

### 3. РЕШЕНИЕ ЗАДАЧ МОДЕЛИРОВАНИЯ НА ОСНОВЕ АЛГОРИТМОВ ИМИТАЦИИ СЛУЧАЙНЫХ ВЕЛИЧИН

#### 3.1. Имитация дискретных случайных величин

Под случайной величиной будем понимать величину, точное значение которой не может быть точно известно заранее.

**Дискретная случайная величина**  $X$  - это величина, которая может принимать некоторые фиксированные значения  $x_1, x_2, \dots, x_n$  с вероятностями  $P_1, P_2, \dots, P_n$ , где  $P_1 + P_2 + \dots + P_n = 1$ . Количество возможных значений дискретной случайной величины (число  $n$ ) может быть как конечным, так и бесконечным.

Если количество возможных значений дискретной случайной величины невелико, то ее имитация выполняется так же, как имитация полной группы событий (см. подраздел 2.1). Так, в примере 2.2 прибыль от ремонта одного изделия представляет собой дискретную случайную величину, которая может принимать значения 35 (с вероятностью 0,2), 60 (с вероятностью 0,4), 40 (с вероятностью 0,15) или 25 (с вероятностью 0,25).

Для имитации дискретных случайных величин, имеющих большое (или бесконечное) количество возможных значений, требуется разрабатывать специальные алгоритмы. Такие алгоритмы разрабатываются на основе методов теории вероятностей. Ниже приводятся алгоритмы имитации некоторых дискретных случайных величин, часто применяемых в практических задачах.

##### **Имитация дискретной равномерной случайной величины**

Дискретная равномерная случайная величина – это случайная величина, принимающая любые целочисленные значения на отрезке  $[A; B]$  (где  $A$  и  $B$  – целые числа) с равной вероятностью.

Примечание. Необходимо различать *дискретную равномерную случайную величину*, принимающую *только целочисленные значения* в некотором заданном диапазоне, и *непрерывную равномерную случайную величину*, принимающую *любые значения* в заданном диапазоне. Как правило, если говорится о “равномерной случайной величине”, то имеется в виду *непрерывная величина*.

Для имитации дискретной равномерной случайной величины применяется следующая формула:

$$X = [A + (B - A + 1) \cdot R], \quad (3.1)$$

где  $R$  – СРРЧ;

$[ ]$  – выделение целой части числа.

**Пример 3.1.** Предприятие выпускает книжные полки по заказам. Для крупных заказов предусмотрена скидка: за изготовление одной полки предприятие берет с заказчика 25 ден.ед., если заказ составляет от одной до десяти полок, и 20 ден.ед. – если заказ превышает 10 полок. Затраты предприятия на выпуск одной полки составляют 12 ден.ед. Из опыта работы предприятия известно, что заказы обычно составляют от одной до 40 полок (заказы на изготовле-

ние свыше 40 полок крайне редки). Предлагается отменить скидку для крупных заказов, т.е. брать с заказчика 25 ден.ед. за каждую полку, независимо от размера заказа. Однако в этом случае можно ожидать, что количество крупных заказов уменьшится, и заказы будут составлять, как правило, от одной до 20 полок. Требуется определить, выгодна ли для предприятия отмена скидки.

Для решения этой задачи разработаем имитационную модель на основе метода Монте-Карло. Используя такую модель, определим среднюю прибыль предприятия от выполнения одного заказа.

Приведем алгоритм имитации работы предприятия при использовании скидки.

1. Разыгрывается размер заказа  $X$  (количество заказанных полок) по формуле (3.1) при  $A=1$ ,  $B=40$ .

2. Определяется прибыль предприятия от выполнения заказа. Если заказано не более десяти полок ( $X \leq 10$ ), то прибыль предприятия от выпуска одной полки будет составлять  $25-12=13$  ден.ед., а прибыль от выполнения заказа в целом –  $13X$ . Если заказано более десяти полок ( $X > 10$ ), то прибыль предприятия от выпуска одной полки будет составлять  $20-12=8$  ден.ед., а прибыль от выполнения заказа в целом –  $8X$ .

Шаги 1-2 повторяются многократно, например, 100 000 раз. Вычисляется средняя прибыль предприятия от выполнения одного заказа (как отношение суммарной прибыли к количеству заказов).

Рассмотрим два испытания алгоритма.

*Первое испытание.* Пусть при использовании формулы (3.1) разыграно СРРЧ  $R=0,0795$ . Тогда  $X=[1+(40-1+1) \cdot 0,0795]=4$ , т.е. смоделирован заказ на выпуск четырех полок. В этом случае предприятие берет с заказчика 25 ден.ед. за каждую полку. Затраты предприятия на изготовление каждой полки – 12 ден.ед. Таким образом, прибыль составит  $(25-12) \cdot 4=52$  ден.ед.

*Второе испытание.* Пусть разыграно СРРЧ  $R=0,378$ . Тогда  $X=[1+(40-1+1) \cdot 0,378]=16$ , т.е. смоделирован заказ на выпуск 16 полок. В этом случае предприятие берет с заказчика 20 ден.ед. за каждую полку. Прибыль составит  $(20-12) \cdot 16 = 128$  ден.ед.

Приведем программу на языке VBA, реализующую алгоритм имитации работы предприятия.

```
Sub primer31_1()  
'Имитация работы с использованием скидок  
n = 100000  
min_zakaz = 1: max_zakaz = 40 : zatr = 12: plata1 = 25: plata2 = 20  
For i = 1 To n  
r = Rnd  
x = Int(min_zakaz + (max_zakaz + 1 - min_zakaz) * Rnd)  
If x <= 10 Then prib = (plata1 - zatr) * x Else prib = (plata2 - zatr) * x  
sumprib = sumprib + prib  
Next i  
sredprib=sumprib/n
```

```
MsgBox ("Средняя прибыль от выполнения заказа: " + Str(Round(sredprib,2)))
End Sub
```

По результатам имитации средняя прибыль от выполнения одного заказа составит около 170,5 ден.ед.

Выполним имитацию работы предприятия без использования скидки. Для имитации размера заказа будет использоваться формула (3.1) при  $A=1$ ,  $B=20$ . Прибыль предприятия от выпуска одной полки будет всегда составлять  $25-12=13$  ден.ед., а прибыль от выполнения заказа -  $13X$ , где  $X$  – размер заказа. Программа для имитации работы предприятия без использования скидки имеет следующий вид.

```
Sub primer31_2()
'Имитация работы без скидок
n = 100000
min_zakaz = 1: max_zakaz = 20 : zatr = 12: plata = 25
For i = 1 To n
r = Rnd
x = Int(min_zakaz + (max_zakaz + 1 - min_zakaz) * Rnd)
prib = (plata - zatr) * x
sumprib = sumprib + prib
Next i
sredprib=sumprib/n
MsgBox ("Средняя прибыль от выполнения заказа: " + Str(Round(sredprib,2)))
End Sub
```

По результатам имитации, прибыль от выполнения одного заказа составит в этом случае около 136,55 ден.ед.

Таким образом, отмену скидки следует признать невыгодой, так как она приводит к снижению прибыли из-за сокращения размеров заказов.

### **Имитация пуассоновской случайной величины**

Пуассоновская случайная величина – это величина, принимающая значения  $m=0, 1, 2, \dots$  с вероятностями, определяемыми по следующей формуле:

$$P_m = \frac{\lambda^m}{m!} e^{-\lambda}, \quad (3.2)$$

где  $\lambda$  – параметр распределения (т.е. некоторое число).

Таким образом, пуассоновская случайная величина может принимать любые неотрицательные целочисленные значения (от нуля до бесконечности).

Например, пуассоновскому закону распределения соответствует количество событий в пуассоновском потоке [1, 16], происходящих за некоторый интервал времени  $t$ . В этом случае  $\lambda=\Lambda t$ , где  $\Lambda$  – интенсивность потока событий (т.е. среднее количество событий, происходящих в единицу времени).

Для имитации пуассоновской случайной величины применяется следующий алгоритм.

1. Значение моделируемой величины принимается равным нулю:  $m=0$ .

2. Вычисляется значение вспомогательной переменной  $P_0$ :  $P_0 = e^{-\lambda}$ .
3. Вспомогательная переменная  $Q$  принимается равной единице:  $Q=1$ .
4. Разыгрывается СРРЧ  $R$ .
5. Вычисляется новое значение вспомогательной переменной  $Q$ :  $Q=Q \cdot R$ .
6. Если  $Q \geq P_0$ , то значение моделируемой величины увеличивается на единицу ( $m=m+1$ ), и выполняется возврат к шагу 4. Если  $Q < P_0$ , то алгоритм завершается. Смоделированное значение пуассоновской случайной величины равно  $m$ .

Приведем программную реализацию рассмотренного алгоритма в виде подпрограммы на языке VBA.

```
Sub puasson(l, m)
p0=exp(-l)
q=1 : m=0
povtor:
r=Rnd
q=q*r
If q>=p0 Then
m=m+1
Goto povtor
End If
End Sub
```

Здесь переменная  $l$  – параметр пуассоновского распределения. Эта величина должна задаваться при вызове подпрограммы.

Применение рассмотренного алгоритма имитации пуассоновской случайной величины показано в примере 3.9.

### 3.2. Имитация непрерывных случайных величин

Под **непрерывной случайной величиной** будем понимать случайную величину, которая может принимать любые значения в некотором диапазоне  $(a; b)$ . Этот диапазон может быть как ограниченным, так и неограниченным.

Для описания непрерывных случайных величин обычно используются две функции: функция распределения  $F(x)$  и плотность распределения (или плотность вероятности)  $f(x)$ . Смысл этих функций показан в прил. 3.

Принцип работы всех методов имитации непрерывных случайных величин следующий. Над СРРЧ  $R$  (т.е. случайными величинами, распределенными по равномерному закону в диапазоне от 0 до 1) выполняются определенные математические преобразования, позволяющие получить другие случайные величины с заданным законом распределения, т.е. с заданной функцией распределения  $F(x)$  или плотностью распределения  $f(x)$ .

### 3.2.1. Имитация непрерывных случайных величин на основе метода обратных функций

Метод обратных функций применяется, если известна функция распределения имитируемой непрерывной случайной величины  $F(x)$ , и можно найти обратную к ней функцию  $F^{-1}(R)$  (смысл аргумента  $R$  будет показан в описании метода).

Метод обратных функций основывается на следующем утверждении, доказываемом методами теории вероятностей. Пусть  $F(x)$  - функция распределения некоторой непрерывной случайной величины, а  $R$  - СРРЧ. Тогда, если записать уравнение  $F(X)=R$  и выразить из него  $X$ , то  $X$  будет представлять собой случайную величину, имеющую функцию распределения  $F(x)$ .

Построение алгоритма (т.е. вывод формул) для имитации непрерывных случайных величин методом обратных функций выполняется в следующем порядке.

1. Находится функция распределения случайной величины  $F(x)$ .
2. Записывается уравнение  $F(X)=R$ .
3. Из уравнения  $F(X)=R$  находится выражение для  $X$ :  $X = F^{-1}(R)$ . Оно представляет собой алгоритм для имитации непрерывной случайной величины  $X$ .

Примечание. В некоторых случаях функция  $F^{-1}(R)$  оказывается неоднозначной (например, если в выражении  $F(X)$  имеется квадрат). В таких случаях требуется проверить все возможные варианты  $F^{-1}(R)$  и выбрать тот из них, при котором имитируемая случайная величина  $X$  принимает значения в заданном диапазоне  $(a; b)$ .

Примечание. Применение метода обратных функций в некоторых случаях оказывается сложным, так как не для всех непрерывных случайных величин можно аналитически (т.е. в виде формулы) записать функцию распределения  $F(x)$ ; в других случаях, даже если функция  $F(x)$  известна, оказывается невозможным записать обратную к ней функцию. Пример такого случая рассматривается в п.3.2.3. В таких случаях применяются другие методы имитации непрерывных случайных величин, или для реализации метода обратных функций используются численные методы.

**Пример 3.2.** При изготовлении пластмассовых плит выполняются две операции: внесение в пластмассу упрочняющих добавок и охлаждение плиты в форме. Из наблюдений за производственным процессом известно, что время внесения добавок – случайная величина, плотность распределения которой может быть задана графиком, приведенным на рис.3.1. Время охлаждения плиты в форме – случайная величина, которую можно считать распределенной по экспоненциальному закону; в среднем охлаждение плиты занимает 30 минут.

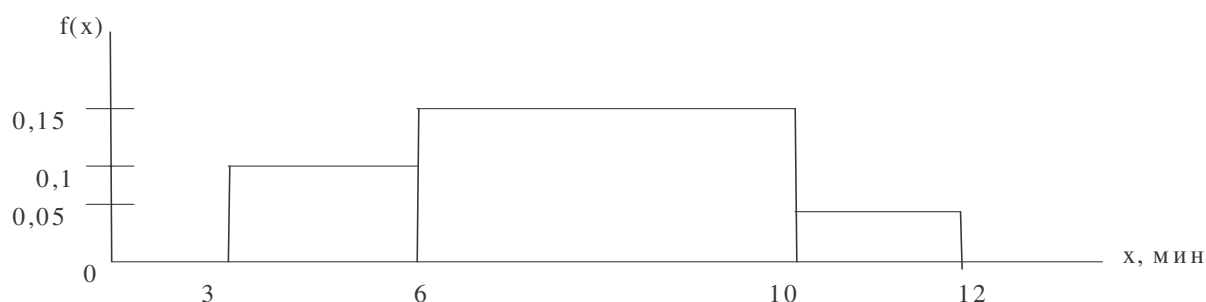


Рис.3.1. Плотность распределения времени внесения добавок (для примера 3.2)

Требуется разработать алгоритмы имитации случайных величин, используемых в задаче, и программную реализацию этих алгоритмов.

Примечание. Применение разработанных алгоритмов имитации будет показано в примере 3.6.

В данной задаче используются две случайные величины: время внесения добавок и время охлаждения плиты.

Рассмотрим разработку *алгоритма имитации времени внесения добавок* на основе метода обратных функций.

По графику, приведенному на рис.3.1, запишем уравнение плотности распределения времени внесения добавок:

$$f(x) = \begin{cases} 0, & x < 3; \\ 0,1, & 3 \leq x < 6; \\ 0,15, & 6 \leq x < 10; \\ 0,05, & 10 \leq x < 12; \\ 0, & x \geq 12. \end{cases} \quad (3.3)$$

Используя формулу (П3.3), находим выражение для функции распределения:

- на интервале  $x < 3$ :  $\int_{-\infty}^x 0 dx = 0$ ;
- на интервале  $3 \leq x < 6$ :  $\int_{-\infty}^3 0 dx + \int_3^x 0,1 dx = 0,1x - 0,3$ ;
- на интервале  $6 \leq x < 10$ :  $\int_{-\infty}^3 0 dx + \int_3^6 0,1 dx + \int_6^x 0,15 dx = 0,15x - 0,6$ ;
- на интервале  $10 \leq x < 12$ :  $\int_{-\infty}^3 0 dx + \int_3^6 0,1 dx + \int_6^{10} 0,15 dx + \int_{10}^x 0,05 dx = 0,05x + 0,4$ ;

- на интервале  $x \geq 12$ :  $\int_{-\infty}^3 0 dx + \int_3^6 0,1 dx + \int_6^{10} 0,15 dx + \int_{10}^{12} 0,05 dx + \int_{12}^x 0 dx = 1$ .

Таким образом, функция распределения времени внесения добавок имеет следующий вид:

$$F(x) = \begin{cases} 0, & x < 3; \\ 0,1x - 0,3, & 3 \leq x < 6; \\ 0,15x - 0,6, & 6 \leq x < 10; \\ 0,05x + 0,4, & 10 \leq x < 12; \\ 1, & x \geq 12. \end{cases} \quad (3.4)$$

Записываем уравнения  $F(X)=R$ :

$$\begin{aligned} 0,1X - 0,3 &= R, & 3 \leq X < 6; \\ 0,15X - 0,6 &= R, & 6 \leq X < 10; \\ 0,05X + 0,4 &= R, & 10 \leq X < 12. \end{aligned}$$

Из уравнений  $F(X)=R$  находим выражение  $X = F^{-1}(R)$ , т.е. формулу (алгоритм) для имитации случайной величины  $X$ :

$$X = \begin{cases} (R + 0,3) / 0,1, & 0 \leq R < 0,3; \\ (R + 0,6) / 0,15, & 0,3 \leq R < 0,9; \\ (R - 0,4) / 0,05, & 0,9 \leq R < 1. \end{cases} \quad (3.5)$$

Границы диапазонов значений  $R$  получены подстановкой соответствующих значений  $X$  в уравнение  $F(X)=R$ . Например, при  $3 \leq X < 6$  уравнение  $F(X)=R$  имеет следующий вид:  $0,1X - 0,3 = R$ . Подставляя в это уравнение значения границ диапазона  $X$  (т.е. 3 и 6), получим границы диапазона  $R$ : 0 и 0,3.

Выполним одно испытание полученного алгоритма. Пусть разыграно значение СРРЧ  $R=0,8197$ . Тогда  $X=(0,8197+0,6)/0,15=9,465$ . Таким образом, в данном испытании смоделировано значение времени внесения добавок, равное 9,465 мин.

Приведем программную реализацию разработанного алгоритма в виде подпрограммы на языке VBA.

```
sub vnesdob(x)
r=rnd
if r<0.3 then x=(r+0.3)/0.1
if (r>=0.3)and(r<0.9) then x=(r+0.6)/0.15
if r>=0.9 then x=(r-0.4)/0.05
end sub
```

Рассмотрим получение алгоритма имитации времени охлаждения плит на основе метода обратных функций. Согласно постановке задачи, можно считать, что эта величина распределена по экспоненциальному закону с математи-



ческим ожиданием (средним значением) 30 мин. Формула плотности распределения для случайной величины, распределенной по экспоненциальному закону, известна из теории вероятностей:

$$f(x) = \begin{cases} 0, & x < 0; \\ \lambda e^{-\lambda x}, & x \geq 0, \end{cases} \quad (3.6)$$

где  $\lambda$  – параметр распределения;  $\lambda = 1/\bar{X}$ , где  $\bar{X}$  – математическое ожидание (среднее значение) экспоненциальной случайной величины.

Таким образом, в данном примере  $\lambda = 1/30$ .

Случайная величина, распределенная по экспоненциальному закону, может принимать (теоретически) любые неотрицательные значения.

По формуле (ПЗ.1) найдем выражение для функции распределения экспоненциальной случайной величины:

- на интервале  $x < 0$ :  $\int_{-\infty}^x 0 dx = 0$ ;
- на интервале  $x \geq 0$ :  $\int_{-\infty}^0 0 dx + \int_0^x \lambda e^{-\lambda x} dx = 1 - e^{-\lambda x}$ .

Таким образом, функция распределения экспоненциальной случайной величины имеет следующий вид:

$$F(x) = \begin{cases} 0, & x < 0; \\ 1 - e^{-\lambda x}, & x \geq 0. \end{cases} \quad (3.7)$$

Записываем уравнение  $F(X) = R$ :

$$1 - e^{-\lambda X} = R.$$

Из уравнения  $F(X) = R$  выражаем  $X$ , т.е. находим формулу (алгоритм) для имитации экспоненциальной случайной величины  $X$ :

$$X = -\frac{1}{\lambda} \ln(1 - R). \quad (3.8)$$

Так как величина  $1 - R$ , как и  $R$ , представляет собой СРРЧ, для имитации экспоненциальной случайной величины можно использовать следующую формулу:

$$X = -\frac{1}{\lambda} \ln R. \quad (3.9)$$

Так как  $\lambda = 1/\bar{X}$ , эту формулу можно записать по-другому:

$$X = -\bar{X} \ln R. \quad (3.10)$$



Для данного примера формула для имитации времени охлаждения плиты будет иметь следующий вид:  $X = -30 \cdot \ln R$ .

Выполним одно испытание полученного алгоритма. Пусть разыграно значение СРРЧ  $R = 0,0795$ . Тогда  $X = -30 \cdot \ln 0,0795 = 75,96$ . Таким образом, в данном испытании смоделировано значение времени охлаждения плиты, равное 75,96 мин.

Приведем программную реализацию алгоритма имитации экспоненциальной случайной величины в виде подпрограммы на языке VBA.

```
sub expon(xsred,x)
r=rnd
x=-xsred*log(r)
end sub
```

Здесь *xsred* – среднее значение имитируемой случайной величины. Эта величина должна задаваться при вызове подпрограммы. Для данного примера эту подпрограмму следует вызывать из основной программы следующим оператором:

```
call expon(30,x)
```

Приведенную подпрограмму можно использовать для имитации любой случайной величины, распределенной по экспоненциальному закону (а не только времени охлаждения плиты в рассматриваемом примере). Для этого достаточно задать значение переменной *xsred*, равное математическому ожиданию имитируемой величины.

Таким образом, алгоритмы для имитации случайных величин, используемых в примере 3.2, получены. Их применение будет показано в примере 3.6.

Приведем (без вывода) алгоритмы имитации некоторых непрерывных случайных величин, часто применяемых в задачах моделирования. Эти алгоритмы также могут быть получены на основе метода обратных функций.

#### **Равномерное распределение:**

$$X = a + (b - a)R, \quad (3.11)$$

где  $a$  и  $b$  – границы диапазона возможных значений равномерной случайной величины.

#### **Распределение Эрланга $k$ -го порядка:**

$$X = -\frac{1}{k\lambda} \sum_{i=1}^k R_i. \quad (3.12)$$

где  $\lambda$  – параметр распределения.

Следует обратить внимание, что для имитации каждого значения случайной величины, распределенной по закону Эрланга  $k$ -го порядка, требуется использовать не одно, а  $k$  СРРЧ.

### Треугольное распределение:

$$X = \begin{cases} a + \sqrt{R(b-a)(c-a)}, & 0 \leq R < (c-a)/(b-a); \\ b - \sqrt{(1-R)(b-a)(b-c)}, & (c-a)/(b-a) \leq R < 1, \end{cases} \quad (3.13)$$

где  $a$  и  $b$  – границы диапазона возможных значений треугольной случайной величины;

$c$  – мода треугольной случайной величины (т.е. точка максимума плотности распределения этой величины).

### 3.2.2. Метод исключений

Этот метод применяется для имитации непрерывных случайных величин, принимающих значения в некотором ограниченном диапазоне  $(a; b)$ . Для применения метода должна быть известна плотность распределения непрерывной случайной величины  $f(x)$ , а также максимальное значение плотности распределения  $f_{\max}$ .

Метод исключений основывается на следующем утверждении (теорема Неймана): если взять два СРРЧ  $R_1$  и  $R_2$  и с их помощью получить случайные числа  $R' = a + (b - a) \cdot R_1$  и  $R'' = f_{\max} \cdot R_2$ , то, если выполняется условие  $R'' \leq f(R')$ , случайное число  $X = R'$  имеет заданную плотность распределения  $f(x)$ .

Алгоритм имитации непрерывной случайной величины с помощью метода исключений реализуется в следующем порядке.

1. Разыгрываются два СРРЧ:  $R_1$  и  $R_2$ .
2. Вычисляются два случайных числа:  $R' = a + (b - a) \cdot R_1$  и  $R'' = f_{\max} \cdot R_2$ .
3. Проверяется условие:  $R'' \leq f(R')$ . Если это условие выполняется, то определяется значение имитируемой случайной величины:  $X = R'$ . Если условие  $R'' \leq f(R')$  не выполняется, то происходит возврат к шагу 1.

**Пример 3.3.** Используя метод исключений, для примера 3.2 разработать алгоритм имитации времени внесения добавок.

Время внесения добавок может составлять от 3 до 12 минут. Максимальное значение плотности распределения известно; оно равно 0,15. Таким образом, для имитации времени внесения добавок можно использовать алгоритм, приведенный выше, при  $a=3$ ,  $b=12$ ,  $f_{\max}=0,15$ .

Приведем два испытания алгоритма.

*Первое испытание.* Разыгрываются два СРРЧ. Пусть разыграны  $R_1=0,0795$ ,  $R_2=0,3780$ . Тогда  $R'=3+(12-3) \cdot 0,0795=3,7155$ ,  $R''=0,15 \cdot 0,3780 = 0,0567$ . Вычислим  $f(R')$ :  $f(3,7155)=0,1$ . Условие  $R'' \leq f(R')$  выполняется. Значит, определяется значение имитируемой случайной величины:  $X=R'=3,7155$ . Таким образом, в данном испытании смоделировано значение времени внесения добавок, равное 3,7155 мин.

*Второе испытание.* Разыгрываются два СРРЧ. Пусть разыграны  $R_1=0,0593$ ,  $R_2=0,7602$ . Тогда  $R'=3+(12-3) \cdot 0,0593=3,5337$ ,  $R''=0,15 \cdot 0,7602=0,11403$ . Вычислим  $f(R')$ :  $f(3,5337)=0,1$ . Условие  $R'' \leq f(R')$  не выполняется. Таким образом, в данном испытании значение имитируемой случайной величины не получено.

Приведем программную реализацию алгоритма имитации времени внесения добавок в виде подпрограммы на языке VBA.

```
Sub iskl()  
nach: r1 = Rnd: r2 = Rnd  
r1s = 3 + 9 * r1 : r2s = 0.15*r2  
If (r1s >= 3) And (r1s <= 6) Then fr = 0.1  
If (r1s > 6) And (r1s <= 10) Then fr = 0.15  
If (r1s > 10) And (r1s <= 12) Then fr = 0.05  
If r2s <= fr Then x = r1s Else GoTo nach
```

Следует обратить внимание, что применение метода исключения возможно не всегда. Например, этим методом нельзя воспользоваться для имитации случайной величины, распределенной по экспоненциальному закону, так как такая величина может принимать любые неотрицательные значения, а метод исключения может применяться только для имитации величин, принимающих значения в некотором *ограниченном* диапазоне.

### 3.2.3. Имитация гауссовского распределения

Случайная величина имеет гауссовское (нормальное) распределение, если ее плотность распределения задается следующим выражением:

$$f(x) = \frac{1}{\sigma\sqrt{2\pi}} \exp\left(-\frac{(x-m)^2}{2\sigma^2}\right), \quad (3.14)$$

где  $m$  - математическое ожидание гауссовской случайной величины, а  $\sigma$  - ее среднее квадратическое отклонение (стандартное отклонение).

Гауссовская случайная величина теоретически может принимать любые значения (от  $-\infty$  до  $+\infty$ ). Однако с вероятностью 0,997 такая величина принимает значения в диапазоне  $(m-3\sigma; m+3\sigma)$ ; это свойство получило название “правила трех сигм”. Поэтому гауссовская случайная величина может использоваться для имитации любой величины, о которой известно, что она обычно принимает значения, близкие к некоторому среднему значению.

Использование метода обратных функций для имитации гауссовской случайной величины затруднено, так как для такой величины невозможно записать формулу функции распределения: интеграл от плотности распределения (3.14) - неберущийся.

Алгоритм имитации гауссовской случайной величины выведен на основе следующего утверждения, доказываемого методами теории вероятностей: сумма  $k$  СРРЧ  $R$  (при достаточно большом  $k$ ) имеет распределение, близкое к гаус-

совскому, с математическим ожиданием  $k/2$  и стандартным отклонением  $\sqrt{k/12}$ . Из этого утверждения может быть выведена следующая формула для имитации гауссовской случайной величины с математическим ожиданием  $m$  и стандартным отклонением  $\sigma$ :

$$X = m + \sigma \sqrt{\frac{12}{k}} \left( \sum_{i=1}^k R_i - \frac{k}{2} \right), \quad (3.15)$$

где  $n$  - количество используемых при имитации СРРЧ.

Чем больше СРРЧ  $R$  суммируется (чем больше  $n$ ), тем выше точность имитации, т.е. тем ближе распределение величины  $X$  к гауссовскому распределению. Для практических задач достаточная точность достигается при  $k=6$ . В этом случае формулу (3.15) можно записать в следующем виде:

$$X = m + \sigma \sqrt{2} \left( \sum_{i=1}^6 R_i - 3 \right). \quad (3.16)$$

Как видно, при использовании формулы (3.16) для имитации *каждого* значения гауссовской случайной величины требуется разыгрывать *шесть* СРРЧ  $R$ .

Приведем программную реализацию алгоритма имитации гауссовской случайной величины по формуле (3.16).

```
Sub gauss(m, sigma, x)
k = 6 : rsum=0
For i = 1 To k
r = Rnd
rsum = rsum + r
Next i
x = m + sigma * Sqr(12 / k) * (rsum - k / 2)
End Sub
```

Здесь переменная  $m$  – математическое ожидание (среднее значение) имитируемой случайной величины,  $\sigma$  – ее стандартное отклонение. Примеры использования этой подпрограммы приведены в подразд.3.3.

Для имитации гауссовской случайной величины может также применяться метод исключения. В качестве диапазона значений случайной величины в этом случае используется диапазон  $(m-3\sigma; m+3\sigma)$ , так как, по правилу “трех сигм”, почти все значения гауссовской случайной величины попадают в этот диапазон. Максимальное значение плотности распределения (3.14) следующее:  $f_{\max} = 1/(\sigma\sqrt{2\pi})$ . Таким образом, для имитации гауссовской случайной величины может использоваться алгоритм, приведенный в п.3.2.2, при  $a=m-3\sigma$ ,  $b=m+3\sigma$ ,  $f_{\max} = 1/(\sigma\sqrt{2\pi})$ .

### 3.2.4. Проверка алгоритмов имитации случайных величин

Проверка алгоритма имитации случайной величины состоит в проверке соответствия величин, получаемых при использовании этого алгоритма, заданному закону распределения моделируемой величины. Если распределение величин, получаемых при использовании алгоритма имитации, близко к заданному закону распределения, то можно считать, что алгоритм имитации правильный.

Рассмотрим два способа проверки алгоритмов имитации случайных величин: с помощью гистограммы и на основе критерия “хи-квадрат”.

#### ***Проверка алгоритмов имитации на основе гистограммы***

Гистограмма распределения представляет собой график (в виде столбиков), отражающий частоту попаданий случайной величины в различные интервалы значений. Гистограмма строится в следующем порядке.

1. Находятся границы диапазона возможных значений случайной величины:  $X_{\min}$ ,  $X_{\max}$ . Если эти границы точно не известны, то они задаются приближенно и могут быть скорректированы по результатам моделирования.

2. Диапазон значений моделируемой величины разбивается на  $k$  интервалов. Количество интервалов  $k$  обычно составляет от 5 до 25. Оно, как правило, подбирается таким образом, чтобы длина интервала  $d=(X_{\max}-X_{\min})/k$ , а также границы интервалов представляли собой “удобные” числа (например, целые).

3. Выполняется большое количество ( $n$ ) испытаний алгоритма имитации. При этом подсчитываются частоты попаданий имитируемой случайной величины в каждый из интервалов:  $m_i$ ,  $i=1, \dots, k$ ,  $m_1+m_2+\dots+m_k=n$ .

4. Вычисляются относительные частоты попаданий имитируемой случайной величины в каждый из интервалов:  $f_i = m_i/(n \cdot d)$ ,  $i=1, \dots, k$ .

Сама гистограмма строится следующим образом: по оси абсцисс откладываются интервалы значений случайной величины (заданные на шаге 2), и на каждом интервале строится столбик (прямоугольник) высотой  $f_i$ .

Если алгоритм имитации случайной величины разработан правильно, то гистограмма должна быть близка к *графику плотности распределения* имитируемой случайной величины.

**Пример 3.4.** Требуется проверить алгоритмы имитации случайных величин, разработанные в примере 3.2, используя гистограммы.

Выполним **проверку алгоритма имитации времени внесения добавок**. Эта случайная величина задана плотностью распределения, график которой приведен на рис.3.1. Время внесения добавок может принимать значения от 3 до 12 минут; таким образом,  $X_{\min}=3$ ,  $X_{\max}=12$ . Разобьем этот диапазон на  $k=9$  интервалов: (3; 4), (4; 5), ..., (11; 12). Длина интервала  $d=1$ .

Приведем программу на языке VBA для получения данных, необходимых для построения гистограммы. В этой программе выполняется большое количество (1000) испытаний алгоритма имитации времени внесения добавок, полу-

ченного в п.3.2.1 (подпрограмма vnesdob). Подсчитываются частоты попаданий имитируемой величины в каждый из интервалов и вычисляются относительные частоты. Границы интервалов заданы в массиве gran. Частоты попаданий в интервалы подсчитываются в массиве m. Относительные частоты вычисляются в массиве f.

```
Sub primer33_1()
Dim gran(1 To 10), f(1 To 10), m(1 To 10)
gran(1) = 3: gran(2) = 4: gran(3) = 5: gran(4) = 6: gran(5) = 7: gran(6) = 8
gran(7) = 9: gran(8) = 10: gran(9) = 11: gran(10) = 12
k = 9 : n = 1000 : d = 1
For i = 1 To n
call vnesdob(x)
For j = 1 To k
If (x > gran(j)) And (x <= gran(j + 1)) Then m(j) = m(j) + 1
Next j
Next i
MsgBox ("Частоты")
For i = 1 To k
MsgBox (Str(gran(i)) + "-" + Str(gran(i + 1)) + ": " + Str(m(i)))
Next i
MsgBox ("Относительные частоты")
For i = 1 To k
f(i) = m(i) / (n * d)
MsgBox (Str(gran(i)) + "-" + Str(gran(i + 1)) + ": " + Str(Round(f(i),3)))
Next i
End Sub

Sub vnesdob(x)
r=Rnd
if r<0.3 then x=(r+0.3)/0.1
if (r>=0.3)and(r<0.9) then x=(r+0.6)/0.15
if r>=0.9 then x=(r-0.4)/0.05
End Sub
```

Результаты выполнения программы приведены в табл.3.1.

Таблица 3.1

Интервал	Частота, $m_i$	Относительная частота, $f_i$	Интервал	Частота, $m_i$	Относительная частота, $f_i$
3 – 4	93	0,093	8 – 9	137	0,137
4 – 5	90	0,09	9 – 10	140	0,140
5 – 6	113	0,113	10 – 11	38	0,038
6 – 7	175	0,175	11 – 12	53	0,053
7 - 8	161	0,161			

На рис.3.2,а приведена гистограмма, построенная по значениям относительных частот. Видно, что гистограмма близка к графику плотности распределения имитируемой величины, приведенному на рис.3.1. Таким образом, можно предполагать, что алгоритм имитации правильный. Более строгая проверка алгоритма (с использованием критерия “хи-квадрат”) будет рассмотрена ниже.



Аналогично выполним **проверку алгоритма имитации времени охлаждения плиты**. Эта случайная величина распределена по экспоненциальному закону с математическим ожиданием 30 мин. Экспоненциальная случайная величина может принимать (теоретически) любые неотрицательные значения. Поэтому точно задать для нее верхнюю границу ( $X_{\max}$ ) невозможно. Попробуем использовать в качестве верхней границы значение 120. Таким образом,  $X_{\min}=0$ ,  $X_{\max}=120$ . Разобьем этот диапазон на  $k=12$  интервалов: (0; 10), (10; 20), ..., (110; 120). Длина интервала  $d=10$ .

Внесем необходимые изменения в программу для построения гистограммы. Имитация времени охлаждения плиты будет выполняться с помощью подпрограммы `expon`, приведенной в п.3.2.1. Границы интервалов задаются в массиве `gran`. Требуется также изменить переменные  $k$  (количество интервалов) и  $d$  (длина интервала). Кроме того, так как верхняя граница (120) задана приближенно, следует предусмотреть подсчет значений, превышающих эту границу; для этого используется переменная `zaverhgran`.

```
Sub primer33_2()
Dim gran(1 To 13), f(1 To 13), m(1 To 13)
gran(1) = 0: gran(2) = 10: gran(3) = 20: gran(4) = 30: gran(5) = 40: gran(6) = 50 : gran(7) = 60
gran(8) = 70: gran(9) = 80: gran(10) = 90: gran(11) = 100: gran(12) = 110 : gran(13) = 120
k = 12: n = 1000: d = 10
For i = 1 To n
Call expon(30, x)
For j = 1 To k
If (x > gran(j)) And (x <= gran(j + 1)) Then m(j) = m(j) + 1
Next j
If (x > gran(k+1)) Then zaverhgran= zaverhgran + 1
Next i
MsgBox ("Частоты")
For i = 1 To k
MsgBox (Str(gran(i)) + "-" + Str(gran(i + 1)) + ": " + Str(m(i)))
Next i
MsgBox ("Относительные частоты")
For i = 1 To k
f(i) = m(i) / (n * d)
MsgBox (Str(gran(i)) + "-" + Str(gran(i + 1)) + ": " + Str(Round(f(i),4)))
Next I
MsgBox ("Превышают верхнюю границу: "+Str(zaverhgran))
End Sub

Sub expon(xsred, x)
r = Rnd
x = -xsred * Log(r)
End Sub
```

Результаты выполнения программы приведены в табл.3.2.



Таблица 3.2

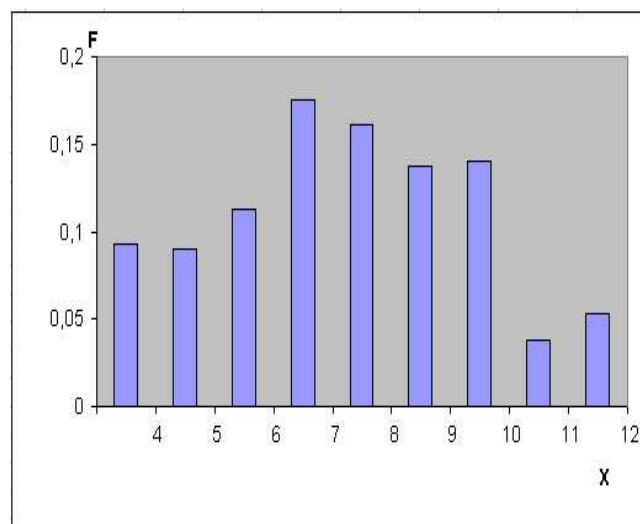
Интервал	Частота, $m_i$	Относительная частота, $f_i$	Интервал	Частота, $m_i$	Относительная частота, $f_i$
0 – 10	285	0,0285	70 – 80	30	0,0030
10 – 20	192	0,0192	80 – 90	27	0,0027
20 – 30	149	0,0149	90 – 100	13	0,0013
30 – 40	108	0,0108	100 – 110	9	0,0009
40 – 50	88	0,0088	110 – 120	2	0,0002
50 – 60	55	0,0055	Более 120	13	-
60 – 70	29	0,0029			

Как видно, почти все значения, полученные при моделировании, попадают в диапазон (0; 120); за пределами этого диапазона оказалось только 13 значений. Поэтому можно использовать полученные результаты для построения гистограммы (рис.3.2,б).

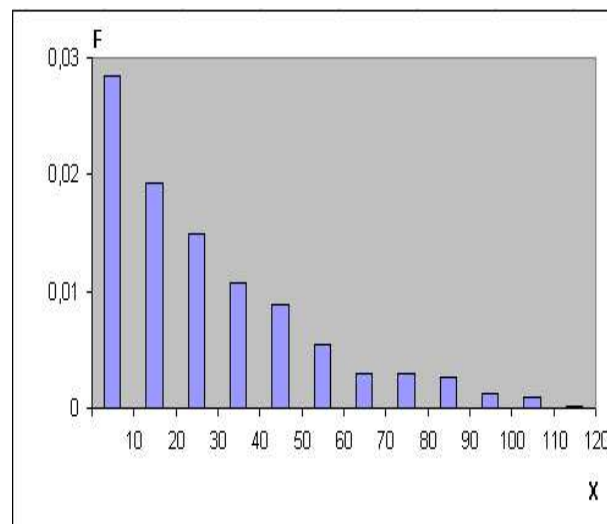
Примечание. Если бы количество значений, выходящих за указанный диапазон ( $X_{\min}$ ;  $X_{\max}$ ), оказалось большим, то потребовалось бы расширить этот диапазон (в данном случае – увеличить верхнюю границу).

Гистограмма приведена на рис.3.2,б. Видно, что она близка к графику плотности распределения экспоненциальной случайной величины. Таким образом, можно предполагать, что алгоритм имитации времени охлаждения плиты также правильный.

Примечание. Основное назначение гистограммы распределения состоит не в проверке правильности имитации случайной величины, а в наглядном представлении значений случайной величины и частот их появления.



а)



б)

Рис.3.2. Гистограммы для проверки алгоритмов имитации, полученных в примере 3.2:  
а) для времени внесения добавок; б) для времени охлаждения плиты.

### **Проверка алгоритмов имитации на основе критерия “хи-квадрат”**

Критерий "хи-квадрат" ( $\chi^2$ ) представляет собой один из методов математической статистики. Одна из возможностей его применения состоит в сле-

дующем. Пусть имеется набор значений некоторой случайной величины. Критерий "хи-квадрат" позволяет проверить гипотезу о том, что случайная величина, для которой получен этот набор, распределена по некоторому конкретному закону. Таким образом, критерий "хи-квадрат" позволяет проверить, соответствуют ли значения случайной величины, полученные в ходе имитации, тому закону распределения, который требовалось имитировать.

Примечание. "Хи-квадрат" - это название одного из распределений непрерывных случайных величин, используемого при проверке по данному критерию.

Проверка соответствия случайной величины заданному закону распределения на основе критерия "хи-квадрат" выполняется в следующем порядке.

1. Определяется диапазон значений случайной величины ( $X_{\min}$ ;  $X_{\max}$ ). Этот диапазон разбивается на  $k$  интервалов и подсчитывается количество попаданий имитируемой случайной величины в каждый из этих интервалов (частоты  $m_i$ ,  $i=1, \dots, k$ ). См. алгоритм построения гистограммы (п.3.2.4), шаги 1-3.

2. По заданному закону распределения имитируемой случайной величины вычисляются *теоретические* вероятности ее попадания в каждый из интервалов:  $P_i$ ,  $i=1, \dots, k$ . Для этого может использоваться, например, формула (ПЗ.4) или (ПЗ.5).

3. Вычисляется значение критерия "хи-квадрат" по формуле

$$\chi^2 = \sum_{i=1}^k \frac{M_i^2}{n \cdot P_i} - n, \quad (3.17)$$

где  $n$  - количество значений случайной величины (количество испытаний).

4. По таблицам распределения "хи-квадрат" определяется квантиль этого распределения  $\chi_{\alpha, s}^2$ . Здесь  $\alpha$  - уровень значимости. Величина  $\alpha$ , как правило, задается из диапазона от 0,01 до 0,1. Чем меньшей задается величина  $\alpha$ , тем более строгой является проверка. Обычно используется значение  $\alpha=0,05$ . Величина  $s$  - число степеней свободы распределения "хи-квадрат". Эта величина определяется следующим образом:  $s=k-1$ .

Квантиль  $\chi_{\alpha, s}^2$  можно также найти с помощью табличного процессора Excel, используя функцию ХИ2ОБР с аргументами Вероятность -  $\alpha$ , Степени свободы -  $s$ .

5. Значение критерия "хи-квадрат" сравнивается с квантилем  $\chi_{\alpha, s}^2$ . Если  $\chi^2 < \chi_{\alpha, s}^2$ , то можно утверждать, что значения случайной величины, полученные при моделировании, соответствуют заданному закону распределения. В этом случае можно считать, что алгоритм имитации правильный. Если условие  $\chi^2 < \chi_{\alpha, s}^2$  не выполняется, это означает, что значения, полученные при моделировании, существенно отклоняются от заданного закона распределения. Это может указывать на ошибку в алгоритме имитации.

**Пример 3.5.** Требуется проверить алгоритмы имитации случайных величин, разработанные в примере 3.2, используя критерий “хи-квадрат”.

Выполним **проверку алгоритма имитации времени внесения добавок.** Частоты  $m_i$  получены ранее (см. табл.3.1). Используя плотность распределения (\*) или функцию распределения (\*\*), указанные в п.3.2.1, найдем теоретические вероятности попадания времени внесения добавок в интервалы (3; 4), (4; 5), ..., (11; 12). Для этого воспользуемся формулой (ПЗ.4) или (ПЗ.5). Например, используя формулу (ПЗ.5), вероятность попадания в интервал (3; 4) можно найти так:

$$P_1 = \int_3^4 0,1 dx = 0,1.$$

Аналогично вычисляются остальные вероятности  $P_2, \dots, P_9$ . Они приведены в табл. 3.3.

Таблица 3.3

Интервал	3 – 4	4 – 5	5 – 6	6 – 7	7 – 8	8 – 9	9 – 10	10 – 11	11 – 12
Вероятность, $P_i$	0,1	0,1	0,1	0,15	0,15	0,15	0,15	0,05	0,05

Найдем значение критерия “хи-квадрат” по формуле (3.17):

$$\chi^2 = \frac{93^2}{1000 \cdot 0,1} + \frac{90^2}{1000 \cdot 0,1} + \dots + \frac{53^2}{1000 \cdot 0,05} - 1000 = 13,01.$$

Найдем квантиль распределения “хи-квадрат”  $\chi_{\alpha,s}^2$ . Зададим уровень значимости  $\alpha=0,05$ . Число степеней свободы определяется следующим образом:  $s=k-1=9-1=8$ . Квантиль  $\chi_{0,05;8}^2$  можно найти из таблиц распределения “хи-квадрат” или с помощью табличного процессора Excel, используя функцию ХИ2ОБР с аргументами Вероятность – 0,05, Степени свободы – 8. В данном примере  $\chi_{\alpha,s}^2 = \chi_{0,05;8}^2 = 15,507$ . Так как  $\chi^2 < \chi_{\alpha,s}^2$ , можно утверждать, что случайные величины, получаемые с помощью алгоритма имитации времени внесения добавок (и вычисляемые с помощью подпрограммы vnesdob), соответствуют плотности распределения (3.3). Таким образом, алгоритм имитации времени внесения добавок можно признать правильным.

Выполним **проверку алгоритма имитации времени охлаждения плиты.** Частоты  $m_i$  приведены в табл.3.2. Используя плотность распределения (3.6) или функцию распределения (3.7), найдем теоретические вероятности попадания времени охлаждения плиты в интервалы (0; 10), (10; 20), ..., (110; 120). Например, вероятность попадания в интервал (0; 10) можно найти по формуле (ПЗ.5) следующим образом:

$$P_1 = \int_0^{10} \frac{1}{30} e^{-\frac{1}{30}x} dx = 0,283.$$

Аналогично вычисляются остальные вероятности:  $P_2, \dots, P_{12}$ . Они приведены в табл. 3.4.

Таблица 3.4

Интервал	0 - 10	10 - 20	20 - 30	30 - 40	40 - 50	50 - 60	60 - 70	70- 80	80- 90	90- 100	100- 110	110- 120
Вероятность, $P_i$	0,283	0,203	0,146	0,104	0,075	0,054	0,038	0,027	0,020	0,014	0,010	0,007

Найдем значение критерия “хи-квадрат” по формуле (3.17):

$$\chi^2 = \frac{285^2}{1000 \cdot 0,283} + \frac{192^2}{1000 \cdot 0,192} + \dots + \frac{2^2}{1000 \cdot 0,0007} - 1000 = 4,778.$$

Найдем квантиль распределения “хи-квадрат”:  $\chi^2_{\alpha,s} = \chi^2_{0,05;11} = 19,675$ . Так как  $\chi^2 < \chi^2_{\alpha,s}$ , можно утверждать, что случайные величины, получаемые с помощью алгоритма имитации времени охлаждения плит (и вычисляемые с помощью подпрограммы expon), соответствуют плотности распределения (3.6). Таким образом, алгоритм имитации времени охлаждения плит также можно признать правильным.

После проверки алгоритмов имитации они могут применяться для решения практических задач анализа и оптимизации объектов моделирования.

### 3.3. Примеры решения задач на основе алгоритмов имитации случайных величин

**Пример 3.6.** По данным примера 3.2 требуется разработать алгоритм и программу для имитации процесса производства пластмассовых плит. Определить среднее время изготовления одной плиты.

При разработке модели будем считать, что время изготовления плиты складывается из двух случайных величин: времени внесения добавок и времени охлаждения плиты в форме.

Приведем алгоритм имитации производства плит.

1. Выполняется имитация времени внесения добавок ( $t_{\text{внес}}$ ). Для этого используется формула (3.5).

2. Выполняется имитация времени охлаждения плиты в форме ( $t_{\text{охлажд}}$ ): экспоненциальная случайная величина со средним значением 30 мин. Для этого используется формула (3.10) при  $\bar{X} = 30$ .

3. Определяется время изготовления плиты:  $t = t_{\text{внес}} + t_{\text{охлажд}}$ .

Шаги 1-3 повторяются многократно (например, 100 000 раз). Определяется среднее время выпуска *одной* плиты: отношение *суммы времен выпуска всех плит* к *количеству* этих плит (т.е. к количеству испытаний).

Рассмотрим одно испытание алгоритма имитации.

Выполняется имитация времени внесения добавок по формуле (3.5). Пусть разыграно  $R = 0,0795$ . Тогда  $t_{\text{внес}} = (0,0795 + 0,3) / 0,1 = 3,795$  мин.

Выполняется имитация времени охлаждения плиты по формуле (3.10). Пусть разыграно  $R=0,3780$ . Тогда  $t_{\text{охлажд}} = -30 \cdot \ln(0,3780) = 29,186$  мин.

Таким образом, в данном испытании смоделировано время изготовления плиты  $t = 3,795 + 29,186 = 32,981$  мин.

Приведем программу на языке VBA, реализующую алгоритм имитации производства плит.

```
Sub primer36()  
'Имитация выпуска пластмассовых плит  
n = 100000  
tzs = 30  
tsum = 0  
For i = 1 To n  
t = 0  
Call vnesdob(tvnes)      'Имитация внесения добавок  
t = t + tvnes  
Call expon(tzs, tohlazd)  'Имитация охлаждения  
t = t + tohlazd  
tsum = tsum + t  
Next i  
tsred = tsum / n  
MsgBox ("Среднее время изготовления плиты " + Str(Round(tsred,2)))  
End Sub  
  
Sub vnesdob(x)  
r = Rnd  
If r < 0.3 Then x = (r + 0.3) / 0.1  
If (r >= 0.3) And (r < 0.9) Then x = (r + 0.6) / 0.15  
If r >= 0.9 Then x = (r - 0.4) / 0.05  
End Sub  
  
Sub expon(xsred, x)  
r = Rnd  
x = -xsred * Log(r)  
End Sub
```

Среднее время изготовления плиты, полученное по результатам моделирования, составляет около 37 минут.

**Пример 3.7.** В ремонтной службе предприятия выполняется наладка и ремонт некоторых приборов. Каждый прибор состоит из пяти схем. Каждая из схем может оказаться неисправной с вероятностью 0,1 (т.е. в 10% случаев). Наладка и ремонт прибора включает следующие операции:

- осмотр: от 2 до 5 мин на каждую схему;
- замена неисправных схем: время замены одной схемы - гауссовская случайная величина со средним значением 5 мин и стандартным отклонением 1 мин;
- наладка. Если в приборе не потребовалась замена схем, то выполняется мелкая наладка; если была заменена хотя бы одна схема, то необходима полная наладка. Время наладки можно считать случайной величиной, распределенной

по экспоненциальному закону. Мелкая наладка занимает в среднем 10 мин, полная – 15 мин.

Требуется разработать алгоритм и программу имитации наладки и ремонта приборов на основе метода Монте-Карло. Определить: а) среднее время наладки и ремонта одного прибора; б) процент случаев, когда требуется полная наладка; с) среднее количество схем, заменяемых в одном приборе.

При разработке модели будем считать, что время наладки и ремонта прибора складывается из нескольких случайных величин: времени осмотра всех схем, времени замены неисправных схем (если такие схемы есть), времени наладки прибора.

Приведем алгоритм имитации процесса наладки и ремонта приборов.

1. Выполняется имитация осмотра пяти схем прибора. Время осмотра каждой схемы - случайная величина, распределенная по равномерному закону в интервале (2; 5). Для имитации такой величины используется формула (3.11), при  $a=2$ ,  $b=5$ . Кроме того, имитируется результат осмотра (исправное или неисправное состояние схемы). Для этого разыгрывается СРРЧ  $R$ ; если  $R < 0,1$ , то схема неисправна, в противном случае – исправна. Действия этого шага повторяются пять раз (для каждой схемы).

2. Если на шаге 1 была смоделирована неисправность хотя бы одной схемы, то выполняется шаг 3, в противном случае – шаг 4.

3. Выполняется имитация замены неисправных схем. Для этого имитируется гауссовская случайная величина по формуле (3.16) при  $m=5$ ,  $\sigma=1$ . Действия этого шага повторяются столько раз, сколько неисправных схем было смоделировано на шаге 1.

4. Выполняется имитация наладки прибора. Для этого используется формула (3.10) при  $\bar{X}=10$ , если на шаге 1 не было смоделировано ни одной неисправной схемы, или  $\bar{X}=15$ , если на шаге 1 была смоделирована хотя бы одна неисправная схема.

5. Вычисляется время наладки и ремонта прибора: сумма значений времени, смоделированных на шагах 1,3,4.

Шаги 1-5 повторяются многократно (например, 100 000 раз). Вычисляются величины, которые требуется определить по результатам моделирования. *Среднее время наладки и ремонта одного прибора* определяется как отношение суммы времени наладки и ремонта всех приборов к количеству этих приборов (т.е. к количеству испытаний). *Процент случаев полной наладки* вычисляется как отношение количества случаев полной наладки к количеству испытаний (умноженное на 100). *Среднее количество схем, заменяемых в одном приборе*, определяется как отношение общего количества неисправных схем к количеству испытаний.

Приведем программу на языке VBA, реализующую алгоритм имитации наладки и ремонта приборов.

Sub primer37()

‘ Имитация наладки и ремонта приборов



```

n = 100000
tpoln = 15: tmelk = 10
tsum = 0: kolneisp = 0
For i = 1 To n
kolneisp1 = 0: t = 0
For j = 1 To 5
Call ravn(2, 5, tosm)           'Имитация осмотра всех схем прибора
t = t + tosm
r = Rnd
If r < 0.1 Then kolneisp1 = kolneisp1 + 1 'Подсчет количества неисправных схем в приборе
Next j
kolneisp = kolneisp + kolneisp1 'Подсчет количества неисправных схем во всех приборах
If kolneisp1 > 0 Then           'Проверка: есть ли в приборе неисправные схемы
For j = 1 To kolneisp1         'Имитация замены неисправных схем
Call gauss(5, 1, tzam)
t = t + tzam
Next j
Call expon(tpoln, tnal)        'Имитация полной наладки
t = t + tnal
kolpn = kolpn + 1
Else
Call expon(tmelk, tnal)        'Имитация мелкой наладки
End If
tsum = tsum + t
Next i
tsred = tsum / n
ppn = (kolpn / n) * 100
kolzamsred = kolneisp / n
MsgBox ("Среднее время наладки и ремонта прибора: " + Str(Round(tsred,2)))
MsgBox ("Процент случаев полной наладки: " + Str(Round(ppn,2)))
MsgBox ("Среднее количество заменяемых схем на один прибор: "+Str(Round(kolzamsred,2)))
End Sub

Sub ravn(a, b, x)
r = Rnd
x = a + (b - a) * r
End Sub

Sub expon(xsred, x)
r = Rnd
x = -xsred * Log(r)
End Sub

Sub gauss(m, sigma, x)
k = 6
For i = 1 To k
r = Rnd: rsum = rsum + r
Next i
x = m + sigma * Sqr(12 / k) * (rsum - k / 2)
End Sub

```

Здесь подпрограмма ravn реализует алгоритм имитации равномерной случайной величины (формула (3.11)).



Результаты моделирования следующие: среднее время ремонта – около 23,61 мин, процент случаев полной наладки – 40,9%, среднее количество заменяемых схем в одном приборе – 0,5.

**Пример 3.8.** Некоторые изделия изготавливаются путем наложения двух пластин. Номинальная толщина первой пластины – 6 мм, второй – 4 мм. Так как в производственном процессе неизбежны отклонения, фактически толщина первой пластины представляет собой гауссовскую случайную величину со средним значением 6 мм и стандартным отклонением 0,2 мм. Толщина второй пластины – также гауссовская случайная величина со средним значением 4 мм и стандартным отклонением 0,1 мм. Готовое изделие считается годным, если его толщина составляет от 9,7 до 10,3 мм.

Требуется разработать алгоритм и программу имитации выпуска изделий на основе метода Монте-Карло. Определить среднюю толщину готового изделия и вероятность выпуска годного изделия.

Приведем алгоритм имитации процесса изготовления изделий.

1. Выполняется имитация толщины первой пластины ( $X_1$ ). Для этого имитируется гауссовская случайная величина по формуле (3.16) при  $m=6$ ,  $\sigma=0,2$ .

2. Выполняется имитация толщины второй пластины ( $X_2$ ). Для этого имитируется гауссовская случайная величина по формуле (3.16) при  $m=4$ ,  $\sigma=0,1$ .

3. Определяется толщина изделия:  $X=X_1+X_2$ .

4. Если  $9,7 \leq X \leq 10,3$ , то изделие годное, в противном случае – бракованное.

Шаги 1-4 повторяются многократно (например, 100 000 раз). Вычисляются величины, которые требуется определить по результатам моделирования. *Средняя толщина готового изделия* определяется как отношение суммы значений толщины всех изделий к количеству этих изделий (т.е. к количеству испытаний). *Вероятность выпуска годного изделия* вычисляется как отношение количества годных изделий к количеству испытаний.

Приведем программу на языке VBA, реализующую алгоритм имитации изготовления изделий.

```
Sub primer38()  
    ' Имитация изготовления изделий из двух пластин  
    n = 100000  
    xsum = 0: kolgodn = 0  
    For i = 1 To n  
        Call gauss(6, 0.2, x1)      'Имитация толщины первой пластины  
        Call gauss(4, 0.1, x2)      'Имитация толщины второй пластины  
        x=x1+x2                     'Определение толщины изделия  
        If (x>=9.7) And (x<=10.3) Then kolgodn=kolgodn+1 'Имитация проверки изделия  
        xsum=xsum+x  
    Next i  
    xsred = xsum / n  
    pgodn = kolgodn / n  
    MsgBox ("Средняя толщина изделия: " + Str(Round(xsred,2)))  
    MsgBox ("Вероятность выпуска годного изделия: " + Str(Round(pgodn,4)))  
End Sub
```

```

Sub gauss(m, sigma, x)
k = 6
For i = 1 To k
r = Rnd: rsum = rsum + r
Next i
x = m + sigma * Sqr(12 / k) * (rsum - k / 2)
End Sub

```

Результаты моделирования следующие: средняя толщина изделия – примерно 10 мм, вероятность выпуска годного изделия – 0,82.

**Пример 3.9.** В автоматизированной системе управления технологическим процессом передаются сигналы от производственного оборудования (объекта управления) к управляющему компьютеру. Длительность передачи сигнала – случайная величина, распределенная по экспоненциальному закону со средним значением 5 мс. В канале связи возможны помехи. Интервалы между моментами помех – случайные величины, распределенные по экспоненциальному закону. Помехи возникают в среднем 20 раз в секунду. Если во время передачи сигнала возникает хотя бы одна помеха, то сигнал искажается.

Требуется разработать алгоритм и программу имитации передачи сигналов на основе метода Монте-Карло. Определить вероятность передачи сигнала без искажений.

Так как *интервалы времени* между помехами представляют собой экспоненциальные случайные величины, *количество* помех за некоторый интервал времени  $t$  представляет собой случайную величину, распределенную по пуассоновскому закону с параметром  $\lambda = \Lambda t$ , где  $\Lambda$  – интенсивность потока помех. В данном примере  $\Lambda = 20$  помех/с, или 0,02 помехи/мс.

Приведем алгоритм имитации процесса передачи сигналов.

1. Выполняется имитация времени передачи сигнала ( $t$ ). Для этого разыгрывается экспоненциальная случайная величина по формуле (3.10) при  $\bar{X} = 5$ .

2. Разыгрывается количество помех ( $m$ ) за время передачи сигнала  $t$ . Для этого используется алгоритм имитации пуассоновской случайной величины, приведенный в подразд. 3.1, при  $\lambda = 0,02t$ .

3. Если  $m = 0$  (за время передачи сигнала не было ни одной помехи), то сигнал не искажен.

Шаги 1-3 повторяются многократно (например, 100 000 раз). Вычисляется вероятность передачи сигнала без искажений как отношение количества сигналов, переданных без искажений, к общему количеству смоделированных сигналов (количеству испытаний).

Приведем программу на языке VBA, реализующую алгоритм имитации передачи сигналов.

```

Sub primer 39()
‘ Имитация передачи сигналов в АСУТП
n = 100000
tsred = 5 : intens = 0.02
For i = 1 To n

```

```

Call expon(tsred,t)  'Имитация времени передачи сигнала
l=intens*t          'Вычисление параметра пуассоновского распределения
Call puasson(l, m)  'Розыгрыш количества помех за время передачи сигнала
If m=0 then kolnorm=kolnorm+1  'Если помех не было, то сигнал передан без искажений
Next i
pnorm = kolnorm / n
MsgBox ("Вероятность нормальной передачи сигнала: " + Str(Round(pnorm,4)))
End Sub

Sub expon(xsred, x)
r = Rnd
x = -xsred * Log(r)
End Sub

Sub puasson(l, m)  'Подпрограмма имитации пуассоновской случайной величины
p0=exp(-l)        'на основе алгоритма, приведенного в подразд. 3.1
q=1 : m=0
povtor:
  r=Rnd
  q=q*r
  If q>=p0 Then
    m=m+1
    Goto povtor
  End If
End Sub

```

По результатам моделирования, вероятность передачи сигнала без искажений составляет примерно 0,909.

**Пример 3.10.** Предприятие выпускает электроприборы, состоящие из четырех блоков. Прибор продолжает работать, пока исправен хотя бы один блок. Время безотказной работы каждого блока - случайная величина, распределенная по экспоненциальному закону; среднее время безотказной работы блока – два года. Срок действия гарантии на прибор - один год.

Требуется разработать алгоритм и программу имитации работы прибора на основе метода Монте-Карло. Определить: а) среднее время безотказной работы прибора; б) вероятность безотказной работы прибора в течение гарантийного срока.

При разработке алгоритма решения этой задачи под временем безотказной работы прибора будем понимать время, в течение которого прибор продолжает работать (даже если некоторые из блоков, входящих в состав прибора, отказали).

Приведем алгоритм имитации работы прибора.

1. Разыгрываются значения времени безотказной работы каждого из блоков ( $t_{rb1}, t_{rb2}, t_{rb3}, t_{rb4}$ ) – четыре случайных величины, распределенные по экспоненциальному закону, со средним значением 2. Для этого используется формула (3.10) при  $\bar{X} = 2$ .

2. Определяется время безотказной работы прибора – максимальная из величин, разыгранных на шаге 1 (так как прибор продолжает работать, пока исправен хотя бы один из блоков):  $t_{rp} = \max(t_{rb1}, t_{rb2}, t_{rb3}, t_{rb4})$ .

3. Если  $t_{rp} \geq 1$  (т.е. не менее одного года), то прибор отработал гарантийный срок без отказа.

Шаги 1-3 повторяются многократно (например, 100 000 раз). По окончании испытаний вычисляются величины, которые требуется определить по результатам моделирования. *Среднее время безотказной работы прибора* определяется как отношение суммы значений времени безотказной работы всех приборов (т.е. суммы величин  $t_{rp}$ ) к количеству этих приборов. *Вероятность безотказной работы в течение гарантийного срока* вычисляется как отношение количества приборов, отработавших гарантийный срок без отказа, к общему количеству приборов.

Приведем программу на языке VBA, реализующую алгоритм имитации работы приборов.

```
Sub primer310()  
' Имитация работы приборов в течение гарантийного срока  
Dim trb(1 To 4)  
n = 100000  
tsred = 2  
For i = 1 To n  
  For j = 1 To 4  
    Call expon(tsred, trb(j))      'Имитация времени безотказной работы блока  
  Next j  
  trp = 0  
  For j = 1 To 4                  'Определение времени работы прибора -  
    If trp < trb(j) Then trp = trb(j)  'максимального из значений времени работы блоков  
  Next j  
  tsum = tsum + trp  
  If trp >= 1 Then bezotk = bezotk + 1 'Подсчет приборов, отработавших гарантийный срок  
Next i  
trpsred = tsum / n  
pbezotk = bezotk / n  
MsgBox ("Среднее время безотказной работы прибора: " + Str(Round(trpsred,2)))  
MsgBox ("Вероятность безотказной работы в течение гарантийного срока: " + Str(Round(pbezotk,4)))  
End Sub  
  
Sub expon(xsred, x)  
r = Rnd  
x = -xsred * Log(r)  
End Sub
```

По результатам моделирования среднее время безотказной работы прибора составляет примерно 4,17 года, а вероятность безотказной работы в течение гарантийного срока - 0,9754.

### 3.4. Интервальные оценки в имитационных моделях

Величины, определяемые по результатам имитационного моделирования, не могут быть абсолютно точными. Выполнив последовательность испытаний имитационной модели (например, 1000 или 100 000 испытаний) и получив значения исследуемых величин (например, среднюю прибыль от выпуска изделия  $\bar{X}$  и вероятность выпуска годного изделия  $P$ ), важно понимать, что по результатам *другой* последовательности испытаний были бы получены *другие* значения этих величин. Для получения *истинного* (абсолютно точного) значения потребовалось бы (теоретически) *бесконечно большое* количество испытаний. Поэтому представляют интерес оценки, полученные с заданной точностью, а также интервальные оценки.

Конкретные значения исследуемых величин, полученные по результатам имитационного моделирования, представляют собой **точечные оценки**. Все результаты, полученные в рассмотренных выше примерах, являются точечными оценками.

Под *точностью* оценки будем понимать ее *отклонение от истинного значения* исследуемой величины (т.е. разность между полученной оценкой и истинным значением), а под *относительной точностью* оценки – то же *отклонение*, выраженное *в процентах* (или в долях) от *максимально возможного отклонения*. Будем обозначать точность как  $\varepsilon_\beta$ , а относительную точность – как  $\varepsilon_{\beta 0}$  (смысл обозначения  $\beta$  будет показан ниже). При этом важно понимать, что истинное значение исследуемой величины не может быть известно точно.

Примечание. Максимально возможное отклонение полученной точечной оценки от истинного значения исследуемой величины определяется по-разному в зависимости от того, какую величину требуется определить по результатам моделирования. Например, можно доказать, что максимально возможное отклонение *точечной оценки среднего значения* исследуемой величины от *истинного значения среднего* можно считать равным  $3\sigma$ , где  $\sigma$  – стандартное отклонение данной величины. Максимально возможное отклонение *точечной оценки вероятности* некоторого события от *истинного значения* этой вероятности равно единице.

В практических задачах обычно требуется получение таких оценок, чтобы их *отклонение* от истинного (точно не известного) значения *не превышало некоторую предельную величину с заданной* (достаточно высокой) *вероятностью*. Эту вероятность будем обозначать как  $\beta$ .

Для решения такой задачи необходимо задать требуемую относительную точность оценки, т.е. величину  $\varepsilon_{\beta 0}$ , обычно выражаемую в процентах. Как правило, требуется значение  $\varepsilon_{\beta 0}$  на уровне 3-5%. Например, если задана относительная точность  $\varepsilon_{\beta 0}=5\%$  и доверительная вероятность  $\beta=0,995$ , это значит, что требуется получить оценку, отклонение которой от истинного значения исследуемой величины с вероятностью 0,995 будет составлять не более 5% от максимально возможного значения такого отклонения.

Указав заданную относительную точность  $\varepsilon_{\beta 0}$  и доверительную вероятность  $\beta$ , можно определить *количество испытаний*, необходимое для обеспече-

ния заданной относительной точности. Очевидно, что чем выше требуемая точность, тем больше необходимое количество испытаний.

Как видно из примеров, рассмотренных в предыдущих подразделах, в задачах моделирования обычно требуется определять оценки *среднего значения* некоторой величины  $\bar{X}$  или *вероятности* некоторого события  $P$ . Можно доказать, что количество испытаний  $n$ , необходимое для получения этих оценок с заданной относительной точностью при доверительной вероятности  $\beta=0,995$ , определяется по следующим формулам:

- для среднего значения:

$$n = \frac{1}{\varepsilon_{\beta 0}^2}; \quad (3.18)$$

- для вероятности:

$$n = \frac{2}{\varepsilon_{\beta 0}^2}, \quad (3.19)$$

где  $\varepsilon_{\beta 0}$  - заданная относительная точность оценки.

Выполнив необходимое количество испытаний  $n$ , можно определить точность полученной оценки  $\varepsilon_{\beta}$  с заданной доверительной вероятностью  $\beta$ .

Для **среднего значения** точность оценки определяется по следующей формуле:

$$\varepsilon_{\beta} = 3\varepsilon_{\beta 0} \cdot \sigma. \quad (3.20)$$

Здесь  $\sigma$  – стандартное отклонение, вычисляемое по формуле:

$$\sigma = \sqrt{\frac{1}{n-1} \sum_{i=1}^n (x_i - \bar{X})^2}, \quad (3.21)$$

где  $x_i, i=1, \dots, n$  – значения исследуемой величины, полученные в каждом из  $n$  испытаний в ходе имитационного моделирования;

$\bar{X}$  - среднее значение, вычисленное по результатам  $n$  испытаний.

Для программной реализации более удобна следующая формула для вычисления стандартного отклонения, которую можно вывести из формулы (3.21):

$$\sigma = \sqrt{\frac{n}{n-1} \left( \frac{\sum_{i=1}^n x_i^2}{n} - \bar{X}^2 \right)}. \quad (3.22)$$



Для **вероятности** точность оценки определяется по следующей формуле:

$$\varepsilon_P = 2,8 \sqrt{\frac{P(1-P)}{n}}, \quad (3.23)$$

где  $P$  – значение вероятности, вычисленное по результатам  $n$  испытаний.

Как показано в примерах выше, вероятность некоторого события  $P$  вычисляется по результатам имитационного моделирования как отношение количества случаев, когда было смоделировано наступление этого события, к общему количеству испытаний ( $n$ ).

Вычислив точность  $\varepsilon_P$ , можно получить интервальную оценку исследуемой величины. **Интервальная оценка** некоторой величины – это числовой интервал, содержащий *истинное* значение этой величины с *заданной* (достаточно высокой) *доверительной вероятностью*.

Примечание. Важно понимать, что построение интервальной оценки с доверительной вероятностью, равной единице, невозможно: для этого требуется бесконечное количество испытаний.

Интервальная оценка для **среднего значения** определяется следующим образом:

$$\bar{X} \pm \varepsilon_P, \quad (3.24)$$

где  $\varepsilon_P$  – точность оценки, найденная по формуле (3.20).

Интервальная оценка для **вероятности** определяется следующим образом:

$$P \pm \varepsilon_P, \quad (3.25)$$

где  $\varepsilon_P$  – точность оценки, найденная по формуле (3.23).

Таким образом, интервальные оценки могут быть получены с использованием имитационного моделирования в следующем порядке.

1. Задается требуемая относительная точность оценки  $\varepsilon_{P0}$ .
2. Определяется необходимое количество испытаний  $n$  по формуле (3.18) или (3.19).
3. Выполняется  $n$  испытаний имитационной модели.
4. По результатам имитационного моделирования вычисляются точечные оценки  $\bar{X}$  или  $P$ .
5. Определяется точность полученных оценок по формулам (3.20) или (3.23).
6. Определяются интервальные оценки по формулам (3.24) или (3.25).

**Пример 3.11.** В условиях примера 3.8 требуется определить среднюю толщину готового изделия и вероятность выпуска годного изделия с относительной точностью 5%.



Найдем необходимое количество испытаний. Количество испытаний для определения *средней* толщины готового изделия вычисляется по формуле (3.18):  $n=1/0,05^2=400$ . Количество испытаний для определения *вероятности* выпуска годного изделия вычисляется по формуле (3.19):  $n=2/0,05^2=800$ . Так как необходимо обеспечить заданную точность для обеих величин, потребуется выполнить 800 испытаний.

Примечание. Если бы в задаче требовалось определить с заданной относительной точностью только среднюю толщину, то было бы достаточно 400 испытаний.

В программе, реализующей имитационную модель, необходимо предусмотреть вычисление величин точности по формулам (3.20) и (3.23) и интервальных оценок по формулам (3.24) и (3.25). Чтобы вычислить точность оценки средней толщины изделия, потребуется также предусмотреть вычисление стандартного отклонения по формуле (3.22). Программа будет иметь следующий вид.

```
Sub primer311()
' Имитация изготовления изделий из двух пластин
n = 800 : epsb0=0.05
xsum = 0: kolgodn = 0
For i = 1 To n
Call gauss(6, 0.2, x1)
Call gauss(4, 0.1, x2)
x=x1+x2
If (x>=9.7) And (x<=10.3) Then kolgodn=kolgodn+1
xsum=xsum+x
xsum2=xsum2+x^2
Next i
xsred = xsum / n
pgodn = kolgodn / n
sigma=sqr((n/(n-1))*(xsum2/n-xsred^2))
epsb_sred=3*epsb0*sigma
xniz=xsred-epsb_sred : xverh=xsred+epsb_sred
epsb_ver=2.8*sqr(pgodn*(1-pgodn)/n)
pniz=pgodn-epsb_ver : pverh=pgodn+epsb_ver
MsgBox ("Средняя толщина изделия: " + Str(Round(xsred,2)))
MsgBox ("Интервальная оценка средней толщины: " + Str(Round(xniz,2))+” – “+ Str(Round(xverh,2)))
MsgBox ("Вероятность выпуска годного изделия: " + Str(Round(pgodn,4)))
MsgBox ("Интервальная оценка вероятности: " + Str(Round(pniz,4))+” – “+Str(Round(pverh,4)))
End Sub

Sub gauss(m, sigma, x)
k = 6
For i = 1 To k
r = Rnd: rsum = rsum + r
Next i
x = m + sigma * Sqr(12 / k) * (rsum - k / 2)
End Sub
```

Здесь переменная  $xsum2$  используется для вычисления суммы квадратов значений толщины. Эта сумма требуется для вычисления стандартного отклонения по формуле (3.22). Смысл остальных переменных, добавленных в программу, следующий:  $epsb0=0,05$  – заданная относительная точность;  $\sigma$  (в основной программе) – стандартное отклонение, вычисляемое по формуле (3.22);  $epsb\_sred$ ,  $epsb\_ver$  – значения точности оценок средней толщины и вероятности, вычисляемые по формулам (3.20) и (3.23) соответственно;  $xniz$ ,  $xverh$  – границы доверительного интервала для средней толщины, вычисляемые по формуле (3.24);  $pniz$ ,  $pverh$  – границы доверительного интервала для вероятности, вычисляемые по формуле (3.25).

Результаты моделирования будут примерно следующими: средняя толщина изделия – 9,99 мм, интервальная оценка – (9,96; 10,02) мм; вероятность выпуска годного изделия – 0,8338, интервальная оценка – (0,7969; 0,8706). Это означает, что с вероятностью 0,995 (или 99,5%) истинная средняя толщина изделий будет составлять от 9,96 до 10,02 мм, а вероятность выпуска годного изделия – от 0,7969 до 0,8706. С вероятностью 0,995 разность между точечной оценкой средней толщины, полученной в результате моделирования (9,99 мм), и истинным (точно не известным) значением средней толщины составляет не более 5% от максимально возможного значения такой разности. Аналогично, с вероятностью 0,995 разность между точечной оценкой вероятности выпуска годного изделия (0,8338) и истинным (точно не известным) значением этой вероятности составляет не более 5% от максимально возможного значения такой разности.

**Определение точности оценок при заданном количестве испытаний.** Если задано количество испытаний имитационной модели ( $n$ ), то можно определить относительную точность оценок, полученных по результатам моделирования. Из формул (3.18) и (3.19) видно, что относительную точность оценок можно определить следующим образом:

- для среднего значения:

$$\varepsilon_{\beta 0} = \frac{1}{\sqrt{n}}; \quad (3.26)$$

- для вероятности:

$$\varepsilon_{\beta 0} = \sqrt{\frac{2}{n}}. \quad (3.27)$$

**Пример 3.12.** Найти относительную точность оценок среднего времени безотказной работы прибора и вероятности его безотказной работы в течение гарантийного срока, полученных в примере 3.10.

Оценки (среднее время безотказной работы прибора - 4,17 года, вероятность безотказной работы в течение гарантийного срока - 0,97) были получены по результатам 100 000 испытаний. Относительную точность оценки среднего времени безотказной работы найдем по формуле (3.26):

$\varepsilon_{\beta 0} = \sqrt{1/100000} = 0,0032$ , или 0,32%. Относительная точность оценки вероятности безотказной работы в течение гарантийного срока определяется по формуле (3.27):  $\varepsilon_{\beta 0} = \sqrt{2/100000} = 0,0045$ , или 0,45%. Это означает, что с вероятностью 0,995 разность между оценкой среднего времени безотказной работы, полученной по результатам моделирования (4,17 года), и истинным (точно не известным) значением среднего времени безотказной работы составляет не более 0,32% от максимально возможного значения такой разности. Разность между оценкой вероятности безотказной работы, полученной по результатам моделирования (0,9754), и истинным (точно не известным) значением вероятности безотказной работы с вероятностью 0,995 составляет не более 0,45% от максимально возможного значения такой разности.

### 3.5. Средства имитационного моделирования в Excel

Для получения СРРЧ в табличном процессоре Excel используется функция СЛЧИС (из категории Математические).

Кроме того, в Excel имеются стандартные средства для имитации некоторых случайных величин, часто применяемых в практических задачах моделирования. Имеется возможность розыгрыша чисел, распределенных по гауссовскому (нормальному), равномерному, пуассоновскому, биномиальному и некоторым другим законам. Для этого используется инструмент Генерация случайных чисел (меню Сервис – Анализ данных).

**Пример 3.13.** Решить в Excel задачу из примера 3.8. Выполнить 100 испытаний.

Задача решается в следующем порядке.

1. Выбрать из меню Excel команду Сервис – Анализ данных – Генерация случайных чисел.

2. В поле Число переменных ввести 1; это означает, что создаваемые случайные числа должны выводиться в один столбец. В поле Число случайных чисел ввести 100; это означает, что требуется получить 100 случайных чисел. В поле Распределение выбрать значение Нормальное. В поле Среднее ввести значение 6, в поле Стандартное отклонение – значение 0,2. Поле Случайное рассеивание можно оставить пустым (его смысл показан ниже). Установить переключатель Выходной интервал, и в соответствующем поле ввести значение A1; это означает, что заданное количество чисел (100) будет выведено в столбец, начиная с ячейки A1. Нажать ОК. В ячейки A1:A100 выводится 100 случайных чисел, распределенных по гауссовскому закону с указанными параметрами. Эти числа имитируют толщину первой пластины.

3. Аналогичным образом получить в столбце B 100 случайных чисел, имитирующих толщину второй пластины. Для этого снова выбрать команду Сервис – Анализ данных – Генерация случайных чисел. Установить параметры: Число переменных – 1, Число случайных чисел – 100, Распреде-

ление – Нормальное, Среднее - 4, Стандартное отклонение –0,1, Выходной интервал - B1. Нажать ОК.

4. В столбце 100 вычислить значения толщины готового изделия (суммы случайных чисел, полученных в столбцах А и В). Для этого в ячейке С1 ввести формулу: =A1+B1. Распространить ее на ячейки С2:С100.

5. В ячейке С101 вычислить среднюю толщину готовых изделий по результатам 100 испытаний. Для этого использовать функцию СРЗНАЧ с аргументом С1:С100.

6. В ячейке D1 ввести функцию И (из категории Логические). В поле Логическое1 ввести условие  $C1 \geq 9,7$ , в поле Логическое2 – условие  $C1 \leq 10,3$ . Нажать ОК. Распространить содержимое ячейки D1 на ячейки D2:D100. В результате в ячейках D1:D100 будут получены значения ИСТИНА, если выполняется условие  $9,7 \leq C1 \leq 10,3$  (т.е. изделие годное), или ЛОЖЬ – если это условие не выполняется (изделие бракованное).

7. В ячейке D101 подсчитать количество годных изделий. Для этого ввести функцию СЧЕТЕСЛИ (из категории Статистические) со следующими аргументами: Диапазон – D1:D100, Условие – Истина. Подсчитывается количество случаев, когда в ячейках D1:D100 имеется значение ИСТИНА, т.е. количество смоделированных годных изделий.

8. В ячейке D102 вычислить вероятность выпуска годного изделия. Для этого разделить количество годных изделий на количество испытаний: =D101/100.

Результаты, полученные в Excel, могут несколько отличаться от полученных в примере 3.8. Точность результатов, полученных в Excel, ниже, так как они получены по результатам меньшего количества испытаний. Однако моделирование в Excel можно считать приемлемым способом получения приближенных оценок для несложных задач.

Примечание. Поле Случайное рассеивание можно использовать для *неоднократного* получения *одного и того же* набора случайных чисел. Для этого в поле Случайное рассеивание необходимо указать некоторое произвольное число. Если затем снова выполнить генерацию случайных чисел, указав *то же* число в поле Случайное рассеивание, то будет получен такой же набор случайных чисел. Конечно, для этого все остальные параметры моделирования (Число случайных чисел, Распределение и т.д.) также должны быть прежними.

**Пример 3.14.** Решить в Excel задачу из примера 3.10. Выполнить 100 испытаний.

В рассматриваемом примере время безотказной работы блоков, входящих в состав прибора, представляет собой случайную величину, распределенную по экспоненциальному закону. В Excel нет стандартных средств для имитации таких величин. Поэтому для розыгрыша случайных чисел, имитирующих время работы блоков, потребуется воспользоваться формулой (3.10). Для розыгрыша СРРЧ будет использоваться функция СЛЧИС.

Задача решается в следующем порядке.

1. Ввести в ячейку A1 формулу  $=2*LN(СЛЧИС())$ . Таким образом выполняется розыгрыш случайного числа, распределенного по экспоненциальному закону, со средним значением  $\bar{X}=2$ , т.е. имитация времени безотказной работы *одного блока*.

2. Распространить формулу, введенную в ячейку A1, на диапазон ячеек A1:D100. Таким образом имитируется время безотказной работы 400 блоков, входящих в 100 приборов. Четыре случайных числа, разыгрываемых в каждой строке, имитируют время безотказной работы четырех блоков, входящих в один прибор.

3. В столбце E найти значения времени безотказной работы каждого из приборов (максимальное из значений времени безотказной работы блоков, входящих в прибор). Для этого в ячейке E1 ввести формулу:  $=МАКС(A1:D1)$ . Распространить ее на ячейки E2:E100.

4. В ячейке E101 вычислить среднее время безотказной работы приборов по результатам 100 испытаний. Для этого использовать функцию СРЗНАЧ с аргументом E1:E100.

5. В столбце F получить значения ИСТИНА для случаев, когда смоделирована безотказная работа прибора в течение гарантийного срока, и значения ЛОЖЬ – для случаев, когда смоделирован отказ прибора. Для этого в ячейке F1 ввести функцию ЕСЛИ (из категории Логические) со следующими аргументами: Логическое выражение –  $E1 \geq 1$ , Значение если истина – ИСТИНА, Значение если ложь – ЛОЖЬ. Нажать ОК. Распространить содержимое ячейки F1 на ячейки F2:D100.

6. В ячейке F101 подсчитать количество приборов, отработавших гарантийный срок без отказа. Для этого ввести функцию СЧЕТЕСЛИ (из категории Статистические) со следующими аргументами: Диапазон – F1:F100, Условие – Истина. Подсчитывается количество случаев, когда в ячейках F1:F100 имеется значение ИСТИНА, т.е. количество случаев, когда смоделирована нормальная работа прибора в течение гарантийного срока.

7. В ячейке F102 вычислить вероятность безотказной работы прибора в течение гарантийного срока:  $=F101/100$ .