**ЛАБОРАТОРНАЯ РАБОТА N 3**

СПОСОБЫ ПОЛУЧЕНИЯ СЛУЧАЙНЫХ ЧИСЕЛ С ТРЕБУЕМЫМ ЗАКОНОМ РАСПРЕДЕЛЕНИЯ

(Часть 2.Непрерывные случайные величины)

Выполнил

студент гр. 33504/2 (подпись) Лелюхин Д. О.

Руководитель (подпись) Чуркин В.В.

# Оглавление

[Оглавление 2](#_Toc483942926)

[ЦЕЛЬ РАБОТЫ 3](#_Toc483942927)

[ПОРЯДОК ПРОВЕДЕНИЯ РАБОТЫ 3](#_Toc483942928)

[РАВНОМЕРНОЕ РАСПРЕДЕЛЕНИЕ 3](#_Toc483942929)

[(дискретное) 3](#_Toc483942930)

[*Код программы:* 3](#_Toc483942931)

[*Результаты работы программы:* 4](#_Toc483942932)

[НОРМАЛЬНОЕ РАСПРЕДЕЛЕНИЕ 4](#_Toc483942933)

[*Код программы* 4](#_Toc483942934)

[ЭКСПОНЕНЦИАЛЬНОЕ РАСПРЕДЕЛЕНИЕ 5](#_Toc483942935)

[*Результаты работы программы* 6](#_Toc483942936)

[ХИ-КВАДРАТ РАСПРЕДЕЛЕНИЕ 6](#_Toc483942937)

[*Код программы* 6](#_Toc483942938)

[РАСПРЕДЕЛЕНИЕ СТЬЮДЕНТА 7](#_Toc483942939)

[*Код программы* 7](#_Toc483942940)

[*Результаты работы программы* 7](#_Toc483942941)

# ЦЕЛЬ РАБОТЫ

1. Практическое освоение методов получения случайных величин, имеющих дискретный характер распределения.
2. Разработка программных датчиков дискретных случайных величин.
3. Исследование характеристик моделируемых датчиков:
   1. Оценка точности моделирования: вычисление математического ожидания и дисперсии, сравнение полученных оценок с соответсвующими теоретическими значениями.
4. Гpафическое пpедставление функции плотности распределения и интегральной функции pаспpеделения.

## ПОРЯДОК ПРОВЕДЕНИЯ РАБОТЫ

1. Написать и отладить подпрограммы получения дискретных псевдослучайных чисел в соответствии с алгоритмами, приведенными в описании.
2. Осуществить проверку точности моделирования полученных датчиков псевдослучайных чисел.
3. Отлаженные подпрограммы собрать в единый пакет "Дискретные распределения" и создать в головной программе "меню" для всего пакета

## РАВНОМЕРНОЕ РАСПРЕДЕЛЕНИЕ

## (дискретное)

### *Код программы:*

clc

clearvars

syms a b n;

a=1;

b=5;

n=10000;

Mteor=(a+b)/2;

Dteor=((b-a)^2)/12;

r=zeros(1,n);

for i=1:n

r(i) = RNUNI(a,b);

end

x=unique(r);

fx = zeros(1,length(x));

Fx = zeros(1,length(x));

for i=1:length(x)

fx(i) = 1/(b-a);

if (a<=x(i)<=b)

Fx(i)=(x(i)-a)/(b-a);

elseif (x(i)<a)

Fx(i)=0;

elseif (x(i)>b)

Fx(i)=1;

else

continue;

end

end

set(gcf, 'Position', [300, 100, 800, 600]);

%Построение графика функции плотности

subplot(2,1,1);

histogram(r,'Normalization','probability','NumBins',20);

grid on;

xlim([a,b]); ylim([0,1]);

title('График функции плотности f(x)','fontsize',15);

xlabel('x'); ylabel('f(x)', 'rotation', 1);

% Построение графика интегральной функции распределения

subplot(2,1,2);

stairs(x,sort(Fx),'LineWidth',0.8,'MarkerFaceColor','c');

grid on;

%xlim([a,b]);

ylim([0,1]);

title('График интегральной функции распределения F(x)','fontsize',15);

xlabel('x'); ylabel('F(x)', 'rotation', 1);

%Вычисление матожидания

M = sum(r) / n;

%Вычисление дисперсии

D = sum(((r - M).^2)/n);

T=table(['M';'D'],[M;D],[M-Mteor;D-Dteor],[Mteor;Dteor]);

T.Properties.VariableNames = {'Eval' 'RNUNI' 'Fault' 'Teor\_value'};

clear i;

function R=RNUNI(a,b)

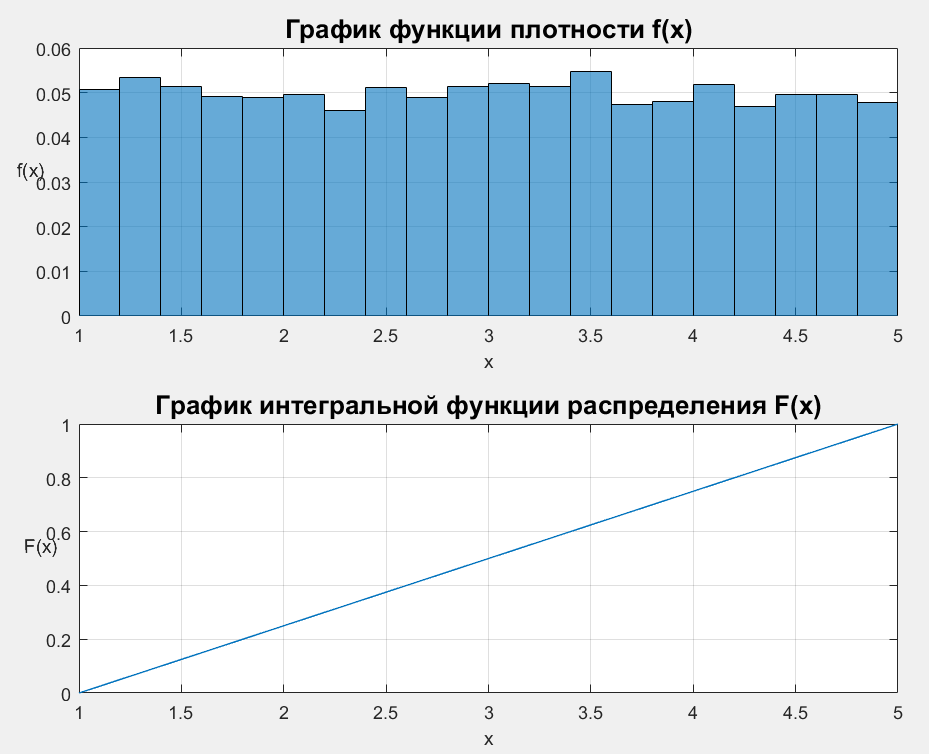
R = a+(b-a)\*rand;

end

### *Результаты работы программы:*

Для a=1, b=5

|  |  |  |  |
| --- | --- | --- | --- |
| Момент | RNUNI | Погрешность | Теоретическое значение |
| M | 3,0129 | 0,0129 | 3 |
| D | 1,3194 | -0,0138 | 1,33 |



## НОРМАЛЬНОЕ РАСПРЕДЕЛЕНИЕ

### *Код программы*

clc

clearvars;

syms r;

Mteor=0;

Dteor=1;

n=10000;

r1 = zeros(1,n);

r2 = zeros(1,n);

for i=1:n

r1(i) = RNNRM1();

r2(i) = RNNRM2();

end

M1 = mean(r1);

S1 = std(r1);

x1=unique(r1);

fx1 = zeros(1,length(x1));

Fx1 = zeros(1,length(x1));

for i=1:length(x1)

fx1(i) = exp((-(x1(i)-M1)^2)/(2\*S1^2))/(S1\*sqrt(2\*pi));

Fx1(i) = (1+erf((x1(i)-M1)/(sqrt(2\*S1^2))))/2;

end

M2 = mean(r2);

S2 = std(r2);

x2 = unique(r2);

fx2 = zeros(1,length(x2));

Fx2 = zeros(1,length(x2));

for i=1:length(x2)

fx2(i) = exp((-(x2(i)-M2)^2)/(2\*S2^2))/(S2\*sqrt(2\*pi));

Fx2(i) = (1+erf((x2(i)-M2)/(sqrt(2\*S2^2))))/2;

end

set(gcf, 'Position', [300, 100, 800, 600]);

%Построение графика функции плотности

subplot(2,2,1);

histogram(r1,'Normalization','probability','NumBins',20);

grid on;

xlim([min(r1),max(r1)]); %ylim([0,1]);

title('График функции плотности f(x)','fontsize',10);

xlabel('x'); ylabel('f(x)', 'rotation', 1);

subplot(2,2,2);

histogram(r2,'Normalization','probability','NumBins',20);

grid on;

xlim([min(r2),max(r2)]); %ylim([0,1]);

title('График функции плотности f(x)','fontsize',10);

xlabel('x'); ylabel('f(x)', 'rotation', 1);

% Построение графика интегральной функции распределения

subplot(2,2,3);

stairs(x1,Fx1,'LineWidth',0.8,'MarkerFaceColor','c');

grid on;

xlim([min(r1),max(r1)]); ylim([0,1]);

title('График интегральной функции распределения F(x)','fontsize',7);

xlabel('x'); ylabel('F(x)', 'rotation', 1);

subplot(2,2,4);

stairs(x2,Fx2,'LineWidth',0.8,'MarkerFaceColor','c');

grid on;

xlim([min(r2),max(r2)]); ylim([0,1]);

title('График интегральной функции распределения F(x)','fontsize',7);

xlabel('x'); ylabel('F(x)', 'rotation', 1);

%Вычисление матожидания

M1 = sum(r1) / n;

M2 = sum(r2) / n;

%Вычисление дисперсии

D1 = sum(((r1 - M1).^2)/n);

D2 = sum(((r2 - M2).^2)/n);

T=table(['M';'D'],[M1;D1],[M2;D2],[Mteor;Dteor]);

T.Properties.VariableNames = {'Eval' 'RNNRM1' 'RNNRM2' 'Teor\_value'};

clear i S1 S2;

function R=RNNRM1()

R = sqrt(-2\*log(rand))\*cos(2\*pi\*rand);

end

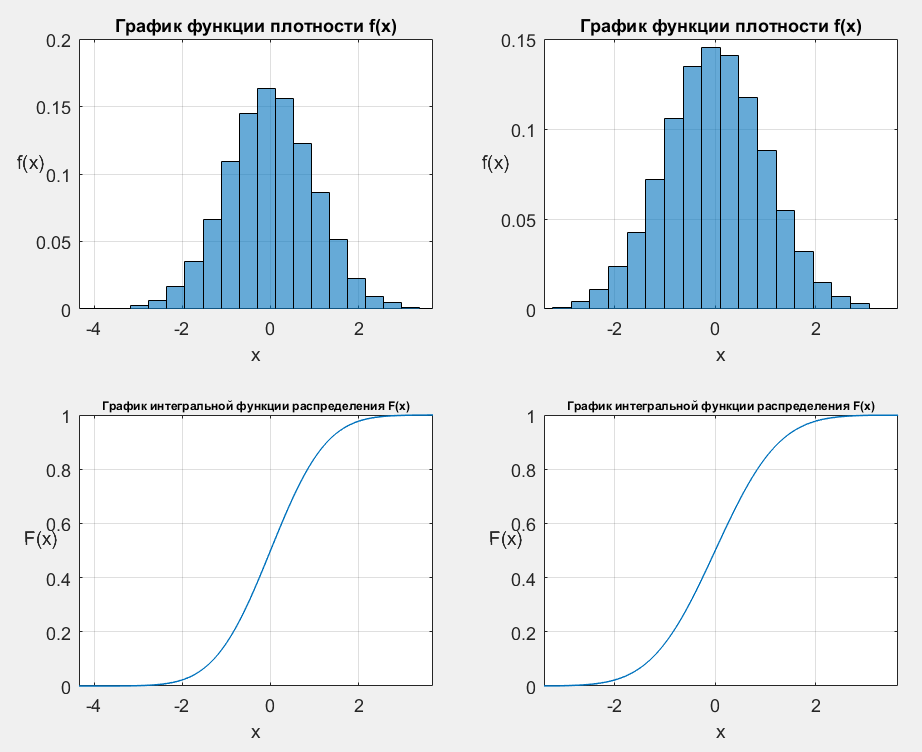
function R=RNNRM2()

R = sum(rand(1,12))-6;

end

|  |  |  |  |
| --- | --- | --- | --- |
|  | RNRM1 | RNRM2 | Теоретическое значение |
| M | 0,0012 | -0,00027 | 0 |
| D | 0,9998 | 0,991460 | 1 |

*RNRM1: RNRM2:*



## ЭКСПОНЕНЦИАЛЬНОЕ РАСПРЕДЕЛЕНИЕ

#### Код программы

clc

clear

syms r;

b=1;

n=10000;

Mteor=b;

Dteor=b^2;

r = zeros(1,n);

for i=1:n

r(i) = RNEXP(b);

end

x = unique(r);

fx = zeros(1,n);

Fx = zeros(1,n);

for i=1:length(x)

if (x(i)<0)

fx(i) = 0;

Fx(i) = 0;

else

fx(i) = b\*exp(-b\*x(i));

Fx(i) = 1-exp(-b\*x(i));

end

end

set(gcf, 'Position', [300, 100, 800, 600]);

%Построение графика функции плотности

subplot(2,1,1);

histogram(r,'Normalization','probability','NumBins',20);

grid on;

xlim([min(r),max(r)]); %ylim([0,1]);

title('График функции плотности f(x)','fontsize',15);

xlabel('x'); ylabel('f(x)', 'rotation', 1);

% Построение графика интегральной функции распределения

subplot(2,1,2);

stairs(x,Fx,'LineWidth',0.8,'MarkerFaceColor','c');

grid on;

xlim([min(r),max(r)]);

ylim([0,1]);

title('График интегральной функции распределения F(x)','fontsize',15);

xlabel('x'); ylabel('F(x)', 'rotation', 1);

%Вычисление матожидания

M = sum(r) / n;

%Вычисление дисперсии

D = sum(((r - M).^2)/n);

T=table(['M';'D'],[M;D],[M-Mteor;D-Dteor],[Mteor;Dteor]);

T.Properties.VariableNames = {'Eval' 'RNEXP' 'Fault' 'Teor\_value'};

clear i;

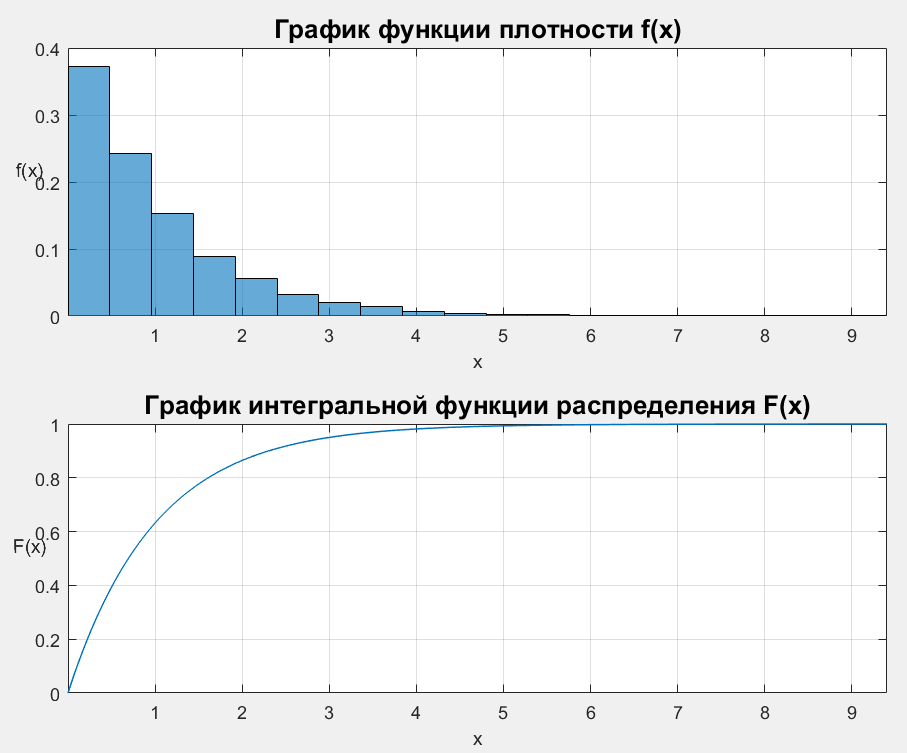
function R=RNEXP(b)

R = -b\*log(rand);

end

### *Результаты работы программы*

|  |  |  |
| --- | --- | --- |
| Момент | RNEXP | Теоретическое значение |
| M | 0,9966 | 1 |
| D | 0,9821 | 1 |



## ХИ-КВАДРАТ РАСПРЕДЕЛЕНИЕ

### *Код программы*

clc

clear

syms r;

N=2;

n=10000;

Mteor=N;

Dteor=2\*N;

r = zeros(1,n);

for i=1:n

r(i) = RNCHIS(N);

end

x = unique(r);

fx = zeros(1,n);

Fx = zeros(1,n);

for i=1:length(x)

if (x(i)>0)

fx(i) = (((x(i)/2)^((N/2)-1))\*(exp(-x(i)/2)))/(2\*gamma(N/2));

Fx(i) = gammainc(N/2,x(i)/2,'upper')/gammainc(N/2,0);

else

fx(i) = 0;

Fx(i) = 0;

end

end

set(gcf, 'Position', [300, 100, 800, 600]);

%Построение графика функции плотности

subplot(2,1,1);

histogram(r,'Normalization','probability','NumBins',20);

grid on;

xlim([min(r),max(r)]); %ylim([0,1]);

title('График функции плотности f(x)','fontsize',15);

xlabel('x'); ylabel('f(x)', 'rotation', 1);

% Построение графика интегральной функции распределения

subplot(2,1,2);

stairs(x,Fx,'LineWidth',0.8,'MarkerFaceColor','c');

grid on;

xlim([min(r),max(r)]);

ylim([0,1]);

title('График интегральной функции распределения F(x)','fontsize',15);

xlabel('x'); ylabel('F(x)', 'rotation', 1);

%Вычисление матожидания

M = sum(r) / n;

%Вычисление дисперсии

D = sum(((r - M).^2)/n);

T=table(['M';'D'],[M;D],[M-Mteor;D-Dteor],[Mteor;Dteor]);

T.Properties.VariableNames = {'Eval' 'RNCHIS' 'Fault' 'Teor\_value'};

clear i;

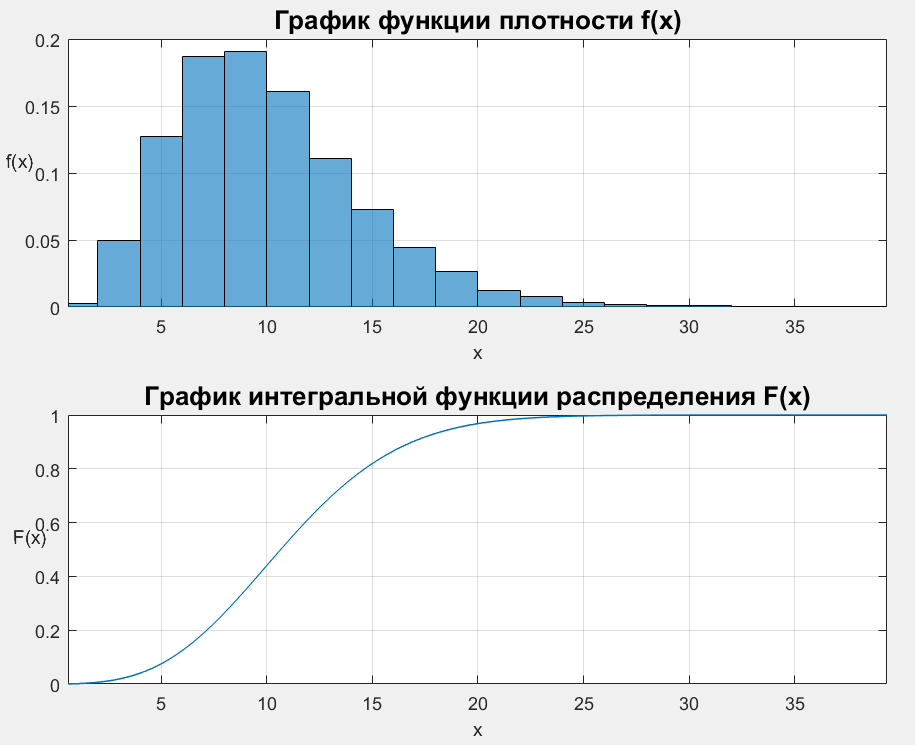
function R=RNCHIS(N)

R = sum((normrnd(0,1,1,N)).^2);

end

#### Результаты работы программы

|  |  |  |
| --- | --- | --- |
| Момент | RNCHIS | Теоретическое значение |
| M | 9,9931 | 10 |
| D | 19,573 | 20 |



## РАСПРЕДЕЛЕНИЕ СТЬЮДЕНТА

## *Код программы*

clc

clear

syms r;

N=10;

n=10000;

Mteor=0;

Dteor=N/(N-2);

r = zeros(1,n);

for i=1:n

r(i) = RNSTUD(N);

end

x = unique(r);

fx = zeros(1,n);

Fx = zeros(1,n);

for i=1:length(x)

fx(i) = (gamma((N+1)/2)\*((1+x(i)^2/N)^(-(N+1)/2)))/(sqrt(N\*pi)\*gamma(N/2));

Fx(i) = tcdf(x(i),N);

end

set(gcf, 'Position', [300, 100, 800, 600]);

%Построение графика функции плотности

subplot(2,1,1);

histogram(r,'Normalization','probability','NumBins',20);

grid on;

xlim([min(r),max(r)]); %ylim([0,1]);

title('График функции плотности f(x)','fontsize',15);

xlabel('x'); ylabel('f(x)', 'rotation', 1);

% Построение графика интегральной функции распределения

subplot(2,1,2);

stairs(x,Fx,'LineWidth',0.8,'MarkerFaceColor','c');

grid on;

xlim([min(r),max(r)]);

ylim([0,1]);

title('График интегральной функции распределения F(x)','fontsize',15);

xlabel('x'); ylabel('F(x)', 'rotation', 1);

%Вычисление матожидания

M = sum(r) / n;

%Вычисление дисперсии

D = sum(((r - M).^2)/n);

T=table(['M';'D'],[M;D],[M-Mteor;D-Dteor],[Mteor;Dteor]);

T.Properties.VariableNames = {'Eval' 'RNSTUD' 'Fault' 'Teor\_value'};

clear i;

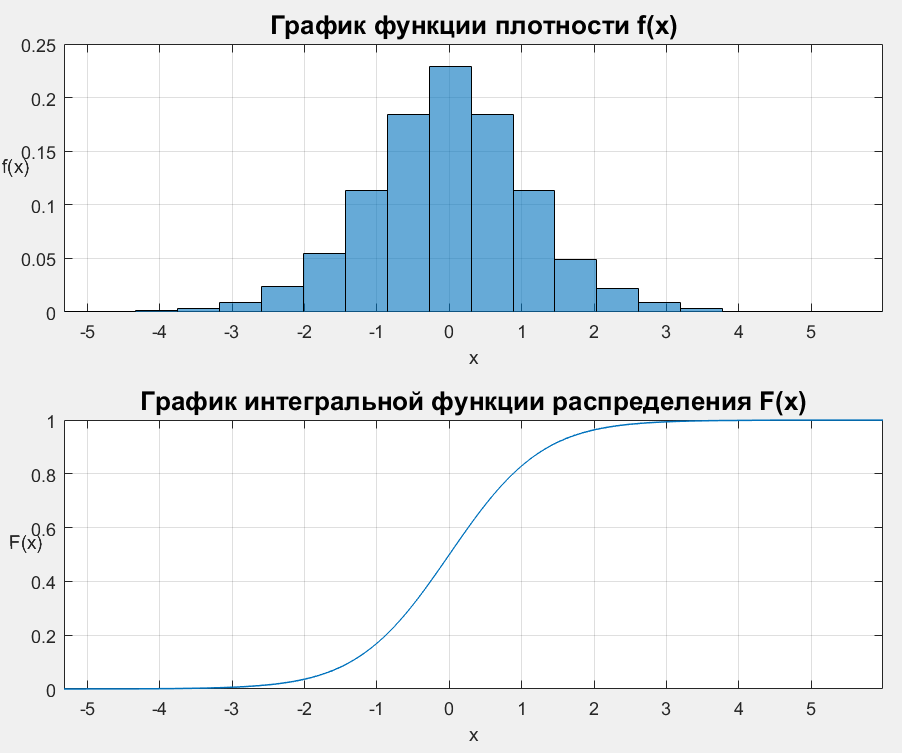
function R=RNSTUD(N)

R = normrnd(0,1,1)/sqrt(RNCHIS(N)/N);

end

### *Результаты работы программы*

|  |  |  |  |
| --- | --- | --- | --- |
| Момент | RNSTUD | Погрешность | Теоретическое значение |
| M | 0,00839 | 0,00839 | 0 |
| D | 1,25795 | 0,00795 | 1.25 |



## Вывод

В ходе выполнения лабораторной работы были разработаны программные датчики случайных величин. Исследование характеристик моделируемых датчиков показало, что точность моделирования высока, так как вычисленные значения математического ожидания и дисперсии совпали с теоретическими значениями лишь с небольшой погрешностью. К каждой программе представлены графики функции плотности распределения и интегральной функции распределения.