсии)

**Вывод**: в ходе данной лабораторной работы был получен навык моделирования винеровского случайного процесса с помощью метода статистических испытаний. Сгенерированы СВ методами сумм и Бокса-Миллера, так же проведено 100 испытаний. Было построено по 5 выборочных траекторий блуждания. Построены регрессии выборочного среднего и волатильности как функций от времени, получена точность

**Приложение**

Листинг программы:

#include <iostream>

#include <ctime>

#include <fstream>

using namespace std;

void SumMethod(double\*\*);

void BoxMillerMethod(double\*\*);

double Random(double, double);

double RndG();

double NormalDistribution();

void RegrSr(double\*);

void RegrW(double\*);

void Graph(string, double\*\*, double\*);

const int N0 = 13;

const int T = 28;

int main()

{

double\*\* x = new double\* [100];

for (int i = 0; i < 100; i++)

{

x[i] = new double[T + 1];

}

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

{

x[0][i] = N0;

}

SumMethod(x);

//BoxMillerMethod(x);

// Мат. ожидания

double\* xMiddle = new double[T + 1];

for (int t = 0; t <= T; t++)

{

xMiddle[t] = 0.;

for (int i = 0; i < 100; i++)

{

xMiddle[t] += x[i][t];

}

xMiddle[t] /= 100.;

}

RegrSr(xMiddle);

for (int t = 0; t <= T; t++)

{

cout << "M[" << t << "] = " << xMiddle[t] << endl;

}

cout << endl;

double\* sigma = new double[T + 1];

for (int t = 0; t <= T; t++)

{

sigma[t] = 0.;

for (int i = 0; i < 100; i++)

{

sigma[t] += (x[i][t] - xMiddle[t]) \* (x[i][t] - xMiddle[t]);

}

sigma[t] /= 100.;

sigma[t] = sqrt(sigma[t]);

}

RegrW(sigma);

for (int t = 0; t <= T; t++)

{

cout << "Sigma[" << t << "] = " << sigma[t] << endl;

}

return 0;

}

void SumMethod(double\*\* x)

{

srand(time(NULL));

double\* xt = new double[T + 1];

double\* xW = new double[T + 1];

double\* eps = new double[T + 1];

xt[0] = N0;

for (int i = 1; i < 100; i++)

{

for (int t = 0; t <= T; t++)

{

eps[t] = NormalDistribution();

xt[t] = (double)N0;

xW[t] = (double)N0 + NormalDistribution() \* sqrt(t);

}

for (int t = 0; t <= T; t++)

{

for (int tt = 0; tt <= t - 1; tt++)

{

xt[t] += eps[tt];

}

x[i][t] = xt[t];

}

}

ofstream file("Sum.txt");

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

{

file << i << "\t" << x[1][i] << "\t" << x[20][i] << "\t" << x[40][i] << "\t" << x[60][i] << "\t" << x[80][i] << "\t" << x[99][i] << "\t" << xW[i] << endl;

}

file.close();

}

void BoxMillerMethod(double\*\* x)

{

srand(time(NULL));

double\* xt = new double[T + 1];

double\* xW = new double[T + 1];

xt[0] = N0;

double\* eps = new double[T + 1];

for (int i = 0; i < 100; i++)

{

for (int t = 0; t <= T; t++)

{

eps[t] = RndG();

xt[t] = (double)N0;

xW[t] = (double)N0 + RndG() \* sqrt(t);

}

for (int t = 0; t <= T; t++)

{

for (int tt = 0; tt <= t - 1; tt++)

{

xt[t] += eps[tt];

}

x[i][t] = xt[t];

}

}

ofstream file("BM.txt");

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

{

file << i << "\t" << x[1][i] << "\t" << x[20][i] << "\t" << x[40][i] << "\t" << x[60][i] << "\t" << x[80][i] << "\t" << x[99][i] << "\t" << xW[i] << endl;

}

file.close();

}

double RndG() //Gauss random var

{

double r = 0.;

double s, v1, v2;

while (true)

{

v1 = Random(-1., 1.);

v2 = Random(-1., 1.);

s = v1 \* v1 + v2 \* v2;

if ((s < 1. && s != 0.))

{

r = v1 \* sqrt(-2. \* log(s) / s);

break;

}

}

return r;

}

double NormalDistribution()

{

double S = 0.;

for (int j = 0; j < 12; j++)

{

S += (double)rand() / RAND\_MAX;

}

return S - 6.;

}

double Random(double min, double max)

{

return (double)(rand()) / RAND\_MAX \* fabs(max - min) + min;

}

void RegrSr(double\* xMiddle)

{

double X\_Middle = 0.;

double Y\_Middle = 0.;

double XY\_Middle = 0.;

double XX\_Middle = 0.;

double a = 0., b = 0.;

//sample mean for x, y, x\*x, y\*x

for (int t = 0; t < T; t++)

{

X\_Middle += t; //x

Y\_Middle += xMiddle[t]; //y

XY\_Middle += t \* xMiddle[t]; //x\*y

XX\_Middle += t \* t; //x\*x

}

X\_Middle /= T;

Y\_Middle /= T;

XY\_Middle /= T;

XX\_Middle /= T;

a = (X\_Middle \* Y\_Middle - XY\_Middle) / (X\_Middle \* X\_Middle - XX\_Middle);

b = (XY\_Middle - a \* XX\_Middle) / X\_Middle;

ofstream file("RegMid.txt");

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

{

file << i << "\t" << xMiddle[i] << "\t" << a \* i + b << endl;

}

file.close();

}

void RegrW(double\* sig)

{

double a;

double sumS0T = 0.;

double sumT = 0.;

for (int t = 0; t < T; t++)

{

sumS0T += sig[t] \* sqrt(t);

sumT += t;

}

a = sumS0T / sumT;

ofstream file("RegW.txt");

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

{

file << i << "\t" << sig[i] << "\t" << a \* sqrt(i) << endl;

}

file.close();

}