

ЛАБОРАТОРНАЯ РАБОТА № 6

ИССЛЕДОВАНИЕ КАЧЕСТВА ГЕНЕРАТОРОВ СЛУЧАЙНЫХ ЧИСЕЛ

Цель работы. изучить и практически освоить оценки качества генераторов случайных чисел (ГСЧ) в различных системах программирования по заданным теоретическим показателям, с помощью критериев согласия и с помощью нормированной автокорреляционной функции на предмет независимости случайных чисел.

Задачи: проанализировать анализ качества ГСЧ статистическими методами: по критерию отклонения математического ожидания, дисперсии, среднего квадратического отклонения;

ТЕОРЕТИЧЕСКАЯ ЧАСТЬ

В практике моделирования и, особенно в практике вероятностных имитационных моделей приходится использовать случайные последовательности или просто случайные числа. При моделировании систем на ЭВМ программная имитация случайных воздействий любой сложности сводится к генерированию некоторых стандартных (базовых) процессов и к их последующему функциональному преобразованию. Получение случайных чисел с требуемым законом распределения обычно выполняется в два этапа

1. Формирование физическим или программным методом случайного числа U_i , равномерно распределенного на $[0; 1]$, $i = 1, 2, \dots$.
2. Программный переход от U_i к случайному числу X_i , имеющему требуемое распределение $F_X(x)$ [1].

В связи с этим особое значение приобретают случайные числа, **равномерно** распределенные в интервале $[0; 1]$. Например, генерирование экспоненциально распределенных случайных чисел t_i может быть выполнено по формуле

$$t_i = -\frac{1}{\lambda} \ln(R_i),$$

где λ – параметр экспоненциального закона, R_i – равномерно распределенное случайное число из интервала $(0; 1]$.

Существуют физические и программные датчики (генераторы) случайных чисел. Программные датчики случайных чисел фактически генерируют псевдослучайные числа. Согласно Лемеру, последовательность псевдослучайных чисел можно считать случайной, если «каждый ее член непредсказуем для непосвященного, и она удовлетворяет ряду традиционных статистических тестов».

Полученные с помощью программных методов случайные последовательности в идеале должны состоять из

- равномерно распределенных,
- статистически независимых,
- воспроизводимых,
- неповторяющихся чисел.

Исследование качества генераторов случайных чисел (ГСЧ) по критерию отклонения математического ожидания, дисперсии среднего квадратического отклонения

Известно, что при равномерном законе распределения случайной непрерывной величины в интервале $[0; 1]$ соответствующие математическое ожидание (m), дисперсия (s^2) и среднеквадратичное отклонение (s) имеют следующие теоретические значения: $m = 0.5$; $s^2 = 1/12$; $s = 0.28867$ ($s = \sqrt{1/12}$).

Критерий заключается в сравнении теоретических параметров равномерного распределения с реальными значениями, полученными для конечной выборки.

1.1. Пример. Анализ качества ГСЧ системы MATLAB

В системе MATLAB всех версий используется функция **rand**, реализующая равномерно распределенные числа в интервале $[0; 1]$.

Среднее значение массива чисел определяется функцией **mean**, дисперсия – функцией **var**, среднее квадратическое отклонение (стандартное отклонение) – функцией **std**.

Программный код анализа случайных чисел:

```
clear, clc
%% Генерирование выборки 500 случайных чисел
x = rand(500, 1);
%% Вычисление среднего значения выборки
m1 = mean(x)
%% Вычисление дисперсии данной выборки
s2 = var(x)
%% Вычисление среднего квадратического отклонения
s = std(x)

%% Расчет относительных погрешностей в процентах
%% по математическому ожиданию
m = 0.5;
Dm = abs((mean(x) - m)/m)*100;
fprintf('\n Относительная погрешность по математическому
ожиданию: %g%%\n', Dm);
%% по дисперсии
d = 1/12;
Dd = abs((var(x) - d)/d)*100;
fprintf(' Относительная погрешность по дисперсии: %g%%\n', Dd);
%% по среднему квадратическому отклонению
sd = sqrt(d);
Ds = abs((std(x) - sqrt(1/12))/sqrt(1/12))*100;

fprintf(' Относительная погрешность по стандартному отклонению:
%g%%\n', Ds);

%% Генерирование дополнительной выборки
y = rand(500, 1);

%% Диаграмма оценки равномерности случайных чисел
fig1 = figure(1);
```

```

set(fig1, 'name', 'Случайные числа функции rand')
plot(x,y,'o', 'markersize', 4);
str = '\bf\fontsize{11}\fontname{times}Проверка на равномерность
случайных чисел';
title(str)
xlabel('\bf\fontsize{11}\fontname{times} Random numbers')
ylabel('\bf\fontsize{11}\fontname{times} Random numbers')

```

В программе выполняется построение диаграммы визуальной оценки равномерности случайных чисел. Числа – кружочки на диаграмме должны равномерно заполнить квадрат со стороной, равной единице. На рис. 1 приведен пример проверки случайной последовательности на равномерность распределения в интервале $[0; 1]$.

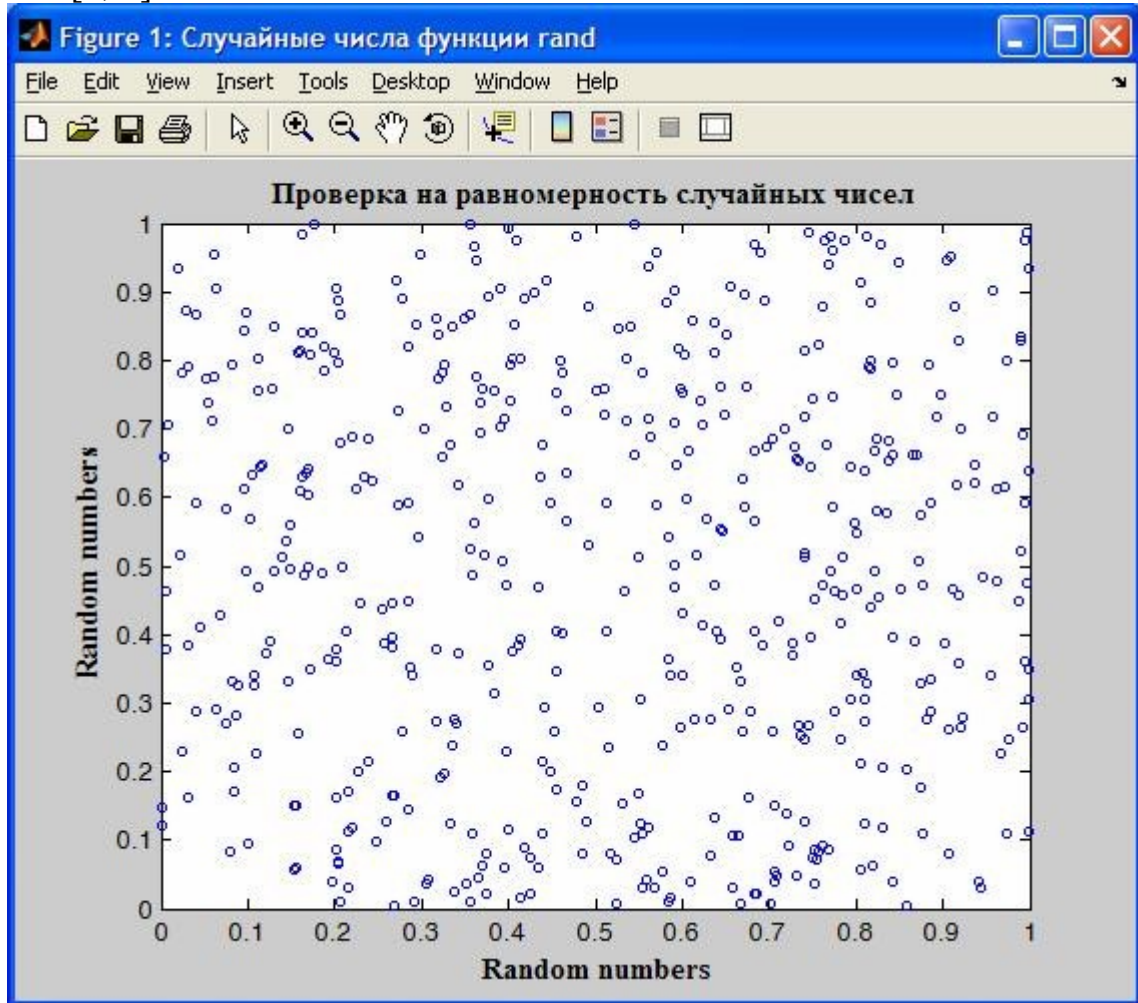


Рисунок 1 Проверка равномерности случайных чисел для функции rand

Задание 1

1. Провести в зависимости от номера Варианта статистическое исследование функции **rand** при различных объемах выборки: малых $n < 25$; средних $n \approx 150$; больших $n > 500$. Результаты испытаний усреднить.

Вариант №1: 11 испытаний (n : 24, 142, 600);

Вариант №2: 12 испытаний (n : 22, 144, 650);

Вариант №3: 13 испытаний (n : 20, 146, 700);

Вариант №4: 14 испытаний (n : 18, 148, 750);

Вариант №5: 15 испытаний (n : 16, 150, 800);

Вариант №6: 16 испытаний (n : 14, 152, 850);

Вариант №7: 17 испытаний (n: 12, 154, 900);
Вариант №8: 18 испытаний (n: 10, 156, 950);
Вариант №9: 19 испытаний (n: 19, 149, 1000).
Вариант №10: 20 испытаний (n: 20, 150, 1010).

2. Построить график изменения относительных погрешностей среднего, дисперсии, стандартного отклонения от числа испытаний.
3. Пункт 1 задания выполнить для выборок, сформированных в системах EXCEL(CALC), PYTHON, C, C#(JAVA). Сформированные выборки импортировать в PYTHON (MATLAB), где произвести необходимый анализ.
4. Построить гистограммы в системе для сформированных выборок.

1.2. Исследование качества ГСЧ, сформированного по методу Фибоначчи

Генератор случайных чисел, использующий метод Фибоначчи, применялся в начале 50-х годов XX века [4]. Рекуррентное соотношение Фибоначчи имеет вид

$$X_{n+1} = (X_n + X_{n-1}) \bmod(M),$$

где X_{n+1} , X_n , X_{n-1} – целые числа, лежащие между нулем и некоторым большим числом M , который называется модулем, n – порядковый номер числа [4].

Для получения случайных чисел R_n из интервала $[0; 1]$ следует вычислить дробь

$$R_n = \frac{X_n}{M}.$$

Программный код формирования случайных чисел по методу Фибоначчи:

```
clear,clc,close all
```

```
N = 500; %% количество генерируемых чисел
```

```
M = 2^30; %% модуль
```

```
%% 1-я последовательность случайных чисел
```

```
X0 = 12345; %% 1-е произвольное число
```

```
X1 = 67890; %% 2-е произвольное число
```

```
for n = 1 : N
```

```
    X = mod( X1 + X0 , M); %% следующее число
```

```
    X0 = X1;
```

```
    X1 = X;
```

```
    Zx(n,1) = X;
```

```
end
```

```
Rx = Zx/M;
```

```
mf = mean(Rx);
```

```
fprintf('\n Среднее выборочное для метода фибоначчи: %g%%\n',  
mf);
```

```
sf2 = var(Rx);
```

```
fprintf(' Выборочная дисперсия для метода фибоначчи: %g%%\n',  
sf2);
```

```

sf = std(Rx);
fprintf(' Выборочное стандартное отклонение: %g%%\n', sf);

%% Расчет относительных погрешностей в процентах
m = 0.5;
fprintf('\n Относительная погрешность по математическому
ожиданию: %g%%\n', abs((mf - m)/m)*100);

d = 1/12;
fprintf(' Относительная погрешность по дисперсии: %g%%\n',
abs((sf2 - d)/d)*100);

%% по среднему квадратическому отклонению
sd = sqrt(d);
fprintf(' Относительная погрешность по стандартному отклонению:
%g%%\n', abs((sf - sd)/sd)*100);

%% 2-я последовательность случайных чисел
Y0 = 333; %% 1-е произвольное число
Y1 = 123; %% 1-е произвольное число

for n = 1 : N

Y = mod( Y1 + Y0, M); %% следующее число
Y0 = Y1;
Y1 = Y;
Zy(n,1) = Y;

end

Ry = Zy/M;

%% Диаграмма оценки равномерности случайных чисел
fig2 = figure(2);
set(fig2, 'name', 'Случайные числа Фибоначчи')
plot(Rx,Ry,'o', 'markersize', 4);
str = '\bf\fontsize{11}\fontname{times}Проверка на
равномерность случайных чисел';
title(str)
xlabel('\bf\fontsize{11}\fontname{times} Random numbers')
ylabel('\bf\fontsize{11}\fontname{times} Random numbers')
Пример выполнения программы:
Среднее выборочное для метода Фибоначчи: 0.49687%
Выборочная дисперсия для метода Фибоначчи: 0.0861675%
Выборочное стандартное отклонение: 0.293543%

Относительная погрешность по математическому ожиданию: 0.626048%
Относительная погрешность по дисперсии: 3.40103%
Относительная погрешность по стандартному отклонению: 1.6863%

```

Задание 2

1. Написать программу на языке программирования формирования простых трехзначных чисел с целью их использования в качестве начальных чисел в методе Фибоначчи. Рассчитать относительные погрешности по математическому ожиданию, дисперсии, стандартному отклонению.
2. Построить гистограммы для сформированных выборок (**Zx** и **Zy**) с разбивкой графического окна.

1.3. Исследование качества ГСЧ, сформированного по методу серединных квадратов

Метод серединных квадратов был предложен Нейманом [2] и заключается в следующем: выбирается число, меньшее 1 разрядностью $2n$. Оно возводится в квадрат. Из полученного результата (разрядность которого должна быть $2 \cdot (2n)$, если нет, то добавляются нули справа от полученного числа) выбираются $2n$ чисел из середины полученного после возведения в квадрат числа. Число записывается после десятичной точки. Далее процедура повторяется.

Для примера выберем 4-х разрядное ($2n = 4$) число $a_0 = 0.1234$. После возведения в квадрат получим число, равное 0.01522756. Из него выбираем четыре серединные цифры, т. е. 5227. Получаем новое случайное (псевдослучайное) число $a_1 = 0.5227$. Описанные действия отобразим в следующем виде:

$$\begin{aligned} a_0 &= 0.1234 \rightarrow a_0^2 = 0.01\mathbf{5227}56; \\ a_1 &= 0.5227 \rightarrow a_1^2 = 0.27\mathbf{3215}29; \\ a_2 &= 0.3215 \rightarrow a_2^2 = 0.10\mathbf{3362}25; \\ a_3 &= 0.3362 \rightarrow a_3^2 = 0.11\mathbf{3030}44; \\ a_4 &= 0.3030 \rightarrow a_4^2 = 0.091809 \rightarrow 0.09\mathbf{1809}00 \\ &\text{и так далее.} \end{aligned}$$

Задание 3

1. Написать программу формирования случайных чисел по методу серединных.
2. Начальное число выбрать (по указанию преподавателя) из следующего списка, приведенного в табл. 1.

Таблица 1

Варианты заданий для метода серединных квадратов	
№1	№1 = 0.1234; №2 = 0.2234; №3 = 0.3234; №4 = 0.4234; №5 = 0.5234; №6 = 0.6234; №7 = 0.7234; №8 = 0.8234; №9 = 0.9234; №10 = 0.9934;
№2	№1 = 0.123456; №2 = 0.223456; №3 = 0.323456; №4 = 0.423456; №5 = 0.523456; №6 = 0.623456; №7 = 0.723456; №8 = 0.823456; №9 = 0.923456; №10 = 0.993456;
№3	№1 = 0.12345678; №2 = 0.22345678; №3 = 0.32345678; №4 = 0.42345678; №5 = 0.52345678; №6 = 0.62345678; №7 = 0.72345678; №8 = 0.82345678; №9 = 0.92345678; №10 = 0.99345678;
№4	№1 = 0.12345678; №2 = 0.22345678; №3 = 0.32345678; №4 = 0.42345678; №5 = 0.52345678; №6 = 0.62345678; №7 = 0.72345678; №8 = 0.82345678; №9 = 0.92345678; №10 = 0.99345678;

3. Провести в зависимости от номера Варианта статистическое исследование ГСЧ (вычисление среднего значения выборки, дисперсии, стандартного отклонения выборки) при различных объемах выборки: малых $n < 25$; средних $n \approx 150$; больших

$n > 500$. Результаты испытаний усреднить и сравнить с аналогичными результатами, проведенными для выборки, сформированной с помощью функции **rand** системы MATLAB;

Вариант №1: (объем выборки: 24, 142, 600);

Вариант №2: (объем выборки: 22, 144, 650);

Вариант №3: (объем выборки: 20, 146, 700);

Вариант №4: (объем выборки: 18, 148, 750);

Вариант №5: (объем выборки: 16, 150, 800);

Вариант №6: (объем выборки: 14, 152, 850);

Вариант №7: (объем выборки: 12, 154, 900);

Вариант №8: (объем выборки: 10, 156, 950);

Вариант №9: (объем выборки: 17, 157, 999);

Вариант №10: (объем выборки: 19, 158, 1010);

4. Построить гистограммы для сформированных выборок случайных чисел по методу серединных квадратов и сравнить с гистограммой для выборок, сформированных с помощью функции **rand** системы MATLAB/PYTHON.

5. Построить гистограммы для сформированных выборок.

6. Построить диаграмму визуального контроля равномерного заполнения квадрата со стороной, равной единице.

1.4. Исследование качества ГСЧ, сформированного по линейному конгруэнтному методу

Формирование случайных (псевдослучайных) чисел по линейному конгруэнтному методу основывается на следующем рекуррентном соотношении:

$$R_{k+1} = (aR_k + c)(\bmod M), \quad k = 0, 1, \dots \quad (1)$$

где R_{k+1} – вновь формируемое число, a – множитель (мультипликативная константа), R_k – предыдущее число (R_0 – назначаемое число), c – приращение (инкремент), \bmod – модуль, бинарная операция для обозначения остатка от деления двух чисел, M – целочисленная константа [2]. Для n -разрядных целых чисел $M = 2^n$. В самом простом случае принимается, что $c = 0$. Массив случайных чисел $\{x_i\}$ из интервала $(0,1)$ будет формироваться следующим образом:

$$\{x_i\} = \{R_i\}/M, \quad (2)$$

где R_i – числа, определяемые по формуле (1).

В стандартной процедуре реализации линейного конгруэнтного метода (1) принимается, что a, c, M – целые положительные числа. Приведем определение конгруэнтности двух чисел X и Y : **два числа Y и X конгруэнтны (сравнимы) по модулю числа M** , если они дают одинаковые остатки при делении на этот модуль M . Таким образом, по формуле (1) число R_{k+1} будет конгруэнтно по модулю M числу $(aR_k + c)$.

При выборе чисел a, c, M придерживаются следующих правил:

1) c, M – должны быть взаимно простыми числами. Причем число M определяет собой период числовой псевдослучайной последовательности: чем больше M , тем длиннее последовательность псевдослучайных чисел;

2) $b = a - 1$ кратно P для любого простого P , являющегося делителем M .

В качестве множителя a рекомендуется принимать первообразный корень по модулю M . Приведем следующее классическое определение:

Первообразный корень по модулю M – натуральное число g такое, что наименьшее положительное число k , для которого разность $g^k - 1$ делится на M (без остатка), совпадает с $\varphi(M)$, где $\varphi(M)$ – число натуральных чисел, меньших M и взаимно простых с M .

Например, при $M=7$ первообразный корень по модулю 7 является число 3. Действительно, $\varphi(M) = 6$, т. е. количество чисел ряда 1, 2, 3, 4, 5, 6, каждое из которых взаимно простое с числом 7.

Два числа называются **взаимно простыми**, если в качестве общего делителя они имеют только единицу).

Числа $3^1 - 1 = 2$, $3^2 - 1 = 8$, $3^3 - 1 = 26$, $3^4 - 1 = 80$, $3^5 - 1 = 242$ не делятся на 7 без остатка, и лишь $3^6 - 1 = 728$ делится на 7 (частное от деления равно 104).

Например, в системе MATLAB формирование простых чисел производится с помощью функции **primes**. Для проверки, являются ли два числа взаимно простыми, можно применить функцию **gcd**, которая определяет наибольший общий делитель для двух чисел.

В самом простом случае принимается, что $c = 0$. При этом можно использовать следующие рекомендации по выбору параметров генератора:

1. Начальное значение R_0 может быть произвольно.

2. Выбор a должен удовлетворять трем требованиям: $a \pmod{8} = 5$,

$M/100 < a < M - \sqrt{M}$, двоичные знаки a не должны иметь очевидного шаблона.

3. В качестве a следует выбирать нечетное число, такое, что

$$a/M \approx 1/2 - \sqrt{3}/6 = 0.21132486540519.$$

Пример формирования модуля M в командном окне MATLAB:

```
>> N = 7*10^6;  
>> m = primes(N);  
>> M = m(end)  
M =  
6999997
```


Задание 4

1. Полагая в формуле (1) $c = 0$, написать в MATLAB(PYTHON) программу формирования случайных чисел, приняв следующие числа N для расчета модуля в зависимости от номера варианта:

№1	№1: $N = 7 \cdot 10^6$; №2: $N = 7.5 \cdot 10^6$; №3: $N = 8 \cdot 10^6$; №4: $N = 8.5 \cdot 10^6$; №5: $N = 9 \cdot 10^6$; №6: $N = 9.5 \cdot 10^6$; №7: $N = 10 \cdot 10^6$; №8: $N = 10.5 \cdot 10^6$; №9: $N = 10.6 \cdot 10^6$;
№2	№1: $N = 7.2 \cdot 10^6$; №2: $N = 7.52 \cdot 10^6$; №3: $N = 8.3 \cdot 10^6$; №4: $N = 8.54 \cdot 10^6$; №5: $N = 9.55 \cdot 10^6$; №6: $N = 9.66 \cdot 10^6$; №7: $N = 10.7 \cdot 10^6$; №8: $N = 10.8 \cdot 10^6$; №9: $N = 10.9 \cdot 10^6$;
№3	№1: $N = 7.11 \cdot 10^6$; №2: $N = 7.22 \cdot 10^6$; №3: $N = 8.33 \cdot 10^6$; №4: $N = 8.44 \cdot 10^6$; №5: $N = 9.55 \cdot 10^6$; №6: $N = 9.66 \cdot 10^6$; №7: $N = 10.77 \cdot 10^6$; №8: $N = 10.88 \cdot 10^6$; №9: $N = 10.99 \cdot 10^6$;
№4	№1: $N = 5.11 \cdot 10^6$; №2: $N = 5.22 \cdot 10^6$; №3: $N = 6.33 \cdot 10^6$; №4: $N = 5.44 \cdot 10^6$; №5: $N = 6.55 \cdot 10^6$; №6: $N = 6.66 \cdot 10^6$; №7: $N = 6.77 \cdot 10^6$; №8: $N = 6.88 \cdot 10^6$; №9: $N = 6.99 \cdot 10^6$;

2. В качестве первого назначаемого случайного числа R_0 (в зависимости от номера варианта) принять следующие значения:

№1	№1: $m(11)$, №2: $m(12)$, №3: $m(13)$, №4: $m(14)$, №5: $m(15)$, №6: $m(16)$, №7: $m(17)$, №8: $m(18)$, №9: $m(18)$, где m – массив простых чисел, сформированный с помощью выражения $m = \text{primes}(N)$;
№2	№1: $m(21)$, №2: $m(22)$, №3: $m(23)$, №4: $m(24)$, №5: $m(25)$, №6: $m(26)$, №7: $m(27)$, №8: $m(28)$, №9: $m(29)$, где m – массив простых чисел, сформированный с помощью выражения $m = \text{primes}(N)$;
№3	№1: $m(31)$, №2: $m(32)$, №3: $m(33)$, №4: $m(34)$, №5: $m(35)$, №6: $m(36)$, №7: $m(37)$, №8: $m(38)$, №9: $m(38)$, где m – массив простых чисел, сформированный с помощью выражения $m = \text{primes}(N)$;
№4	№1: $m(41)$, №2: $m(42)$, №3: $m(43)$, №4: $m(44)$, №5: $m(45)$, №6: $m(46)$, №7: $m(47)$, №8: $m(48)$, №9: $m(49)$, где m – массив простых чисел, сформированный с помощью выражения $m = \text{primes}(N)$;

3. Вычислить период формируемой случайной последовательности;
4. Произвести статистический анализ созданного ГСЧ по линейному конгруэнтному методу;
5. Построить гистограммы полученных распределений случайных чисел.
6. Построить функции плотности и распределения для сформированных выборок случайных чисел. Совместить диаграммы с теоретическими функциями.

Статистическое тестирование выборки псевдослучайных чисел по критерию Колмогорова – Смирнова

По критерию Колмогорова – Смирнова (КС–критерию) осуществляется проверка простой статистической гипотезы H_0 (нулевой гипотезы) о том, что функция распределения $F(x)$ случайной величины X совпадает с некоторой

известной функцией $F_0(x)$ при некотором уровне значимости α . КС-критерием можно пользоваться уже при объеме выборки $n \geq 20$.

Например, в системе MATLAB КС-критерий реализован в функции **kstest**. Рассмотрим пример использования функции **kstest** для проверки гипотезы о том, что функция распределения (F) выборки, сформированной с помощью функции **rand**, соответствует функции распределения (F0) экспоненциального закона с параметром 1 той же самой выборки.

Программное решение примера в командном окне MATLAB:

```
>> x = rand(25,1); F0 = expcdf(x,1);  
>> H = kstest(x,[x,F0])  
H =  
1
```

Полученный результат **H = 1** означает, что нулевая гипотеза отвергается, т. е. выборочная функция равномерного распределения (**F**) в интервале $[0; 1]$ имеет значительные расхождения с предполагаемой функцией экспоненциального распределения (**F0**) с уровнем значимости $\alpha = 0.05$ (по умолчанию). Если закладывается другой уровень значимости, отличный от 0.05, то тогда он должен быть введен в функцию **kstest**. На том же примере это будет выглядеть так (с уровнем значимости 0.012):

```
>> x = rand(25,1); F0 = expcdf(x,1);  
>> H = kstest(x,[x,F0],0.012)  
H =  
1
```

По-прежнему нулевая гипотеза отвергается.

Рассмотрим пример использования функции **kstest** для проверки гипотезы о том, что функция распределения выборки, сформированной с помощью **rand**, соответствует функции распределения равномерного закона из интервала $[0; 1]$ той же самой выборки.

Решение примера в командном окне MATLAB:

```
>> x = rand(25,1); F0 = unifcdf(x,0,1);  
>> H = kstest(x,[x,F0])  
H =  
0
```

Нулевая гипотеза о равномерном распределении выборки принимается.

Задание 5

1. По критерию Колмогорова – Смирнова протестировать выборки случайных чисел, сформированных по методу срединных квадратов.
2. По критерию Колмогорова – Смирнова протестировать выборки случайных чисел объема 100, 500, 1000, сформированных по линейному конгруэнтному методу.

Исследование качества ГСЧ по критерию независимости случайных чисел с помощью нормированной автокорреляционной функции

Корреляционная функция называется автокорреляционной, если производится статистический анализ одного случайного процесса (или одной выборки случайных чисел).

Нормированной корреляционной функцией называется отношение центрированной корреляционной функции к дисперсии случайного процесса [3].

Для нахождения значений нормированной корреляционной функции необходимо предварительно центрировать значения случайного процесса или значения отсчетов случайной величины, т.е. выполнить операцию

$$n_1 - m,$$

где n_1 – случайное число, $m = 0.5$ – теоретическое математическое ожидание равномерного распределения.

Для определения корреляционной функции по результатам опыта выбирается достаточно большой объем выборки, чтобы можно было в широком диапазоне формировать разницу между двумя соседними значениями случайных чисел. Эту разницу для непрерывного времени обычно обозначают через τ и тогда корреляционная функция обозначается, как $R(\tau)$. Если объем выборки составляет N , то диапазон вычисления корреляционной функции будет определяться как $N - \tau$. Величина τ задает область определения корреляционной функции. Например, τ может меняться от 0 до 6 – 8. При этом N должно быть много больше 6 или 8. Область суммирования принимает значения от 1 (первое случайное число выборки) до $N - \tau$.

После этого корреляционная функция вычисляется по следующей экспериментальной формуле:

$$R(\tau) = \frac{1}{N - \tau} \sum_{j=1}^{N-\tau} n_j n_{j+\tau}, \quad (3)$$

где n_j – случайное число из заданной выборки случайных чисел.

Расчет по приведенной формуле: если взято какое-либо случайное число, то другое случайное число отстоит от первого на величину τ .

Обозначим нормированную корреляционную функцию как \tilde{R} . Центрированную корреляционную функцию обозначим через R^o . Тогда нормированная корреляционная функция будет определяться в виде отношения

$$\tilde{R} = \frac{R^o}{S},$$

где S – дисперсия данной выборки случайных чисел.

Вычисление R^o можно выполнять по приведенной экспериментальной формуле (3), если в ней применяются центрированные случайные числа.

ГСЧ считается хорошим, если при τ , не равным нулю, модуль нормированной корреляционной функции меньше 0.1, т. е. $|\tilde{R}| < 0.1$.

Приведем пример программного анализа независимости последовательности случайных чисел, формируемых функцией **rand**, с помощью автокорреляционной функции.

Программный код решения примера:

```
clear,clc
%% Ввод параметров в интерактивном режиме
V1 = inputdlg({'Введите число больше  
10.....'},...
    'Сдвиг больше 1','Корреляционная функция',1,{'800','6'});

%% Преобразование к числам с плавающей точкой
V2 = str2num(char(V1));
% Гарантированное выделение целой части
V = fix(V2(1));
z = fix(V2(2));
% Формирование выборки случайных чисел
N = rand(V,1);
%% Центрирование выборки случайных чисел относительно
математического ожидания
Nc = N - 0.5;

%% Расчет автокорреляционной функции
sum1 = Nc(1:(V-z));
sum2 = Nc((1+z):V);
Rc = sum(sum1.*sum2)/(V-z);
s = var(N);
Rn = (Rc/s);
%% Проверка качества случайных чисел
if abs(Rn) < 0.1
    fprintf('\n\t ГСЧ высокого качества\n')
else
    fprintf('\n\t ГСЧ низкого качества\n')
end
%% Интерактивное сообщение
helpdlg('Смотрите результаты в командном окне','Корреляционная
функция')
```

В программе по умолчанию исследуется объем выборки величиной 800 со сдвигом, равным 6 между числами.

Задание 6

1. Произвести расчет нормированной корреляционной функции для интервального сдвига **z** в пределах от 0 до 50.
2. Построить график нормированной автокорреляционной функции, т. е. зависимость **Rn** от **z**.
3. Произвести расчет нормированной корреляционной функции для объема выборки в соответствии с номером Варианта:

Вариант №1: N = 410;	Вариант №2: N = 520;
Вариант №3: N = 630;	Вариант №4: N = 740;
Вариант №5: N = 850;	Вариант №6: N = 960;
Вариант №7: N = 1070;	Вариант №8: N = 1180;

Вариант №9: N = 1190; Вариант №10: N = 1210.

4. Выполнить первые три пункта задания для анализа ГСЧ в EXCEL(CALC).
5. Выполнить первые три пункта задания для анализа ГСЧ в C#(JAVA).
6. Выполнить первые три пункта задания для анализа ГСЧ в PYTHON.
7. Выполнить первые три пункта задания для анализа ГСЧ в C.
8. Сделать заключение о системе программирования, в которой ГСЧ является наиболее качественным.

ПРИЛОЖЕНИЕ

Схема вариантов лабораторной работе 6

N	Выполняемые задания	N	Выполняемые задания
1	1.9, 3.1(1)8, 5.1	11	4.2(1-3), 5.2, 6.5
2	2, 4.4(1-3), 5.2	12	2, 4.3(1-3), 5.2
3	3.2(2)5, 5.1, 6.3	13	3.2(6).1, 5.1, 6.1
4	2, 3.3(3)2, 5.1	14	2, 4.3(4-6), 5.2
5	4.4(4-6), 5.2, 6.2	15	1.2, 3.3(7)7, 5.1
6	1.6, 4.4(7-9), 5.2	16	2, 4.1(7-9), 5.2
7	4.2(7-9), 5.2, 6.4	17	4.1(4-6), 5.2, 6.6
8	1.3, 3.4(4)4, 5.1	18	1.10, 3.4(8).6, 5.1
9	2, 4.2(4-6), 5.2	19	1.5, 3.1(9).9, 5.1
10	1.4, 3.1(5).3, 5.1	20	3.2(10)10, 5.1, 6.7