МИНИСТЕРСТВО ОБРАЗОВАНИЯ И НАУКИ РОССИЙСКОЙ ФЕДЕРАЦИИ

ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ АВТОНОМНОЕ

ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ

«САМАРСКИЙ НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ

УНИВЕРСИТЕТ ИМЕНИ АКАДЕМИКА С.П. КОРОЛЕВА»

(САМАРСКИЙ УНИВЕРСИТЕТ)

Отчёт по лабораторной работе №1  
по курсу «Моделирование информационно-вычислительных систем»

Выполнил:

Алеев И.И.

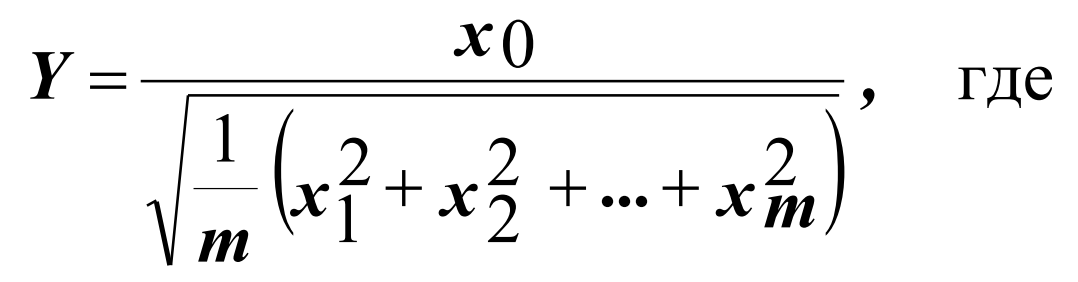
студент гр. №6304-090301D  
Проверила:

Симонова Е.В.

# Задание на моделирование

Получить N = 1000 реализаций случайной величины, имеющей t—распределение Стьюдента с **m** степенями свободы. Моделирование выполнить

специальным методом с учетом того, что t-распределению Стьюдента с m степенями свободы подчиняется распределение случайной величины:

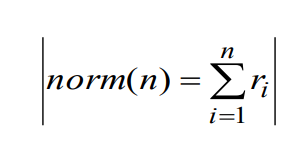


x0, x1, …, xm – взаимно независимые реализации случайной величины, имеющей нормированное нормальное распределение с параметрами (0,1).

Параметр m выбрать самостоятельно. Статистический контроль провести с использованием15 интервалов разбиения, a = 0.05

# Описание метода моделирования случайной величины

Для получения реализаций случайной величины Х с нормированным нормальным распределением с параметрами (0,1) использовался метод генерации равномерной случайной величины в диапазоне от -1, до 1. Используется свойство, что множество реализаций случайной величины с равномерным распределением. Для получения на ЭВМ реализаций нормальной случайной величины используют центральную предельную теорему теории вероятностей, из которой, в частности, следует, что сумма реализаций ri случайной величины R,равномерно распределенной в интервале [0, 1), при достаточно большом значении n, может рассматриваться как случайная величина, описываемая нормальным законом распределения вероятностей. Это обстоятельство позволяет легко алгоритмизировать процесс формирования реализации нормальной случайной величины.



Обычно при n = 12 распределение величины norm уже весьма близко к нормальному. Нетрудно убедиться, что величина, формируемая с помощью приведенного алгоритма, в предположении независимости отдельных величин ri , имеет дисперсию (n/12) и среднее, равное (n/2). Сформировать из этой величины любую другую, распределенную нормально, но с другими значениями параметров (среднее – m, дисперсия – σ^2) можно с помощью алгоритма



# Листинг программы моделирования

using System;

using System.Windows.Forms;

namespace MIVS\_LAB

{

public partial class Form1 : Form

{

public Form1()

{

InitializeComponent();

f = new TaskForm();

f.Hide();

}

TaskForm f;

Random rgen = new Random((int)DateTime.Now.Second);

double h = 0;

const int n = 15;

const float alpha = 0.05f;

const float y0 = 23.7f;

int N = 1000;

int m = 2;

private double AvgDist(double a, double b)

{

return a + rgen.NextDouble() \* (b - a);

}

private double NormDist(float micro, float D)

{

double S = 0;

double w = Math.Sqrt(D);

for (int i = 0; i < m\*m; i++)

{

S += AvgDist(micro -w, micro + w) - 6;

}

return (S - micro \* m\*m) / Math.Sqrt(m\*m\*D);

}

private double StudentDistribution(int m)

{

float[] x = new float[m];

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

{

x[i] = (float)NormDist(0, 1);

}

double sq\_sum = 0;

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

{

sq\_sum = x[i] \* x[i];

}

return x[0] \* Math.Sqrt(m) / (Math.Sqrt(sq\_sum / m));

}

private float[] grid(float from, float to, int count)

{

float[] ret = new float[count];

h = Math.Abs(to - from) / count;

ret[0] = from;

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

{

ret[i] = ret[i - 1] + (float)h;

}

return ret;

}

private void button1\_Click(object sender, EventArgs e)

{

int.TryParse(maskedTextBox1.Text, out m);

float[] a = new float[N];

var cnt = new int[n];

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

{

a[i] = (float)StudentDistribution(m);

//a[i] = (float)NormDist(0, 1);

}

float max\_A = a[0];

float min\_A = a[0];

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

{

if (a[i] > max\_A)

max\_A = a[i];

if (a[i] < min\_A)

min\_A = a[i];

}

var gr = grid(min\_A, max\_A, n);

System.Array.Sort(a);

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

{

if (a[i] < gr[0])

cnt[0]++;

else if (a[i] >= gr[n - 1])

cnt[n - 1]++;

else

{

int gr\_i = 1;

while (a[i] >= gr[gr\_i] || a[i] < gr[gr\_i - 1])

gr\_i++;

cnt[gr\_i]++;

}

}

chart1.Series[0].Points.Clear();

chart2.Series[0].Points.Clear();

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

{

chart1.Series[0].Points.AddXY(gr[i], 1f\*cnt[i] / N );

}

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

{

chart2.Series[0].Points.AddXY(a[i], i \* 1f / N);

}

double hi2\_sum = 0;

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

{

hi2\_sum += (cnt[i]\* cnt[i]) \* n / N;

}

double hi2 = n \* hi2\_sum / N;

if (hi2 < y0)

label2.Text = ("Гипотеза принята (alpha = 0.05, [n-1] = 14, hi^2 = " + hi2 + ").");

else

label2.Text = ("hi ^ 2 = " + hi2 + ").");

chart1.Invalidate();

chart2.Invalidate();

}

private void button2\_Click(object sender, EventArgs e)

{

f.Show();

}

}

}

# Результат моделирования