

Лабораторная работа №1

МОДЕЛИРОВАНИЕ БАЗОВОЙ СЛУЧАЙНОЙ ВЕЛИЧИНЫ

Цель работы: построить датчик базовой случайной величины по заданному алгоритму и выполнить тестирование датчика на соответствие основным свойствам базовой случайной величины.

1.1. Описание объекта моделирования

Базовой случайной величиной (БСВ) в моделировании называют непрерывную случайную величину z , равномерно распределенную на интервале $(0 \leq z \leq 1)$. Её плотность распределения вероятностей задаётся формулой:

$$f(z) = 1, \quad (0 \leq z \leq 1). \quad (1.1)$$

Математическое ожидание M и дисперсия D базовой случайной величины имеют следующие значения: $M(z) = 0.5$ и $D(z) = 1/12$.

БСВ моделируются с помощью программных датчиков. Датчик БСВ – это программа, выдающая по запросу одно случайное значение $z \in \{0 \leq z \leq 1\}$. Путём многократного обращения к датчику БСВ получают выборку независимых случайных значений $z_1, z_2, z_3, \dots, z_n$.

Программный датчик БСВ обычно вычисляет значения $z_1, z_2, z_3, \dots$ по какой-либо рекуррентной формуле вида

$$z_{i+1} = f(z_i) \quad (1.2)$$

при заданном стартовом значении z_0 .

Например, мультипликативный конгруэнтный датчик БСВ задаётся двумя параметрами: модулем m и множителем k . Обычно эти параметры представляют собой достаточно большие целые числа. При заданных m, k псевдослучайные числа z_i вычисляются по рекуррентной формуле:

$$\begin{aligned} A_i &= (k \cdot A_{i-1}) \bmod m_i, \quad i = 1, 2, \dots \\ z_i &= A_i / m, \end{aligned} \quad (1.3)$$

где m – модуль,

k – множитель,

A_0 – начальное значение,

$\bmod$ – операция вычисления остатка от деления произведения (kA_{i-1}) на m .

Датчик (1.3) даёт периодическую псевдослучайную последовательность $\bar{z}_{i=\overline{1,T}}$ с длиной периода $T \leq m-1$. Чтобы длина периода T была максимальной, модуль m берут близким к максимальному представимому в компьютере целому числу. Для упрощения операций деления и вычисления остатков в двоичных

ЭВМ часто берут $m = 2^n$. Рекомендуется также брать достаточно большой множитель k , причем взаимно простой с m .

Рекомендаций, гарантирующих высокое качество датчиков, не существует. Поэтому необходимо проводить их специальное тестирование.

С помощью статистических тестов проверяют два свойства датчика БСВ, делающих его точной моделью идеальной БСВ:

во-первых, проверяют *равномерность распределения* чисел, выдаваемых датчиком на интервале $[0, 1]$, и,

во-вторых, их *статистическую независимость*.

Обозначим равномерное распределение вероятностей на интервале $[0, 1]$ как $R[0,1]$ и утверждение, что БСВ z имеет распределение $R[0,1]$, запишем в виде $z \sim R[0,1]$.

Проверку $z \sim R[0,1]$ можно выполнить с помощью частотного теста. Последовательность проверки следующая:

1. Интервал $[0,1]$ разбить на K равных отрезков, например, $K=10$.
2. Подсчитать, сколько чисел z_i попало в каждый из k отрезков, то есть число попадания $n_1, \dots, n_k$.
3. Найти относительные частоты попаданий в отрезки: $p_1 = n_1/T, \dots, p_k = n_k/T$.
4. Построить гистограмму частот $p_1, \dots, p_k$ на K отрезках интервала $[0,1]$.

Простейшую проверку статистической независимости БСВ можно осуществить, оценивая линейную корреляцию между числами z_i и z_{i+s} , отстоящими друг от друга в псевдослучайной последовательности на фиксированный шаг $s \geq 1$. Тогда во всей выборке $z_1, z_2, \dots, z_n$ имеем следующие $(n-s)$ реализаций пар: $(z_1, z_{1+s}), (z_2, z_{2+s}), \dots, (z_{n-s}, z_n)$.

По этим реализациям можно рассчитать оценку $\hat{R}$ коэффициента корреляции для значений БСВ по формуле

$$\hat{R} = 12 \frac{1}{T-s} \left(\sum_{i=1}^{T-s} z_i z_{i+s} \right) - 3. \quad (1.4)$$

1.2. Порядок выполнения работы

1. Построить датчик БСВ с периодом $T > 500$.
2. Оценить математическое ожидание и дисперсию псевдослучайных значений z_i и сравнить их с теоретическими значениями M и D .
3. Проверить датчик БСВ на равномерность и построить гистограмму распределения относительных частот $p_1, \dots, p_k$ на K отрезках интервала $[0,1]$.
4. Проверить датчик БСВ на независимость, определяя коэффициент корреляции для разных значений s и T . Построить в одном графическом окне графики зависимости $\hat{R} = f(T)$ для $s=2, s=5, s=10$.
5. Оформить отчет о проделанной работе.

1.3. Содержание отчета

1. Цель, задание и последовательность выполнения работы.
2. Результаты сравнений математического ожидания и дисперсии псевдослучайных значений z_i с теоретическими значениями M и D .
3. Гистограмма распределения относительных частот попаданий псевдослучайных величин в отрезки интервала $[0,1]$.
4. Графики зависимости коэффициента корреляции для $s=2, s=5, s=10$.
5. Выводы о результатах моделирования БСВ.

1.4 Варианты заданий

№ пп	Задание
1	Линейный конгруэнтный метод: $X_{k+1} = (aX_k + c) \bmod m,$ где $a > 0, c > 0, m > 0$ – некоторые целочисленные константы, выбор которых определяет качество датчика. Рекомендуются следующие значения: $a = 1664525, c = 1013904223, m = 2^{32}$. Получаемая последовательность БСВ зависит от выбора стартового числа X_0 и при разных его значениях получаются различные последовательности случайных чисел. Длина периода равна m тогда и только тогда, когда: 1. НОД (c, m) = 1 (то есть c и m взаимно просты); 2. $(a - 1)$ кратно p для всех простых p – делителей m; 3. $(a - 1)$ кратно 4, если m кратно 4. При реализации следует выбирать $m = 2^e$, где e – число бит в машинном слове. Рекомендуется использовать только старшие разряды. Для этих констант выписаны условия, гарантирующие удовлетворительное качество получаемых случайных чисел.
2	Генератор «little-frog»: $A_0 = 1, A_i = A_{i-1}M \bmod (2^r);$ $z_i = A_i 2^{-r},$ где r – число двоичных разрядов, используемых для представления числа в компьютере; M – множитель, достаточно большое число, взаимно-простое с 2^r; z_i – БСВ. Длина периода T последовательности $\{z_i\}$ равна 2^{r-2}. Последовательность A_i предварительно разбивается на подпоследовательности длины m, начинающиеся с чисел $A_{km}, k = 0, 1, 2, \dots$ (то есть с начальных значений последовательностей) и каждая подпоследовательность используется для построения соответствующих случайных чисел $A_{(k+1) \cdot m} = A_{km} \cdot M^m \bmod (2^r)$. Для моделирования k-й подпоследовательности $z_{km} = A_{km} \cdot 2^{-r}$. Множитель M для генератора, обеспечивающего "прыжки" с шагом длиной m, вычисляется на основе равенства $M \equiv M^m \bmod (2^r)$.
3	Датчик Фибоначчи $X_{i+1} = (X_i + X_{i-k}) \bmod (2^m);$ $z_{i+1} = \frac{X_{i+1}}{2^m},$

	где m - чётное число; k – целое число, $k \geq 16$. Начальные состояния $X_0, \dots, X_{16}$ создаются Random на интервале $[0, 2^m]$
4	Фибоначчи с запаздываниями: $X_i = (X_{i-24} + X_{i-55}) \bmod (2^m);$ $z_i = \frac{X_i}{2^m},$ где $i \geq 55$; m – чётное число; $X_0, X_1, \dots, X_{54}$ – произвольные целые числа, полученные Random на интервале $[0, 2^m]$. Числа 24 и 55 выбраны так, чтобы определялась последовательность с длиной периода 2^{55-1}. Числа 24 и 55 называют запаздыванием, а числа X_i – последовательностью Фибоначчи с запаздыванием.
5	Аддитивный генератор Fish: $A_i = (A_{i-55} + A_{i-24}) \bmod (2^{32})$ $B_i = (B_{i-52} + B_{i-19}) \bmod (2^{32})$ $z_i = \frac{A_i}{2^{32}}, z_{i+1} = \frac{B_i}{2^{32}}$ состоит из двух аддитивных генераторов A_i и B_i, начальные состояния которых создаются Random. Эти последовательности прореживаются попарно в зависимости от случая: если значение младшего значащего бита B_i равно 1, то пара используется, если 0 – игнорируется.
6	Алгоритм Pike: $A_i = (A_{i-55} + A_{i-24}) \bmod (2^{32})$ $B_i = (B_{i-57} + B_{i-7}) \bmod (2^{32})$ $C_i = (C_{i-58} + C_{i-19}) \bmod (2^{32})$ $z_i = \frac{A_i}{2^{32}}, z_{i+1} = \frac{B_i}{2^{32}}, z_{i+2} = \frac{C_i}{2^{32}}$ состоит из трех аддитивных генераторов A_i, B_i и C_i. Начальные состояния этих генераторов создаются Random. При генерации сравниваются биты переноса при сложении. Если все три одинаковы (все нули или все единицы), то тактируются все три генератора. Если нет, то тактируются только два совпадающих генератора.
7	Алгоритм Mush $A_i = (A_{i-55} + A_{i-24}) \bmod (2^{32})$ $B_i = (B_{i-52} + B_{i-19}) \bmod (2^{32})$ $z_i = \frac{A_i}{2^{32}}, z_{i+1} = \frac{B_i}{2^{32}}$ Взаимно прореживающий генератор: если бит переноса A установлен, тактируется B и при переполнении устанавливаем бит переноса; если бит переноса B установлен, то тактируется A и при переполнении устанавливается бит переноса. Окончательным выходом является XOR выходов A и B.
8	Мультипликативно-конгруэнтный датчик

	$A_i = (kA_{i-1}) \bmod m, \quad i = 1, 2, \dots,$ $z_i = \frac{A_i}{m},$ где m – модуль; k – множитель; A_0 – начальное значение псевдослучайной последовательности A. Рекомендуются следующие шесть вариантов множителя k для датчика с модулем $m = 2^{31} - 1$: 1078318381, 1203248318, 397204094, 2027812808, 1323257245, 764261123. Во всех этих случаях длина периода $T = m - 1$. Поэтому в качестве стартового значения A_0 можно брать любое целое число от 1 до $m - 1$.
9	Линейная смешанная формула $A_i = (k_1 A_{i-1} + \dots + k_r A_{i-r} + C) \bmod m, \quad i = 1, 2, \dots,$ $z_i = \frac{A_i}{m},$ где r – порядок рекурсии $A_0, A_1, \dots, A_{r+1}$ – стартовые значения, $k_1, \dots, k_r$ – множители. Длина периода $T < m^r$.
10	Квадратичный конгруэнтный метод $A_i = (k_2 A_{i-1}^2 + k_1 A_{i-2} + C) \bmod m, \quad i = 1, 2, \dots,$ $z_i = \frac{A_i}{m},$ A_0, A_1 – стартовые значения, k_1, k_2 – множители. Длина периода $T < m-1$.
11	Метод произведений. Алгоритм построения датчика методом произведений сводится к следующему: 1. Два следующих друг за другом случайных числа A_i и A_{i+1} умножаются, пусть произведение этих чисел записано, как $A = a_1 a_2 a_3 a_4,$ где a_i – значение i-го разряда числа A. 2. Из A выделяют средние разряды, которые представляют собой следующее случайное число: $\begin{array}{c} a_1 a_2 a_3 a_4 \\ \downarrow \\ A_{i+2} \end{array}$ Очевидно, что количество разрядов чисел псевдослучайной последовательности должно быть четным, а под произведение выделяется количество разрядов в два раза больше, чем у перемножаемых чисел. Если произведение получилось меньше по числу разрядов, чем выделено, то слева добавляются нули. Стартовыми значениями являются числа A_0 и A_1, которые подбираются для получения необходимого периода T.
12	Метод Макларена - Марсальи Обозначим через x_i, y_i ($i = 0, 1, 2, \dots$) две псевдослучайные последовательности, порождаемые независимыми датчиками Д1 и Д2 соответственно; $V = \{V(0), \dots, V(K-1)\}$ – вспомогательная таблица целых чисел. Вначале таблица V заполняется первыми K членами последовательности x_i: $V(j) = x_i, \quad j = 0, 1, \dots, K-1.$ Методом Макларена - Марсальи число z_i определяется путем выполнения следующей

	последовательности операций: $j = \lfloor y_i K \rfloor,$ $z_i = V(j),$ $V(j) = x_{K+i}, i = 0, 1, 2, \dots,$ где операция $\lfloor y_i K \rfloor$ означает вычисление целой части от $y_i K$. Д2 используется здесь для случайного выбора элемента из таблицы V, а Д1 заменяет этот элемент новым случайным значением. Длина периода равна $T_1 T_1^{K+1}$, где T_1, T_2 – длина периодов у датчиков Д1 и Д2 соответственно. Например, при $T_1 = T_2 = 2^{31} - 2$ и $K = 9$ можно получить $T \approx 2,2 \cdot 10^{105}$. Датчик Д1 – Random, датчик Д2 – любой другой.
13	Метод середины квадратов Алгоритм построения датчика методом середины квадратов сводится к следующему: 1. Число A_i возводится в квадрат, пусть результат записан, как $A_i = a_1 a_2 a_3 a_4,$ где a_j – значение j-го разряда числа A_i. 2. Из A_i выделяют средние разряды, которые представляют собой следующее случайное число: $a_1 a_2 a_3 a_4$ $\downarrow$ A_{i+1} Количество разрядов числа A_i должно быть четным, а под произведение выделяется количество разрядов в два раза больше, чем у A_i. Если результат возведения числа A_i в квадрат получился меньше по числу разрядов, чем выделено, то слева добавляются нули. Стартовым значениям является число A_0, которое подбирается для получения необходимого периода T.
14	Генератор «big-frog» $A_0 = 1, A_i = A_{i-1} M \bmod (2^r);$ $z_i = A_i 2^{-r},$ где r – число двоичных разрядов, используемых для представления числа в компьютере; M – множитель, достаточно большое число, взаимно-простое с 2^r; z_i – БСВ. Длина периода T последовательности $\{z_i\}$ равна 2^{r-2}. Последовательность A_i предварительно разбивается на подпоследовательности длиной u, начинающиеся с чисел $A_{ku}, k = 0, 1, 2, \dots$ и каждая подпоследовательность используется для построения соответствующих случайных чисел $A_{(k+1)u} = A_{ku} \cdot M^u \bmod (2^r)$. Для моделирования k-й подпоследовательности $z_{ku} = A_{ku} \cdot 2^{-r}, u = mN$, где N – количество подпоследовательностей Множитель M для генератора, обеспечивающего "прыжки" с шагом длиной u, вычисляется на основе равенства $M \equiv M^u \bmod (2^r)$.