

# ОБ ИМИТАЦИОННОМ МОДЕЛИРОВАНИИ СЛУЧАЙНЫХ ВЕЛИЧИН С ПОМОЩЬЮ ИНТЕНСИВНОСТИ<sup>1</sup>

**Зверкина Г. А.**<sup>2</sup>

*(ФГБУН Институт проблем управления  
им. В.А. Трапезникова РАН, Москва)*

**Кошелев А. А.**<sup>3</sup>

*(Московский государственный университет  
им. М.В. Ломоносова, Москва)*

*Как известно, случайные величины обычно моделируются с помощью функции, обратной к функции распределения. Однако часто при моделировании поведения сложных технических моделей исследователь сталкивается с изменением параметров моделируемых случайных величин. Часто в этих случаях меняется интенсивность завершения определенного случайного периода. Следовательно, важно иметь возможность моделировать случайную величину в условиях (случайного) изменения интенсивности этой случайной величины. В данной статье предлагается новый метод моделирования случайной величины по её интенсивности. Текст снабжён иллюстративным материалом результатов численных экспериментов, подтверждающих эффективность предложенного метода. Естественно, этот текст – только начало работы. В дальнейшем будут уточнены технические детали эффективности применения предложенного метода.*

Ключевые слова: интенсивность случайной величины, численное моделирование, вопросы оптимизации моделирования.

## **1. Введение**

Как хорошо известно, значительное количество известных моделей в теории массового обслуживания, в теории надёжности и теории сетей массового обслуживания описывается линейчатыми процессами (см. [3]).

---

<sup>1</sup> Работа выполнена при частичной финансовой поддержке РФФИ, грант №20-01-00575А

Авторы признательны М.П. Фархадову за ценное обсуждение содержания статьи.

<sup>2</sup> Галина Александровна Зверкина, к.ф.-м.н., доцент (zverkina@gmail.com).

<sup>3</sup> Александр Анатольевич Кошелев, (koshelev030698@yandex.ru).

Как правило, большинство таких процессов являются эргодическими, т.е. с течением времени распределение состояния такого процесса (слабо) сходится к некоторому стационарному инвариантному (относительно соответствующих переходных вероятностей) распределению.

Обычно для стационарного распределения можно вычислить некоторые важные характеристики, однако в значительном количестве случаев теоретическое определение скорости сходимости распределения изучаемой системы к стационарному распределению оценить или вычислить невозможно.

Поэтому для получения оценок скорости сходимости распределения исследуемой системы к стационарному распределению может быть использовано имитационное моделирование поведения системы с помощью программных средств.

Напомним данное в [3] определение линейчатого процесса (с некоторой модификацией для того, чтобы кусочно-линейный процесс был марковский):

*Определение 1.* Кусочно-линейным марковским процессом называется случайный процесс  $X_t = (\nu(t), \vec{v}(t))$ , определяемый следующим образом. Пространство состояний процесса  $\mathcal{X}$  – это множество пар  $(i, \vec{\xi}_i)$ , где  $i$  – элемент конечного или счётного пространства, а  $\vec{v}_i$  – вектор  $= (v_1, \dots, v_{|i|})$ , где  $|i| \geq 0$  является «размером» базового состояния  $i$ ;  $v_j \geq 0$ . Поведение процесса  $X_t$  описывается так.

Пусть  $X_t = (i, \vec{y})$ ,  $\vec{y} = (y_1, \dots, y_{|i|})$ . Тогда с вероятностью  $\lambda_i(X_t) \times p_{ij}(X_t) dt$  за время  $dt$  произойдёт случайный переход  $X_t$  в базовое состояние  $j$ . После перехода в состояние  $j$  новое значение компоненты  $\vec{v}(t)$  случайно и определяется измеримой по  $\vec{y}$  функцией распределения (ф.р.)

$$B_{ij}^{(0)}(\vec{x}|\vec{y}) = \mathbf{P} \{ \vec{v}(t + dt) < \vec{x} | \vec{v}(t) = \vec{y}, \nu(t) = i, \nu(t + dt) = j \}.$$

Вероятность того, что за малое время  $h$  произойдёт более одного случайного перехода, есть  $o(h)$ . При этом если  $\nu(t + dt) = \nu(t)$ , то  $\vec{v}(t + dt) = \vec{v}(t) + \vec{\alpha}_i dt$ , где  $\vec{\alpha}_i = (\alpha_{i1}, \dots, \alpha_{i|i|})$  –

вектор с неотрицательными компонентами, если на интервале  $(t, t + dt)$  не было случайных переходов (см. [3]).

Таким образом, поведение кусочно-линейного процесса определяется случайными величинами (сл.в.), для которых обычно известна *интенсивность* окончания того или иного периода. Кроме того, часто бывает известна функция распределения периодов кусочно-линейного процесса – в этом случае может применяться *классический* метод моделирования случайных величин с помощью обратной функции к функции распределения.

То есть случайная величина  $\xi$  с функцией распределения  $F(x) \stackrel{\text{def}}{=} \mathbf{P} \{ \xi \leq x \}$  может быть промоделирована как

$$(1) \quad \xi \stackrel{\text{def}}{=} F^{-1}(\mathcal{U}),$$

где

$$F^{-1}(y) \stackrel{\text{def}}{=} \inf \{ x \in \mathbb{R} : F(x) \geq y \}, \quad y \in \mathbb{R},$$

а  $\mathcal{U}$  – равномерно распределённая случайная величина на интервале  $[0; 1)$ .

Такое моделирование одной системы надёжности проводилось в работе [4]; при этом было обнаружено, что начиная с некоторого момента за счет накопления ошибки точность вычисления падает и становится примерно равной около  $10^{-4}$ , что демонстрирует статья [4]. Случайные величины в работе [4] моделировались с точностью до 15 значащих цифр по схеме, описанной выше по формуле (1).

Естественно, в дальнейшем следует учитывать тот факт, что имитационное моделирование непрерывных случайных величин фактически является моделированием дискретных случайных величин с шагом, определяемым используемым программным продуктом.

В значительном количестве изучаемых моделей ТМО и смежных областей распределения работы/восстановления/интервала между заявками и пр. известны. Но не всегда даже для известных распределений можно найти явное выражение для обратной функции  $F^{-1}(\cdot)$  – например, для часто встречающегося в приложениях гамма-распределения. Также в ряде сложных сто-

хастических систем «поведение» элементов системы может существенно зависеть от общего состояния системы (т.е. состояний других элементов системы) и не может быть определено до начала моделирования поведения сложной системы. В этом случае будем рассматривать случайную величину (длину периода восстановления) как случайную величину, заданную с помощью (случайно) меняющейся интенсивности окончания того или иного периода (отказа, ремонта, появления требования в системе, обслуживания требования и др.) меняется в зависимости от полного состояния изучаемой системы (см., например, [2, 8, 9, 10, 11, 12, 13, 14]).

Поэтому целью настоящей работы является представление алгоритма моделирования случайной величины с помощью её *интенсивности*.

## 2. Случайные величины и интенсивности

Пусть при исследовании поведения некоторого случайного процесса нам необходимо учесть поведение некоторого объекта, на поведение которого может влиять состояние *всей* исследуемой системы.

Пусть в некоторый момент времени  $t$  известно, что еще не закончился период работы (ремонта, ожидания появления нового требования и пр.). Если распределение этого периода  $\xi$  имеет *непрерывную* функцию распределения  $F(s)$ , то вероятность того, что на бесконечно малом промежутке времени  $(t, t + \Delta t)$  произойдет окончание интересующего нас периода, имеет вид

$$\begin{aligned} & \mathbf{P} \{ \xi \in (t, t + \Delta t) \mid \xi > t \} = \\ & = \frac{\mathbf{P} \{ \xi < t + \Delta t \} - \mathbf{P} \{ \xi < t \}}{\mathbf{P} \{ \xi > t \}} = \frac{F(t + \Delta t) - F(t)}{1 - F(t)}. \end{aligned}$$

При  $\Delta t \rightarrow 0$  имеем

$$(2) \quad \mathbf{P} \{ \xi \in (t, t + \Delta t) \mid \xi > t \} = \lambda(t)\Delta t + o(\Delta t),$$

где

$$\lambda(t) = \frac{p(t)}{1 - F(t)} = \frac{F'(t)}{1 - F(t)} = -\frac{d}{dt} \ln(1 - F(t)),$$

$F(t)$  — функция распределения,  $p(t)$  — плотность распределения. Функция  $\lambda(t)$  называется интенсивностью случайной величины (см., например, [7]). Мы предполагаем, что интенсивность может зависеть от времени и от некоторых дополнительных параметров, связанных с поведением других элементов сложной системы. Заметим, что

$$F(s) = 1 - \exp\left(-\int_0^s \lambda(u) du\right), \quad p(s) = \lambda(s) \exp\left(-\int_0^s \lambda(u) du\right).$$

Отсюда видно, что п.в. (почти всюду) положительная случайная величина с абсолютно непрерывным распределением, являющаяся периодом некоторого процесса восстановления, определяется однозначно либо функцией распределения, либо плотностью, либо интенсивностью — подробнее см. [1, 6].

### **3. Идея предлагаемого метода моделирования случайных величин**

В данной работе предлагается для моделирования поведения сложных систем (массового обслуживания, надёжности и пр.), поведение которых определяется кусочно-линейными процессами, использовать не «классический» метод моделирования случайных величин по схеме (1), а опираясь на формулу (2).

А именно, рассматривается последовательность независимых одинаково распределённых величин  $\{\mathcal{U}_i\}$ , равномерно распределённых на  $[0; 1)$ .

Время с момента  $t = 0$  до произвольного достаточно большого времени  $T$  разбивается на интервалы точками  $0 < \tau_1 < \tau_2 < \dots < \tau_n = T$ . В рассматриваемый момент времени  $t = \tau_i < T$  предполагается, что время пребывания некоторого параметра рассматриваемой сложной системы (обслуживания, надёжности и пр.) в данном состоянии равно величине  $s \geq 0$ ; интенсивность окончания этого периода равна  $\lambda(s)$ . Тогда изменение этого состояния (т.е. окончание пребывания в данном состоянии)

на интервале времени  $(t; t + \Delta t)$  (где  $\Delta t \stackrel{\text{def}}{=} \tau_{i+1} - \tau_i$ ) близко к значению  $\lambda(s)\Delta t$ . Поэтому с вероятностью  $\mathbb{P}_s \stackrel{\text{def}}{=} \mathbf{P} \{ \lambda(s)\Delta t > \mathcal{U}_i \}$  в момент  $\tau_{i+1}$  закончится рассматриваемый период времени пребывания исследуемого параметра в данном состоянии.

Таким образом, общая схема моделирования случайной величины по её интенсивности выглядит так:

1. Выбираем точки  $0 < \tau_1 < \tau_2 < \dots < \tau_n = T$ .

2. Если выполнено неравенство  $\lambda(0)(\tau_1 - 0) > \mathcal{U}_0$ , то принимаем, что  $\xi = 0$ ; если это не так, то переходим к следующему этапу.

3. Если выполнено неравенство  $\lambda(\tau_1)(\tau_2 - \tau_1) > \mathcal{U}_1$ , то принимаем, что  $\xi = \tau_1$ ; если это не так, то переходим к следующему этапу.

4. Повторяем описанную в п.3 процедуру для  $\tau_i$  последовательно до тех пор, пока не случится событие

$$\lambda(\tau_i)(\tau_{i+1} - \tau_i) > \mathcal{U}_i.$$

Полагаем  $\xi = \tau_i$ . Напомним, что  $\mathbf{P} \{ \mathcal{U} < a \} = a$  при  $a \in [0; 1]$ .

Очевидно, что для любой сл.в.  $\xi$  с конечным математическим ожиданием эта процедура конечна с вероятностью 1. Заметим, что при

$$(3) \quad \int_0^{\infty} \lambda(s) ds = +\infty,$$

описанная процедура конечна с вероятностью 1, но применение её в случае бесконечности  $\mathbb{E} \xi$  нецелесообразно.

Данный метод удобнее классического (формула (1)) тем, что можно моделировать положительные случайные величины не зависимо от того, насколько сложно (и возможно) вычисление обратной функции к функции распределения случайной величины.

Например, для функции интенсивности  $\lambda(s) = \frac{C}{\sqrt[3]{s^3 + 1}}$  невозможно выписать функцию распределения, поскольку интеграл

$\int \frac{C}{\sqrt[3]{s^3 + 1}} ds$  не вычисляется в явном виде, но сама подын-

тегральная функция соответствует формуле (3), т.е. задаёт собственную функцию распределения сл.в.

Основной проблемой данного метода является то, что от выбор интервалов  $\Delta\tau_i \stackrel{\text{def}}{=} \tau_{i+1} - \tau_i$  в зависимости от значения величины  $\lambda(\tau_i)$  может существенно влиять на точность результатов предложенного метода моделирования случайных величин. Авторы предполагают в дальнейшем более детально исследовать этот вопрос.

Также важно и качество используемого в программе генератора случайных чисел.

#### 4. Результаты моделирования

Ниже представлены результаты моделирования случайных величин по интенсивности как в случае, когда она имеет экспоненциальный момент, так и в случае, когда случайная величина имеет лишь конечное число степенных моментов.

Для этого рассматривались случайные величины с постоянной интенсивностью  $\lambda_1(s) \equiv \lambda_0 = \text{const}$ , что соответствует экспоненциальному распределению  $F_1(s) = 1 - \exp(-\lambda_0 s)$ . Также рассматривалась переменная интенсивность  $\lambda_2(s) = \frac{C}{1+s}$ , которая соответствует функции распределения  $F_2(s) = 1 - \frac{1}{(1+s)^C}$ , которую далее мы будем называть полиномиальным распределением.

Предложенный выше метод моделирования случайных величин был реализован на языке программирования C в программе *mod.c* (см. Приложение). С помощью стандартного генератора псевдослучайных целых неотрицательных чисел `rand()` был сконструирован генератор случайных чисел с равномерным распределением:

```
1 #define frand() ((double) rand() / (RAND_MAX + 1.0))
```

Для того чтобы функция `frand()` при каждом запуске не давала одинаковые последовательности случайных чисел, она

была дополнена функцией `rand` с аргументом, равным значению функции `time(NULL)`, которая возвращает текущее время в секундах с 1 января 1970 года [5].

Далее была реализована функция `modulus`, моделирующая случайную величину по методу, описанному выше, с использованием интенсивности. Чтобы убедиться в корректности работы программы, а также подобрать оптимальный шаг  $\Delta$ , были рассмотрены и смоделированы две случайные величины с функциями распределения  $F_1(s)$  и  $F_2(s)$  с различными параметрами. В качестве критерия достоверности результатов моделирования сравнивались выборочные и теоретически вычисленные моменты случайных величин, а также гистограммы полученных выборок промоделированных случайных величин и плотностей соответствующих распределений.

Далее обозначаем  $\xi_1$  — случайная величина, распределённая по показательному закону  $F_1(s)$ , а  $\xi_2$  — по полиномиальному закону  $F_2(s)$ .

Приведем основные вероятностные характеристики этих случайных величин: функцию распределения  $F(s)$ , плотность распределения  $p(s)$ , интенсивность  $\lambda(s)$ , математическое ожидание  $\mathbb{E}\xi$ , дисперсию  $\mathbb{D}\xi$ . Для показательного распределения:

$$F_1(s) = 1 - e^{-\lambda_0 s}, \quad p_1(s) = \begin{cases} \lambda_0 e^{-\lambda_0 s}, & s \geq 0 \\ 0, & s < 0, \end{cases} \quad \lambda_1(s) = \lambda_0,$$

$$\mathbb{E}\xi_1 = \frac{1}{\lambda_0}, \quad \mathbb{D}\xi_1 = \frac{1}{\lambda_0^2},$$

где  $\lambda_0 = \text{const} > 0$ , а для полиномиального

$$F_2(s) = 1 - \frac{1}{(1+s)^C}, \quad \lambda_2(s) = \frac{C}{1+s},$$

$$p_2(s) = \begin{cases} \frac{C}{(1+s)^{C+1}}, & s \geq 0 \\ 0, & s < 0; \end{cases}$$



$$\mathbb{E}\xi_2 = \frac{1}{C-1}, \quad \mathbb{D}\xi_2 = \frac{2}{C^2-3C+2} - \frac{1}{(C-1)^2},$$

где  $C = \text{const} > 2$ .

## 5. Приложения

### 5.1. Приложение 1

По выборке, полученной моделированием для каждого из распределений, были подсчитаны математические ожидания и дисперсии для различных шагов  $\Delta$ . Выборки состояли из 100000 сгенерированных чисел. Далее приведем результаты вычислений. В качестве параметров распределений были взяты  $\lambda_0 = 2, C = 8$ .

$$\lambda_0 = 2, C = 8, \Delta = 0,1$$

Эксперимент N	$\mathbb{E}\xi_1$	$\mathbb{D}\xi_1$	$\mathbb{E}\xi_2$	$\mathbb{D}\xi_2$
1	0,401235	0,200112	0,028346	0,004858
2	0,399677	0,200059	0,028995	0,004950
3	0,400694	0,201088	0,028509	0,004851

$$\lambda_0 = 2, C = 8, \Delta = 0,01$$

Эксперимент N	$\mathbb{E}\xi_1$	$\mathbb{D}\xi_1$	$\mathbb{E}\xi_2$	$\mathbb{D}\xi_2$
1	0,487648	0,243396	0,132102	0,024996
2	0,488431	0,240377	0,130432	0,024451
3	0,487839	0,244927	0,132280	0,025362

$$\lambda_0 = 2, C = 8, \Delta = 0,001$$

Эксперимент N	$\mathbb{E}\xi_1$	$\mathbb{D}\xi_1$	$\mathbb{E}\xi_2$	$\mathbb{D}\xi_2$
1	0,496119	0,245180	0,141784	0,027176
2	0,498286	0,249556	0,141512	0,027039
3	0,495537	0,245814	0,141822	0,027003

$$\lambda_0 = 2, C = 8, \Delta = 0,0001$$

Эксперимент N	$\mathbb{E}\xi_1$	$\mathbb{D}\xi_1$	$\mathbb{E}\xi_2$	$\mathbb{D}\xi_2$
1	0,466462	0,219541	0,139288	0,025180
2	0,468898	0,218354	0,138768	0,025246
3	0,468516	0,218312	0,139447	0,025323

$$\lambda_0 = 2, C = 8, \Delta = 0,00001$$

Эксперимент N	$\mathbb{E}\xi_1$	$\mathbb{D}\xi_1$	$\mathbb{E}\xi_2$	$\mathbb{D}\xi_2$
1	0,327335	0,107316	0,112233	0,014785
2	0,327366	0,107663	0,112095	0,014777
3	0,328139	0,107909	0,112430	0,014857

Из эксперимента видно, что для выбранных значений  $\lambda_0 = 2$  и  $C = 8$  оптимальный шаг для моделирования случайной величины  $\Delta = 0,001$ .

В результате моделирования было замечено, что величина шага  $\Delta$  может существенно влиять на результат численного эксперимента.

## 5.2. Приложение 2.

Ниже приводятся гистограммы промоделированных случайных величин с указанным в заголовке распределением (голубые столбики) в сравнении со значениями плотности распределения (100000 реализаций, шаг моделирования  $\Delta = 0,001$ ).



Рис. 1. Гистограмма промоделированных случайных величин с распределением  $F_2$  (голубые столбики) в сравнении со значениями плотности распределения (100000 реализаций)

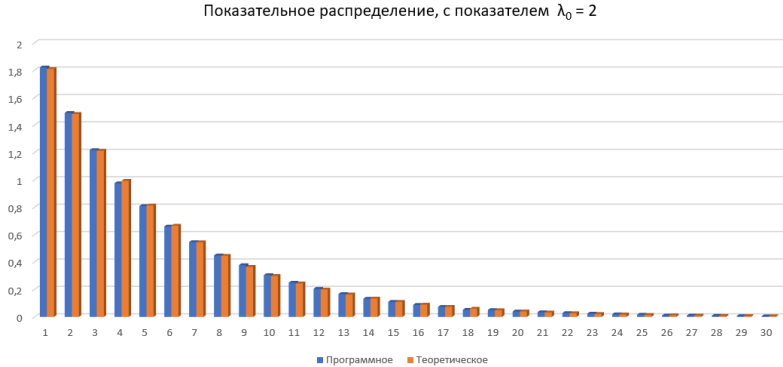


Рис. 2. Гистограмма промоделированных случайных величин с распределением  $F_1$  (голубые столбики) в сравнении со значениями плотности распределения (100000 реализаций)

### 5.3. Приложение 3

Текст программы *mod.c*

```

1 #include <stdio.h>
2 #include <stdlib.h>
3 #include <math.h>
4 #include <time.h>
5
6 #define lambda0 2.0
7 #define C 8.0
8 #define delta1 0.001
9 #define delta2 0.001
10 #define frand() ((double) rand() / (RAND_MAX + 1.0))
11
12 void sort(double *p0, int n);
13 double lambda(double t, int num);
14 void modulus(double *mass, double *mE, double *disp,
15             double delta, int num);
16 void nuller(double *mass, int n);
17
18 void sort(double *p0, int n)
19 {
20     double first;
21     int j, i;
22     for(i = 0; i < n-1; i++)
23         for(j = i+1; j < n; j++)

```

```
23     if(p0[i] > p0[j])
24     {
25         first = p0[i];
26         p0[i] = p0[j];
27         p0[j] = first;
28     }
29 }
30
31 double lambda(double t, int num)
32 {
33     double l;
34     if(num == 1)
35     {
36         l = lambda0; //1 --- Exponential distribution
37     }
38     if(num == 2)
39     {
40         l = C / (1 + t); //2 --- Polynomial distribution
41     }
42     return l;
43 }
44
45 void modulus(double *mass, double *mE, double *disp,
46             double delta, int num)
47 {
48     int i;
49     double randnum, t = 0, tNew = 0, sumt = 0, tsq = 0;
50     for(i = 0; i < 100000; i++)
51     {
52         while(1)
53         {
54             randnum = frand();
55             if(delta * lambda(t, num) > randnum)
56             {
57                 tNew = t;
58                 mass[i] = t;
59                 break;
60             }
61             t += delta;
62         }
63         sumt += tNew;
64         tsq += t*t;
65         t = 0;
```

```

65     }
66     *mE = sumt / 100000;
67     *disp = (tsq / 100000 - *mE * *mE) * 100000 / 99999;
68     return;
69 }
70
71 void nuller(double *mass, int n)
72 {
73     int i;
74     for(i = 0; i < n; i++)
75     {
76         mass[i] = 0;
77     }
78 }
79
80 int main(void)
81 {
82     double rv1[100000], rv2[100000], mexp[30], mpol[16];
83     double x, mE = 0, disp = 0, dt1 = 0.1, left1 = 0,
84     sum = 0;
85     int i, j;
86     FILE *f1, *f2;
87     f1 = fopen("out1.txt", "w");
88     f2 = fopen("out2.txt", "w");
89     srand(time(NULL));
90     modulus(rv1, &mE, &disp, delta1, 1);
91     printf("Exponential distribution\n delta = %lf\n M[
92     exp] = %lf\t M[exp_theor] = %lf\n D[exp] = %lf\t D[
93     exp_theor] = %lf\n", delta1, mE, 1.0 / lambda0, disp,
94     1.0 / pow(lambda0, 2));
95     modulus(rv2, &mE, &disp, delta2, 2);
96     printf("Polynomial distribution\n delta = %lf\n M[
97     pol] = %lf\t M[pol_theor] = %lf\n D[pol] = %lf\t D[
98     pol_theor] = %lf\n", delta2, mE, 1.0 / (C - 1), disp,
99     2.0 / (C*C - 3 * C + 2) - 1.0 / pow(C - 1, 2));
100
101     sort(rv1, 100000);
102     sort(rv2, 100000);
103     nuller(mexp, 30);
104     nuller(mpol, 16);
105     j = 0;
106     left1 = dt1;
107     for(i = 0; i < 100000 && j < 30; i++)

```

```
101     {
102         if(rv1[i] > left1)
103         {
104             left1 += dt1;
105             j++;
106         }
107         mexp[j] += 1;
108     }
109     j = 0;
110     left1 = dt1;
111     for(i = 0; i < 100000 && j < 16; i++)
112     {
113         if(rv2[i] > left1)
114         {
115             left1 += dt1;
116             j++;
117         }
118         mpol[j] += 1;
119     }
120
121     left1 = 0.05;
122     for(j = 0; j < 30; j++)
123     {
124         fprintf(f1, "%lf\t %lf\t %lf\t %lf\n", left1, (
125         mexp[j] / 100000) / dt1, lambda0 * exp(-lambda0 * (
126         left1)), 1 - exp(-lambda0 * (left1)));
127         left1 += dt1;
128     }
129
130     left1 = 0.05;
131     for(j = 0; j < 16; j++)
132     {
133         fprintf(f2, "%lf\t %lf\t %lf\t %lf\n", left1, (
134         mpol[j] / 100000) / dt1, C / pow(1 + left1, C + 1), 1
135         - 1 / pow(1 + left1, C));
136         left1 += dt1;
137     }
138
139     fclose(f1);
140     fclose(f2);
141     return 0;
142 }
```

## Литература

1. БОРОВКОВ А.А. *Вероятностные процессы в теории массового обслуживания*. – М.: Физматлит, 1972. – 368 с.
2. ВЕРЕТЕННИКОВ А.Ю. *О скорости сходимости к стационарному распределению в системах обслуживания с одним прибором* // Автоматика и телемеханика. – 2013. – №10. – С. 23–35. Engl. transl.: *On the rate of convergence to the stationary distribution in the single-server queuing system* // Automation and Remote Control. – 2013. – Vol. 74, Iss. 10. – P. 1620–1629.
3. ГНЕДЕНКО Б.В., КОВАЛЕНКО И.Н. *Введение в теорию массового обслуживания*. – М.: Наука, 1966.
4. ЗВЕРКИНА Г.А. *Об экспоненциальной сходимости коэффициента готовности* // Управление большими системами. – 2021. – Вып. 90. – С. 5–35.
5. КЕРНИГАН Б.У., РИЧЧИ Д.М. *Язык программирования С*. 2-е изд. – М.: ООО «И.Д. Вильямс», 2015. – 288 с.
6. КОКС Д., СМИТ В. *Теория восстановления*. – М.: Советское радио, 1967.
7. ТАРТАКОВСКИЙ А.Г. *О последовательном оценивании интенсивности процессов восстановления* // Проблемы передачи информации. – 1985. – №21:1. – С. 40–47; *Problems Information Transmission*. – 1985. – Vol. 21:1. – P. 30–36.
8. VERETENNIKOV A. *On convergence rate for Erlang–Sevastyanov type models with infinitely many servers* // *Theory of Stochastic Processes*. – 2017. – No. 1. – P. 88–102.
9. VERETENNIKOV A. *On mean-field GI/GI/1 queueing model: existence and uniqueness* // *Proc. of the Int. Conf. "Analytical and Computational Methods in Probability Theory and Its Applications" (ACMPT-2017), 23–27 October 2017 / Ed.: A.V. Lebedev. – RUDN, Moscow, Russia. – P. 182–186.*
10. VERETENNIKOV A. *On mean-field (GI/GI/1) queueing model: existence and uniqueness* // *Queueing Systems*. – 2020. – Vol. 94(3). – P. 243–255.

11. VERETENNIKOV A. *On Polynomial Recurrence for Reliability System with a Warm Reserve* // Markov Processes and Related Fields. – 2019. – Vol. 25. – P. 745–761.
12. VERETENNIKOV A. *On Recurrence and Availability Factor for Single-Server System With General Arrivals* // Reliability: Theory and Applications (RT& A). – 2016. – Vol. 11, No. 3(42). – P. 49–58.
13. VERETENNIKOV A.YU. *On the rate of convergence for infinite server Erlang-Sevastyanov's problem* // Queueing Systems. – 2014. – Vol. 76, Iss. 2. – P. 181–203.
14. VERETENNIKOV A.YU., ZVERKINA G.A. *Simple Proof of Dynkin's Formula for Single-Server Systems and Polynomial Convergence Rates* // Markov Processes and Related Fields. – 2014. – Vol. 20. - P. 479–504.

## **ON SIMULATION OF RANDOM VARIABLES USING INTENSITY**

**Galina Zverkina**, V.A. Trapeznikov Institute of Control Sciences of RAS, Moscow, Cand.Sc., assistant professor (zverkina@gmail.com).

**Alexandr Koshelev**, Moscow State University, Moscow, student (koshelev030698@yandex.ru).

*Abstract: As is well known, random variables are usually modelled using the inverse function of the distribution function. However, often when modelling the behaviour of complex technical models, a researcher is faced with a change in the parameters of the simulated random variables. Often in these cases, the intensity of the completion of a certain random period changes (or “failure rate”, or “hazard rate”). Therefore, it is important to be able to simulate a random variable under conditions of a (random) change in the intensity of this random variable. This paper proposes a new method for modelling a random variable by its intensity. The text is provided with illustrative material of the results of numerical experiments confirming the efficiency of the proposed method. Naturally, this text is just the beginning of work. In the future, the technical details of the effectiveness of the application of the proposed method will be clarified.*

**Keywords:** random variable intensity, numerical simulation, discussion of simulation optimization.



УДК 519.2, 519.6

ББК 22.171, 22.176, 22.19

DOI: 10.25728/ubs.2021.94.2

*Статья представлена к публикации  
членом редакционной коллегии Я.И. Квинто.*

*Поступила в редакцию 15.11.2021.*

*Дата опубликования 30.11.2021.*