**Лабораторная часть**

Рассмотрим *моделирование* двухфазной системы массового обслуживания в программных средах MATLAB и *GPSS*/PC. Сначала выполним *моделирование* двухфазной системы в MATLAB. Примем, что *параметр* входного простейшего потока равен 1.7, а интенсивность обслуживания каждой фазы равна 0.67, т. е. \lambda = 1.7, \mu = 0.67. В основе моделирования стоит задача интегрирования системы дифференциальных уравнений вида (1.2) на отрезке времени, когда вероятности состояний примут свои установившиеся значения. *По* установившимся значениям вероятностей рассчитываются операционные характеристики системы. Входными данными программы, таким образом, будут являться интенсивности входного потока и обслуживания, а также *матрица* *коэффициентов системы* дифференциальных уравнений.

**1. Пример моделирования в системе matlab**

Программа моделирования оформлена в виде М -функции без аргументов и без возвращаемого значения. Это позволит в одном М -файле использовать дополнительные М -функции.

Программный код решения примера:

function twophase;

clc,close all

% ВХОДНЫЕ ДАННЫЕ

L = 1.7;

M = 0.67;

global A

A = [-L, M, 0, 0, 0;

0, -(L+M), M, 0, M;

L, 0, -M, M, 0;

0, L, 0, -2\*M, 0;

0, 0, 0, M, -M];

% ИНТЕРВАЛ ИНТЕГРИРОВАНИЯ

T = [0, 20];

% НАЧАЛЬНЫЕ УСЛОВИЯ

P0 = [1;zeros(length(A)-1,1)];

% ВЫЗОВ РЕШАТЕЛЯ ДИФФЕРЕНЦИАЛЬНЫХ УРАВНЕНИЙ

[t,P] = ode45(@faza,T,P0);

% ПОСТРОЕНИЕ ДИАГРАММ ВЕРОЯТНОСТЕЙ СОСТОЯНИЙ

figure(1);

line(t,P(:,1),'linew',2, 'color','r'),

line(t,P(:,2), 'marker','o','color', [0,102,102]/255),

line(t,P(:,3), 'lines','--','linew',2, 'color', 'k'),

line(t,P(:,4), 'lines','-.','linew',2, 'color', 'b'),

line(t,P(:,5), 'linestyle',':','linew',2, 'color','m'),

str='\bfВероятности состояний двухфазной системы обслуживания:';

title(sprintf('%s %s = %g; %s = %g',str,'\lambda',L,'\mu', M));

legend('P\_0\_0(t)','P\_0\_1(t)','P\_1\_0(t)','P\_1\_1(t)','P\_b\_1(t)');

xlabel('\bf - - - - - - t - - - - - - ');

ylabel('\bf P(t)');

%%% ОПЕРАЦИОННЫЕ ХАРАКТЕРИСТИКИ СИСТЕМЫ

% Ncp - СРЕДНЕЕ ЧИСЛО ТРЕБОВАНИЙ В СИСТЕМЕ

k =[0, 1, 2];

Pk = [P(end,1);P(end,2)+P(end,3);P(end,4)+P(end,5)];

Ncp = k\*Pk;

fprintf('\n\t ОПЕРАЦИОННЫЕ ХАРАКТЕРИСТИКИ ДВУХФАЗНОЙ СИСТЕ-МЫ:\n');

fprintf('\t Среднее число требований в системе: Ncp = %f\n',Ncp);

% Pf1 - ВЕРОЯТНОСТЬ НАЧАЛА ОБСЛУЖИВАНИЯ

Pf1 = P(end,1)+P(end,2);

fprintf('\t Вероятность начала обслуживания: Pf1 = %f\n',Pf1);

% Pomk - ВЕРОЯТНОСТЬ ОТКАЗА В ОБСЛУЖИВАНИИ

Pomk = 1-Pf1;

fprintf('\t Вероятность отказа: Pomk = %f\n',Pomk);

% Q - ОТНОСИТЕЛЬНАЯ ПРОПУСКНАЯ СПОСОБНОСТЬ

Q = 1-Pomk;

fprintf('\t Относительная пропускная способность системы: Q = %f\n',Q);

% Lef - ЭФФЕКТИВНАЯ ЧАСТОТА ПОСТУПЛЕНИЯ ТРЕБОВАНИЙ В СИСТЕМУ

Lef = L\*Pf1;

fprintf('\t Эффективная частота поступления требований в систему: Lef = %f\n',Lef);

% Ab - АБСОЛЮТНАЯ ПРОПУСКНАЯ СПОСОБНОСТЬ

Ab = Lef\*Q;

fprintf('\t Абсолютная пропускная способность системы: Ab = %f\n',Ab)

% Ts - ПОЛНОЕ ВРЕМЯ ПРЕБЫВАНИЯ ТРЕБОВАНИЙ В СИСТЕМЕ

Ts = Ncp/Lef;

fprintf('\t Полное время пребывания требования в системе: Ts = %f\n',Ts);

% Tcp1 - СРЕДНЕЕ ВРЕМЯ ОБСЛУЖИВАНИЯ ОДНОГО ТРЕБОВАНИЯ

Tcp1 = 2\*(1/M);

fprintf('\t Среднее время обслуживания требования в системе: Tcp = %f\n',Tcp1);

% Td - СРЕДНЕЕ ВРЕМЯ ЗАДЕРЖКИ

Td = Ts - Tcp1;

fprintf('\t Среднее время задержки в обслуживании: Td = %f\n',Td);

% Финальные (стационарные) вероятности

fprintf('\n\t Финальные (стационарные) вероятности:\n');

k = 0;

for J = 1 : length(P0)

if J == 1 | J == 2

fprintf('\t P0%d = %f\n ',J-1, P(end, J));

elseif J > 2 & J < length(P0)

fprintf('\t P1%d = %f\n ', k, P(end, J));

k = k + 1;

else

fprintf('\t Pb1 = %f\n', P(end,end));

end

end

% М-функция описания правых частей дифференциальных уравнений

function f = faza(t,P)

global A

f = A\*P;

Результаты моделирования по определению операционных характеристик

ОПЕРАЦИОННЫЕ ХАРАКТЕРИСТИКИ ДВУХФАЗНОЙ СИСТЕМЫ

Среднее число требований в системе: Ncp = 1.345670

Вероятность начала обслуживания: Pf1 = 0.224859

Вероятность отказа: Pomk = 0.775141

Относительная пропускная способность системы: Q = 0.224859

Эффективная частота поступления требований в систему: Lef = 0.382260

Абсолютная пропускная способность системы: Ab = 0.085955

Полное время пребывания требования в системе: Ts = 3.520298

Среднее время обслуживания требования в системе: Tcp = 2.985075

Среднее время задержки в обслуживании: Td = 0.535224

Финальные (стационарные) вероятности:

P00 = 0.063570

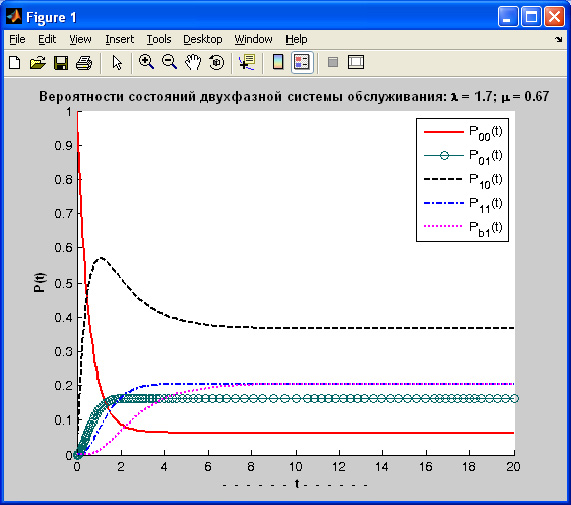
P01 = 0.161289

P10 = 0.365900

P11 = 0.204626

Pb1 = 0.204615

Переходные процессы вероятностей состояний показаны на [рис. 1.3](https://www.intuit.ru/studies/professional_skill_improvements/1602/courses/479/lecture/21080?page=2#image.1.3).



**Рис. 1.3.**Переходные процессы вероятностей состояний в двухфазной системе

*Задание 1*

1. Вычислите по формуле Литтла среднее количество требований в системе.
2. Проверьте нормировочное условие для вероятностей состояний системы в трех точках временного интервала интегрирования.
3. В соответствии с номером компьютера, на котором выполняется лабораторная работа (1, 2, 3 и т. д.), увеличьте интенсивность входного потока на этот номер. Выполните два предыдущих пункта задания.
4. Результаты расчета операционных характеристик запишите в текстовый файл с именем compX.txt, где Х — номер компьютера, на котором выполняется лабораторная работа.
5. Напишите программу с различными интенсивностями обслуживания в фазах обслуживания. Рассчитайте также операционные характеристики двухфазной системы обслуживания.

**2. Пример моделирования в системе GPSS/PC**

Для моделирования в системе *GPSS*/PC необходимо подготовить данные по функциям распределения случайных величин для имитации пуассоновского входного потока и экспоненциального обслуживания. В обоих случаях используется экспоненциальное распределение, поскольку интервалы времени между требованиями в пуассоновском потоке распределены по экспоненциальному закону. В рассматриваемом случае параметры пуассоновского потока и экспоненциального обслуживания заданы и равны соответственно \lambda = 1.7, \mu = 0.67.

Для подготовки данных по функциям распределения случайных величин использован следующий программный код:

clear all,clc

L = 1.7; %% Интенсивность входного потока

M = 0.67; %% Интенсивность обслуживания

%% Функции экспоненциального распределения

x = 0 : 0.2 : 20;

F = 1 - exp(-L\*x);

F2 = 1 - exp(-M\*x);

%%% Запись в текстовый файл input.txt

fid = fopen('input.txt', 'w');

%% Для блока generate

fprintf(fid, 'puas function RN1,C50\r\n');

for J = 1 : length(x)

if J <= 50

if mod(J, 5)

fprintf(fid, '%g,%g/', F(J), x(J));

else

fprintf(fid, '\r\n');

fprintf(fid, '%g,%g/', F(J), x(J));

end

end

end

fprintf(fid, '\r\n;-----------------------------------\r\n');

%% Для блоков advance

fprintf(fid, '\r\nexpM function RN2,C50\r\n');

for J = 1 : length(x)

if J <= 50

if mod(J, 5)

fprintf(fid, '%g,%g/', F2(J), x(J));

else

fprintf(fid, '\r\n');

fprintf(fid, '%g,%g/', F2(J), x(J));

end

end

end

fclose(fid);

В программе функция mod() применена для формирования заданного количества массива строк с данными для выбора случайных чисел, распределенных по экспоненциальному закону. Сформированные массивы будут представлены в *GPSS*-программе.

*GPSS*-программа приводится ниже. В ней задано условие обработки 500 требований.

Программный код решения примера в системе *GPSS*/PC:

simulate

puas function RN5,C50

0,0/0.28823,0.2/0.493383,0.4/0.639405,0.6/

0.743339,0.8/0.817316,1/0.869971,1.2/0.907449,1.4/0.934125,1.6/

0.953112,1.8/0.966627,2/0.976246,2.2/0.983093,2.4/0.987966,2.6/

0.991434,2.8/0.993903,3/0.995661,3.2/0.996911,3.4/0.997802,3.6/

0.998435,3.8/0.998886,4/0.999207,4.2/0.999436,4.4/0.999598,4.6/

0.999714,4.8/0.999797,5/0.999855,5.2/0.999897,5.4/0.999927,5.6/

0.999948,5.8/0.999963,6/0.999974,6.2/0.999981,6.4/0.999987,6.6/

0.99999,6.8/0.999993,7/0.999995,7.2/0.999997,7.4/0.999998,7.6/

0.999998,7.8/0.999999,8/0.999999,8.2/0.999999,8.4/1,8.6/

1,8.8/1,9/1,9.2/1,9.4/1,9.6/

1,9.8/

;------------------------------

expM function RN6,C50

0,0/0.12541,0.2/0.235092,0.4/0.331019,0.6/

0.414916,0.8/0.488291,1/0.552465,1.2/0.60859,1.4/0.657677,1.6/

0.700608,1.8/0.738154,2/0.770992,2.2/0.799712,2.4/0.82483,2.6/

0.846798,2.8/0.866011,3/0.882815,3.2/0.897511,3.4/0.910364,3.6/

0.921605,3.8/0.931437,4/0.940035,4.2/0.947556,4.4/0.954133,4.6/

0.959885,4.8/0.964916,5/0.969316,5.2/0.973164,5.4/0.976529,5.6/

0.979473,5.8/0.982047,6/0.984299,6.2/0.986268,6.4/0.98799,6.6/

0.989496,6.8/0.990813,7/0.991965,7.2/0.992973,7.4/0.993854,7.6/

0.994625,7.8/0.995299,8/0.995889,8.2/0.996404,8.4/0.996855,8.6/

0.99725,8.8/0.997595,9/0.997896,9.2/0.99816,9.4/0.998391,9.6/

0.998593,9.8/

tab1 table m1,0,5,12

tab2 table mp2,0,5,10

\*\*\*\*\*\*\*\* Basic Program \*\*\*\*\*\*\*

5 generate fn$puas

10 savevalue FULL+,1

;--- Probability P00 -----------

15 test E f1,0,met1

20 test E f2,0,met1

25 savevalue P\_00+,1

;--- Probability P01 -----------

30 met1 test E f1,0,met2

35 savevalue P\_01+,1

;--- Probability P10 -----------

40 met2 test E f2,0,met3;

45 savevalue P\_10+,1

;--- Probability P11+Pb1---------

50 met3 test NE f1,0,met4

55 test NE f2,0,met4

60 test NE q2,0,met4

65 savevalue P\_11b1+,1

70 met4 assign 2,1

75 gate NU 1,exitNOT

;---------- 1st phase ---------------

80 seize 1

85 advance fn$expM

90 release 1

95 mark 2

;---------- Delay ------------------

100 queue 2

105 gate NU 2

110 depart 2

120 tabulate tab2

;---------- 2nd phase ---------------

125 seize 2

130 advance fn$expM

135 release 2

;------------------------------------

140 tabulate tab1

145 terminate 1

150 exitNOT savevalue exit0+,1

155 terminate

start 500

;end

Комментарии к программе

* Введение таблицы tab1 с аргументом М1 (системный стандартный числовой атрибут) — временем пребывания в модели транзакта, обрабатываемого программой в данный момент, позволяет табулировать величину М1 и найти среднее значение времени пребывания транзакта в модели.
* Введение таблицы tab2 с аргументом mp2 (стандартный числовой атрибут транзакта) — значением времени, равного разности относительного модельного времени и содержимого 2-го параметра текущего транзакта. Табулируется время пребывания транзакта в заблокированном состоянии. Отметка времени осуществляется блоком mark.
* В программе использованы стандартные числовые атрибуты f1, f2, которые определяют состояния соответствующих устройств. Если устройство свободно, то величина стандартного числового атрибута равна 0, и 1 — во всех остальных случаях.

Результат выполнения программы система *GPSS*/PC оформляет в виде файла стандартного отчета, который имеет следующий вид:

GPSS/PC Report file REPORT.GPS. (V 2, # 37349) 01-08-2010 10:28:19 page 1

START\_TIME END\_TIME BLOCKS FACILITIES STORAGES FREE\_MEMORY

0 675 30 2 0 135296

LINE LOC BLOCK\_TYPE ENTRY\_COUNT CURRENT\_COUNT RETRY

5 1 GENERATE 3092 0 0

10 2 SAVEVALUE 3092 0 0

15 3 TEST 3092 0 0

20 4 TEST 501 0 0

25 5 SAVEVALUE 159 0 0

30 MET1 TEST 3092 0 0

35 7 SAVEVALUE 501 0 0

40 MET2 TEST 3092 0 0

45 9 SAVEVALUE 982 0 0

50 MET3 TEST 3092 0 0

55 11 TEST 2591 0 0

60 12 TEST 1768 0 0

65 13 SAVEVALUE 1453 0 0

70 MET4 ASSIGN 3092 0 0

75 15 GATE 3092 0 0

80 16 SEIZE 501 0 0

85 17 ADVANCE 501 1 0

90 18 RELEASE 500 0 0

95 19 MARK 500 0 0

100 20 QUEUE 500 0 0

105 21 GATE 500 0 0

110 22 DEPART 500 0 0

120 23 TABULATE 500 0 0

125 24 SEIZE 500 0 0

130 25 ADVANCE 500 0 0

135 26 RELEASE 500 0 0

140 27 TABULATE 500 0 0

145 28 TERMINATE 500 0 0

150 EXITNOT SAVEVALUE 2591 0 0

155 30 TERMINATE 2591 0 0

FACILITY ENTRIES UTIL. AVE.\_TIME AVAILABLE OWNER PEND INTER RETRY DELAY

1 501 0.800 1.08 1 3090 0 0 0 0

2 500 0.740 1.00 1 0 0 0 0 0

QUEUE MAX CONT. ENTRIES ENTRIES(0) AVE.CONT. AVE.TIME AVE.(-0) RETRY

2 19 0 500 207 2.95 3.98 6.79 0

TABLE MEAN STD.DEV. RETRY RANGE FREQUENCY CUM.%

TAB1 6.06 5.22 0

- 0 50 10.00

0 - 5 233 56.60

5 - 10 120 80.60

10 - 15 67 94.00

15 - 20 24 98.80

20 - 25 6 100.00

TAB2 3.98 5.03 0

- 0 207 41.40

0 - 5 137 68.80

5 - 10 97 88.20

10 - 15 40 96.20

15 - 20 18 99.80

20 - 25 1 100.00

XACT\_GROUP GROUP\_SIZE RETRY

POSITION 0 0

GPSS/PC Report file REPORT.GPS. (V 2, # 37349) 01-08-2010 10:28:19 page 2

SAVEVALUE VALUE RETRY

FULL +3092 0

P\_00 +159 0

P\_01 +501 0

P\_10 +982 0

P\_11B1 +1453 0

EXIT0 +2591 0

Расчет операционных характеристик двухфазной системы по файлу стандартного отчета будем на окончании помечать буквой g.

Расчет *вероятности отказа* в обслуживании определим как отношение числа требований, получивших отказ (число 2591 в сохраняемой ячейке EXIT0 ), к общему числу (число 3092 в сохраняемой ячейке FULL ):

Pomkg = 2591/3092 = 0.838.

*Относительная пропускная способность* и вероятность того, что вновь поступившее требование будет принято на обслуживание 1-й фазой, равны между собой и определяются как противоположная *вероятность отказа в обслуживании*, т. е.

Qg = Pf1g = 1 – Pomkg = 1 – 0.838 = 0.162.

Расчет *эффективной частоты* поступления требований в систему:

Lefg = \lambda*Pf1g = 1.7*0.162 = 0.228.

*Абсолютная пропускная способность* двухфазной системы:

Abg = Lefg*Qg = 0.228*0.162 = 0.037.

*Полное время* пребывания требования в системе определяется как среднее значение таблицы с именем tab1:

Tsg = 6.06.

*Среднее время блокировки* 1-й фазы определяется как среднее значение таблицы с именем tab2:

Tdg = 3.98.

*Среднее время обработки* одного требования в двухфазной системе можно определить двумя путями: как сумму значений полей AVE._TIME для устройств 1 и 2 ( FACILITY ) и как разность между Tsg и Tdg:

Tcpg = 1.08 + 1.00 = 2.08, (6.06 – 3.98 = 2.08).

*Среднее число требований* в двухфазной системе определим по формуле Литтла:

Ncpg = Lefg*Tsg = 0.228*6.06 = 1.382.

*Финальные (стационарные) вероятности* определим как частное от деления числа требований, отвечающим заданным условиям (успешное событие), к общему числу требований, поступивших в систему. Для этого в программе были созданы ячейки сохраняемых величин: для вероятности Р00 — ячейка Р_00, для вероятности Р01 — ячейка Р_01, для вероятности Р10 — ячейка Р_10, для суммы вероятностей Р11+Рb1 (поскольку они равны между собой) — ячейка P11b1. В соответствии с результатами, приведенными в файле стандартного отчета, получим

P00 = 159/3092 = 0.0514 ;

P01 = 501/3092 = 0.162 ;

P10 = 982/3092 = 0.318 ;

P11 + Pb1 = 1453/3092 = 0.4699.

Проведем контрольную проверку. Для этого сложим вероятности (которые являются несовместными):

0.0514 + 0.162 + 0.318 + 0.4699 = 1.0013.

Поскольку полученный результат практически равен единице, можно считать, что расчет вероятностей произведен успешно.

*Задание 2*

1. Выполните прогон программы с датчиками случайных чисел RN в соответствии с номером компьютера, за которым выполняется лабораторная работа, т. е. RN1, RN2 и т. д. Сравните полученные операционные характеристики с соответствующими характеристиками, определенными при моделировании двухфазной системы в MATLAB.
2. Видоизмените программу так, чтобы можно было построить функцию распределения времени блокировки 1-й фазы.
3. Напишите программу с различными интенсивностями обслуживания в фазах обслуживания. Рассчитайте также операционные характеристики двухфазной системы обслуживания. Сравните с операционными характеристиками, полученными при моделировании в системе MATLAB.
4. Смоделируйте в MATLAB и в *GPSS*/PC (*GPSS* World) трехфазную систему обслуживания и рассчитайте операционные характеристики по параметрам примера для двухфазной системы, т. е. \lambda = 1.7, \mu = 0.67.

**Контрольные вопросы**

1. Каким видом дифференциальных уравнений описывается многофазная система массового обслуживания?
2. Что называется интенсивностью входного потока требований?
3. Что называется интенсивностью обслуживания многофазной системы?
4. Что называется простейшим потоком требований?
5. Что называется функцией распределения случайной величины?
6. Какова функциональная связь между функцией распределения случайной непрерывной величины и ее плотностью?
7. Могут ли быть многофазные системы с ожиданием? Если могут, то как будет выглядеть схема моделирования многофазной системы с ожиданием?
8. Какие ограничения накладываются на численные значения интенсивности входного потока требований и интенсивности обслуживания в многофазной системе при ее моделировании?
9. Сколько вы знаете аналогов формулы Литтла? Приведите известные вам формулы.

**Теоретическая часть**

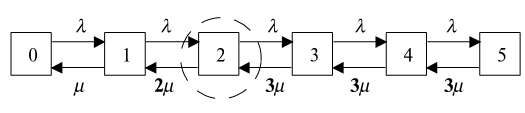
*Многоканальные системы массового обслуживания* — это системы с параллельно включенными приборами обслуживания. Для них принято использовать символику Кендалла, которая состоит из основных четырех позиций вида A/B/m/K/М, где А — закон поступления требований в систему, В — закон обслуживания требований, m — число параллельно функционирующих приборов (каналов, узлов) обслуживания, K — допустимое число требований в системе, т. е. число требований в очереди плюс число требований, принятых на обслуживание. В системе A/B/m/K/М последний символ М — число источников нагрузки [[2]](https://www.intuit.ru/studies/professional_skill_improvements/1602/courses/479/literature#literature.2). При этом параметры системы постоянны, а процессы в системах рассматриваются с позиций теории размножения и гибели.

Предметом данной лабораторной работы будут системы с простейшим пуассоновским потоком, экспоненциальным обслуживанием и, возможно, экспоненциальном уходом из очереди "нетерпеливых" требований. При сделанных условиях исследуемые системы будут иметь следующие обозначения: M/M/m/K, M/M/m/K с нетерпеливыми требованиями, M/M/m/K/M, где первые буквы М означают пуассоновский *поток требований* и экспоненциальное обслуживание, а в последнем примере крайняя правая буква М — это число требований, формируемых конечным числом источников нагрузки. В частности, может быть система с отказами, т. е. M/M/m с отказами.

При моделировании сначала требуется составить дифференциальные уравнения относительно вероятностей состояний. Эти уравнения называются дифференциальными уравнениями Колмогорова [[2]](https://www.intuit.ru/studies/professional_skill_improvements/1602/courses/479/literature#literature.2). При правильном составлении уравнений их решения при заданных начальных условиях стремятся к своим установившимся значениям. Эти значения называются стационарными вероятностями, на основе которых рассчитываются операционные характеристики системы.

Для составления уравнений Колмогорова можно использовать мнемоническое правило [[4]](https://www.intuit.ru/studies/professional_skill_improvements/1602/courses/479/literature#literature.4). Сначала определим понятие *потока вероятности*: это такой *поток*, который переводит систему из одного состояния в другое соседнее и определяется как *произведение* вероятности P_{i}(t) i -го состояния, из которого происходит переход, на интенсивность потока событий (интенсивность поступления требований или интенсивность обслуживания). Теперь приведем мнемоническое правило составления уравнений Колмогорова: производная вероятности любого состояния равна сумме *потоков вероятности*, переводящих систему в это состояние, минус сумма всех *потоков вероятности*, выводящих систему из этого состояния [[4]](https://www.intuit.ru/studies/professional_skill_improvements/1602/courses/479/literature#literature.4).

Прежде чем применять правило Колмогорова, целесообразно изобразить размеченный *граф* состояний заданной системы. На [рис. 2.1](https://www.intuit.ru/studies/professional_skill_improvements/1602/courses/479/lecture/21082?page=1#image.2.1) показан пример размеченного графа состояний для системы M/M/3/5 ( m = 3, K = 5, \lambda — интенсивность входного *потока требований*, \mu — интенсивность обслуживания одним прибором).



**Рис. 2.1.**Размеченный граф состояний системы М/М/3/5

Используем мнемоническое правило составления уравнений Колмогорова для состояния 2, которое обведено окружностью. При этом потоки вероятности будут направлены по стрелкам. Если стрелка входит в *окружность*, то *поток вероятности* принимается положительным. Если стрелка выходит из окружности, то *поток вероятности* будет отрицательным. Относительно каждого состояния можно проводить воображаемую *окружность*. Дифференциальные уравнения имеют следующий вид:

|  |  |
| --- | --- |
| \frac{dP_{0}}{dt}=-\lambda P_{0}+\mu P_{1};\\ \frac{dP_{1}}{dt}=\lambda P_{0}-(\lambda +\mu) P_{1}+2\mu P_{2};\\ \frac{dP_{2}}{dt}=\lambda P_{1}-(\lambda +2\mu) P_{2}+3\mu P_{3};\\ \frac{dP_{3}}{dt}=\lambda P_{2}-(\lambda +3\mu) P_{3}+3\mu P_{4};\\ \frac{dP_{4}}{dt}=\lambda P_{3}-(\lambda +3\mu) P_{4}+3\mu P_{5};\\ \frac{dP_{5}}{dt}=\lambda P_{4}-3\mu P_{5};\\ | ( 2.1) |

Для решения систем дифференциальных уравнений типа (2.1) обычно задают естественные начальные условия:

|  |  |
| --- | --- |
| P_{0}(0)=1,\qquad P_{i}(0)=0,\qquad i=\overline{1,5}. | ( 2.2) |

Если в системе (2.1) *производные* приравнять нулю, то можно будет получить соотношения для расчета стационарных вероятностей состояний системы. При этом следует использовать нормировочное условие для n = 5:

|  |  |  |
| --- | --- | --- |
| \sum\limits_{k=0}^{5} P_{k}(t)=1. |  | ( 2.3) |

**Практическая часть**

**2.1. Пример моделирования системы типа М/М/M/K**

Система M/M/m/K — это система с пуассоновским входящим *потоком требований*, с экспоненциальным законом обслуживания в m приборах, с допустимым числом требований в системе, не превышающим K, которое не менее, чем заданное количество приборов обслуживания. Параметры системы постоянны, т. е. \lambda = const, \mu = const. Систему M/M/m/K называют многоканальной системой массового обслуживания с ограниченной длиной очереди.

*Пример 1*. Проинтегрируйте систему массового обслуживания М/М/4/7 при естественных граничных условиях и параметрах \lambda = 1.23, \mu = 0.678. Рассчитайте операционные характеристики системы [[2]](https://www.intuit.ru/studies/professional_skill_improvements/1602/courses/479/literature#literature.2).

Для решения примера сначала составим дифференциальные уравнения в соответствии с мнемоническим правилом Колмогорова:

\frac{dP_{0}}{dt}=-\lambda P_{0}+\mu P_{1};\\
\frac{dP_{1}}{dt}=\lambda P_{0}-(\lambda +\mu) P_{1}+2\mu P_{2};\\
\frac{dP_{2}}{dt}=\lambda P_{1}-(\lambda +2\mu) P_{2}+3\mu P_{3};\\
\frac{dP_{3}}{dt}=\lambda P_{2}-(\lambda +3\mu) P_{3}+4\mu P_{4};\\
\frac{dP_{4}}{dt}=\lambda P_{3}-(\lambda +4\mu) P_{4}+4\mu P_{5};\\
\frac{dP_{5}}{dt}=\lambda P_{4}-(\lambda +4\mu) P_{5}+4\mu P_{6};\\
\frac{dP_{6}}{dt}=\lambda P_{5}-(\lambda +4\mu) P_{6}+4\mu P_{7};\\
\frac{dP_{7}}{dt}=\lambda P_{6}-4\mu P_{7};\\

Естественные начальные условия:

P_{0}(0)=1,\qquad P_{i}(0)=0,\qquad i=\overline{1,7}.

Программный код решения примера в MATLAB:

function MMmK;

clc,close

% Параметры системы

L = 3.52;

M = 0.678;

m = 4;

K = 7;

%%%%%%%%%%%%%%%%%%%%%%%%

global A

A = [-L,M,0,0,0,0,0,0;

L,-(L+M),2\*M,0,0,0,0,0;

0,L,-(L+2\*M),3\*M,0,0,0,0;

0,0,L,-(L+3\*M),4\*M,0,0,0;

0,0,0,L,-(L+4\*M),4\*M,0,0;

0,0,0,0,L,-(L+4\*M),4\*M,0;

0,0,0,0,0,L,-(L+4\*M),4\*M;

0,0,0,0,0,0,L,-4\*M];

%% Численное интегрирование дифф. уравнений

P0 = [1;zeros(length(A)-1,1)];

T = [0,20];

[t,P] = ode23(@cmo, T, P0);

%% Построение диаграммы вероятностей состояний

%% line(t,P,'linew',2) %% с различными цветами

line(t,P(:,1),'linew',2, 'color','r') %% Po

line(t,P(:,2), 'linew',2,'lines','--') %% P1

line(t,P(:,3), 'linew',2,'lines','-.') %% P2

line(t,P(:,4), 'linew',2,'lines',':') %% P3

line(t,P(:,5), 'marker','o', 'color', 'm') %% P4

line(t,P(:,6), 'marker','h', 'color','k') %% P5

line(t,P(:,7), 'marker','p','color','r') %% P6

line(t,P(:,8), 'marker','>') %% P7

grid on

N = length(A)-1;

arr = [0:N]';

str = num2str(arr);

legend(strcat('\bf\itP\rm\bf\_', str, '(\itt\rm\bf)'));

title(sprintf('%s Вероятности состояний системы M/M/%d/%d', '\bf\fontsize{12}',m, K));

xlabel('\bf\it\fontsize{12} - - - - - - - - t - - - - - - - -')

ylabel('\bf\fontsize{12}\itP\rm\bf(\itt\rm\bf)');

set(gca, 'fontweight','bold', 'fontsize',10)

fprintf('\n Стационарные вероятности:\n');

for J = 1 : length(A)

fprintf('\tP%d = %f\n', J-1, P(end,J));

end

fprintf('\n\t ОПЕРАЦИОННЫЕ ХАРАКТЕРИСТИКИ:\n');

Pnot = P(end,end);

fprintf(' Вероятность отказа Pnot = %f\n', P(end,end));

Q = 1 - Pnot;

fprintf(' Относительная пропускная способность Q = %f\n', Q);

Ab = L\*Q;

fprintf(' Абсолютная пропускная способность A = %f\n', Ab);

Pq = sum(P(end, m+1:end));

fprintf(' Вероятность наличия очереди Pq = %f\n', Pq);

Ps = sum(P(end, m:end));

fprintf(' Вероятность загрузки всех каналов обслуживания Ps = %f\n', Ps);

Ns = [0:length(A)-1]\*P(end,:)';

fprintf(' Среднее количество требований в системе Ns = %f\n', Ns);

fprintf(' Среднее время пребывания требования в системе Ts = %f\n', Ns/L);

Nq = [0:(K-m)]\*P(end,m:K)';

fprintf(' Средняя длина очереди Nq = %f\n', Nq);

fprintf(' Среднее время пребывания требования в очереди Tq = %f\n', Nq/L);

function f = cmo(t,P);

%% Функция описания правых частей

%% дифференциальных уравнений

global A

f = A\*P;

Результат выполнения программы

Стационарные вероятности:

P0 = 0.004352

P1 = 0.022604

P2 = 0.058629

P3 = 0.101614

P4 = 0.131597

P5 = 0.171220

P6 = 0.221807

P7 = 0.288176

ОПЕРАЦИОННЫЕ ХАРАКТЕРИСТИКИ

Вероятность отказа: Pnot = 0.288176

Относительная пропускная способность: Q = 0.711824

Абсолютная пропускная способность: A = 2.505620

Вероятность наличия очереди: Pq = 0.812800

Вероятность загрузки всех каналов обслуживания: Ps = 0.914415

Среднее количество требований в системе: Ns = 5.175269

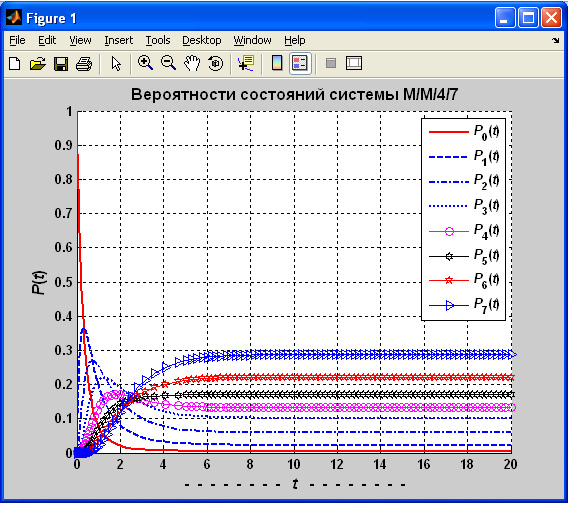
Среднее время пребывания требования в системе: Ts = 1.470247

Средняя длина очереди: Nq = 1.139458

Среднее время пребывания требования в очереди: Tq = 0.323710

Среднее время пребывания требования в системе было рассчитано по формуле Литтла. Для расчета среднего времени пребывания требования в очереди использовался аналог формулы Литтла [[2]](https://www.intuit.ru/studies/professional_skill_improvements/1602/courses/479/literature#literature.2).

Диаграмма с вероятностями состояний системы М/М/4/7 показана на [рис. 2.2](https://www.intuit.ru/studies/professional_skill_improvements/1602/courses/479/lecture/21082?page=1#image.2.2).



**Рис. 2.2.**Вероятности состояний системы М/М/4/7

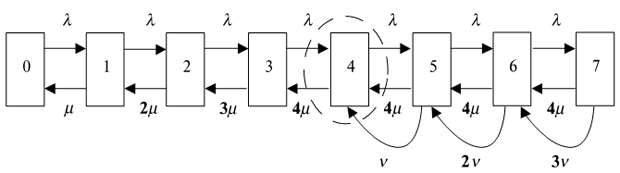
*Задание 1*

1. Рассчитайте нормировочное условие для трех значений времени.
2. Определите вероятность того, что в системе будет не более k требований. Значение k выбирайте случайно по равномерному закону из интервала целых чисел [1; 6] (см. help randperm ).
3. Примените диалоговое окно inputdlg (см. help inputdlg ) для ввода параметров системы.
4. Напишите программу для произвольно задаваемых параметров системы \lambda, \mu и значений m, K. Предусмотрите также ввод длительности интервала интегрирования.

**2.2. Пример моделирования системы типа М/М/M/K с ограниченным временем ожидания**

*Пример 2*. Получите операционные характеристики системы M/M/4/7 с ограниченным временем ожидания при следующих параметрах системы: \lambda = 3.52 \mbox{ }с^{–1} — интенсивность входного *потока требований*, \mu = 0.678 \mbox{ }с^{–1} — интенсивность обслуживания каждым прибором, \nu = 1.3\mbox{ }с^{–1} — интенсивность ухода из очереди "нетерпеливых" требований. Моделирование проведите в MATLAB и *GPSS*/PC.

Размеченный граф состояний системы показан на [рис. 2.3](https://www.intuit.ru/studies/professional_skill_improvements/1602/courses/479/lecture/21082?page=2#image.2.3).



**Рис. 2.3.**Граф состояний системы с ограниченным временем ожидания

На [рис. 2.3](https://www.intuit.ru/studies/professional_skill_improvements/1602/courses/479/lecture/21082?page=2#image.2.3) вокруг состояния 4 проведена воображаемая окружность для пояснения применения мнемонического правила составления дифференциальных уравнений Колмогорова. В соответствии с мнемоническим правилом получаем следующую систему дифференциальных уравнений:

\frac{dP_{0}}{dt}=-\lambda P_{0}+\mu P_{1};\\
\frac{dP_{1}}{dt}=\lambda P_{0}-(\lambda +\mu) P_{1}+2\mu P_{2};\\
\frac{dP_{2}}{dt}=\lambda P_{1}-(\lambda +2\mu) P_{2}+3\mu P_{3};\\
\frac{dP_{3}}{dt}=\lambda P_{2}-(\lambda +3\mu) P_{3}+4\mu P_{4};\\
\frac{dP_{4}}{dt}=\lambda P_{3}-(\lambda +4\mu) P_{4}+(4\mu +\nu) P_{5};\\
\frac{dP_{5}}{dt}=\lambda P_{4}-(\lambda +4\mu+\nu) P_{5}+(4\mu +2\nu)P_{6};\\
\frac{dP_{6}}{dt}=\lambda P_{5}-(\lambda +4\mu+2\nu) P_{6}+(4\mu+3\nu) P_{7};\\
\frac{dP_{7}}{dt}=\lambda P_{6}-(4\mu+3\nu) P_{7};\\

где P_{i},\mbox{  } i=\overline{0,7} — вероятности состояний системы.

Для интегрирования системы дифференциальных уравнений зададим естественные граничные условия:

P_{0}(0)=1,\qquad P_{i}(0)=0,\qquad i=\overline{1,7}.

Программный код решения примера в системе MATLAB:

function MMmKv

%% Система M/M/4/7 с ограниченным временем ожидания в очереди

% Параметры системы

L = 3.52;

M = 0.678;

v = 1.3;

m = 4;

K = 7;

% Матрица коэффициентов системы дифференциальных уравнений

global A

A = [-L, M, zeros(1,6);

L, -(L+M), 2\*M, zeros(1,5);

0, L, -(L+2\*M), 3\*M, zeros(1,4);

0, 0, L, -(L+3\*M), 4\*M, zeros(1,3);

zeros(1,3), L, -(L+4\*M), (4\*M+v), 0, 0;

zeros(1,4), L, -(L+4\*M+v), (4\*M+2\*v), 0;

zeros(1,5), L, -(L+4\*M+2\*v), (4\*M+3\*v);

zeros(1,6), L ,-(4\*M+3\*v)];

% Решение дифференциальных уравнений

T = [0,20]; % Время интегрирования

P0 = [1;zeros(length(A)-1,1)]; % Начальные условия

[t, P] = ode23(@cmo, T, P0);

%% Построение диаграмм вероятностей состояний

% line(t,P,'linew',2) %% сполошные линии с различными цветами

line(t,P(:,1),'linew',2, 'color','r') %% Po

line(t,P(:,2), 'linew',2,'lines','--') %% P1

line(t,P(:,3), 'linew',2,'lines','-.') %% P2

line(t,P(:,4), 'linew',2,'lines',':') %% P3

line(t,P(:,5), 'marker','o', 'color', 'm') %% P4

line(t,P(:,6), 'marker','h', 'color','k') %% P5

line(t,P(:,7), 'marker','p','color','r') %% P6

line(t,P(:,8), 'marker','>') %% P7

grid on

N = length(A)-1;

arr = [0:N]';

str = num2str(arr);

legend(strcat('\bf\itP\rm\bf\_', str, '(\itt\rm\bf)'));

title(sprintf('%s Вероятности состояний системы M/M/%d/%d', '\bf\fontsize{12}',m, K));

xlabel('\bf\it\fontsize{12} - - - - - - - - t - - - - - - - -')

ylabel('\bf\fontsize{12}\itP\rm\bf(\itt\rm\bf)');

set(gca, 'fontweight','bold', 'fontsize',10)

Pcm = P(end,:); % Стационарные вероятности

fprintf('\n Стационарные вероятности:\n');

for J = 1 : length(A)

fprintf('\tP%d = %f\n', J-1, Pcm(J));

end

fprintf('\n\t ОПЕРАЦИОННЫЕ ХАРАКТЕРИСТИКИ:\n');

Pnot = P(end,end);

fprintf(' Вероятность отказа Pnot = %f\n', P(end,end));

Q = 1 - Pnot;

fprintf(' Относительная пропускная способность Q = %f\n', Q);

Ab = L\*Q;

fprintf(' Абсолютная пропускная способность A = %f\n', Ab);

Pq = sum(P(end, m+1:end));

fprintf(' Вероятность наличия очереди Pq = %f\n', Pq);

Ps = sum(P(end, m:end));

fprintf(' Вероятность загрузки всех каналов обслуживания Ps = %f\n', Ps);

Ns = [0:length(A)-1]\*P(end,:)';

fprintf(' Среднее количество требований в системе Ns = %f\n', Ns);

fprintf(' Среднее время пребывания требования в системе Ts = %f\n', Ns/L);

Nq = [0:(K-m)]\*P(end,m:K)';

fprintf(' Средняя длина очереди Nq = %f\n', Nq);

fprintf(' Среднее время пребывания требования в очереди Tq = %f\n', Nq/L);

function f = cmo(t,P);

%% Функция описания правых частей

%% дифференциальных уравнений

global A

f = A\*P;

Результат выполнения программы

Стационарные вероятности:

P0 = 0.007887

P1 = 0.040947

P2 = 0.106289

P3 = 0.183963

P4 = 0.238701

P5 = 0.209568

P6 = 0.138707

P7 = 0.073939

ОПЕРАЦИОННЫЕ ХАРАКТЕРИСТИКИ

Вероятность отказа: Pnot = 0.073939

Относительная пропускная способность: Q = 0.926061

Абсолютная пропускная способность: A = 3.259736

Вероятность наличия очереди: Pq = 0.660915

Вероятность загрузки всех каналов обслуживания: Ps = 0.844877

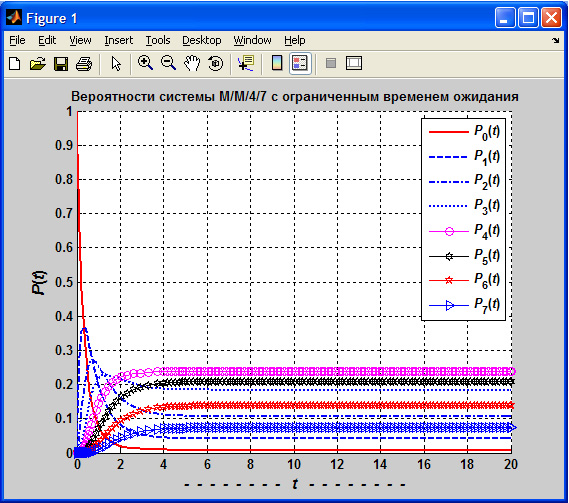
Среднее количество требований в системе: Ns = 4.157868

Среднее время пребывания требования в системе: Ts = 1.181213

Средняя длина очереди: Nq = 1.073958

Среднее время пребывания требования в очереди: Tq = 0.305102

Диаграмма вероятностей состояний системы М/М/4/7 с ограниченным временем ожидания показана на [рис. 2.4](https://www.intuit.ru/studies/professional_skill_improvements/1602/courses/479/lecture/21082?page=2#image.2.4).



**Рис. 2.4.**Вероятности состояний системы М/М/4/7 с ограниченным временем ожидания

*Задание 3*

1. Постройте зависимость среднего времени пребывания в очереди от интенсивности входного потока, изменяя интенсивность от 0.Х до 2.Х, где Х — номер компьютера (1, 2, 3, ...), за которым выполняется лабораторная работа.
2. Напишите программу для анализа системы с параметрами, вводимыми с клавиатуры пользователем, т. е. \lambda, \mu, \nu, m, K. Предусмотрите также ввод интервала *интегрирования дифференциальных уравнений*.

Для решения примера в системе *GPSS*/PC подготовим данные о функциях распределения входного потока, экспоненциальном обслуживании и экспоненциальном уходе из очереди "нетерпеливых" требований.

Программа в MATLAB формирования данных:

clear all,clc

L = 3.52; %% Интенсивность входного потока

M = 0.678; %% Интенсивность обслуживания

v = 1.3; %% Интенсивность ухода из очереди

%% Функции экспоненциального распределения

x = 0 : 0.2 : 20;

F = 1 - exp(-L\*x);

F2 = 1 - exp(-M\*x);

F3 = 1 - exp(-v\*x);

%%% Запись в текстовый файл MC2.txt

fid = fopen('MC2.txt', 'w');

%% Для блока generate

fprintf(fid, 'puas function RN1,C50\r\n');

for J = 1 : length(x)

if J <= 50

if mod(J, 5)

fprintf(fid, '%g,%g/', F(J), x(J));

else

fprintf(fid, '\r\n');

fprintf(fid, '%g,%g/', F(J), x(J));

end

end

end

fprintf(fid, '\r\n;-----------------------------------\r\n');

%% Для блоков advance

fprintf(fid, '\r\nexpM function RN2,C50\r\n');

for J = 1 : length(x)

if J <= 50

if mod(J, 5)

fprintf(fid, '%g,%g/', F2(J), x(J));

else

fprintf(fid, '\r\n');

fprintf(fid, '%g,%g/', F2(J), x(J));

end

end

end

fprintf(fid, '\r\n;-----------------------------------\r\n');

%% Для функции распределения ухода из очереди

fprintf(fid, '\r\nexpM function RN3,C50\r\n');

for J = 1 : length(x)

if J <= 50

if mod(J, 5)

fprintf(fid, '%g,%g/', F3(J), x(J));

else

fprintf(fid, '\r\n');

fprintf(fid, '%g,%g/', F3(J), x(J));

end

end

end

fclose(fid);

*GPSS*-программа приводится ниже. В ней задано условие обработки 1000 требований.

Программный код решения примера в системе *GPSS*/PC:

simulate

puas function RN1,C50

0,0/0.505397,0.2/0.755368,0.4/0.879004,0.6/

0.940155,0.8/0.970401,1/0.98536,1.2/0.992759,1.4/0.996419,1.6/

0.998229,1.8/0.999124,2/0.999567,2.2/0.999786,2.4/0.999894,2.6/

0.999948,2.8/0.999974,3/0.999987,3.2/0.999994,3.4/0.999997,3.6/

0.999998,3.8/0.999999,4/1,4.2/1,4.4/1,4.6/

1,4.8/1,5/1,5.2/1,5.4/1,5.6/

1,5.8/1,6/1,6.2/1,6.4/1,6.6/

1,6.8/1,7/1,7.2/1,7.4/1,7.6/

1,7.8/1,8/1,8.2/1,8.4/1,8.6/

1,8.8/1,9/1,9.2/1,9.4/1,9.6/

1,9.8/

;-----------------------------------

expM function RN2,C50

0,0/0.126808,0.2/0.237536,0.4/0.334223,0.6/

0.418649,0.8/0.492369,1/0.556741,1.2/0.612949,1.4/0.662031,1.6/

0.704888,1.8/0.742311,2/0.774988,2.2/0.803521,2.4/0.828436,2.6/

0.850192,2.8/0.869189,3/0.885777,3.2/0.900261,3.4/0.912909,3.6/

0.923953,3.8/0.933596,4/0.942017,4.2/0.949369,4.4/0.95579,4.6/

0.961396,4.8/0.966291,5/0.970566,5.2/0.974298,5.4/0.977558,5.6/

0.980403,5.8/0.982888,6/0.985058,6.2/0.986953,6.4/0.988607,6.6/

0.990052,6.8/0.991314,7/0.992415,7.2/0.993377,7.4/0.994217,7.6/

0.99495,7.8/0.995591,8/0.99615,8.2/0.996638,8.4/0.997064,8.6/

0.997437,8.8/0.997762,9/0.998045,9.2/0.998293,9.4/0.99851,9.6/

0.998699,9.8/

;-----------------------------------

expV function RN3,C50

0,0/0.228948,0.2/0.405479,0.4/0.541594,0.6/

0.646545,0.8/0.727468,1/0.789864,1.2/0.837974,1.4/0.87507,1.6/

0.903672,1.8/0.925726,2/0.942731,2.2/0.955843,2.4/0.965953,2.6/

0.973748,2.8/0.979758,3/0.984392,3.2/0.987966,3.4/0.990721,3.6/

0.992845,3.8/0.994483,4/0.995746,4.2/0.99672,4.4/0.997471,4.6/

0.99805,4.8/0.998497,5/0.998841,5.2/0.999106,5.4/0.999311,5.6/

0.999469,5.8/0.99959,6/0.999684,6.2/0.999756,6.4/0.999812,6.6/

0.999855,6.8/0.999888,7/0.999914,7.2/0.999934,7.4/0.999949,7.6/

0.999961,7.8/0.99997,8/0.999977,8.2/0.999982,8.4/0.999986,8.6/

0.999989,8.8/0.999992,9/0.999994,9.2/0.999995,9.4/0.999996,9.6/

0.999997,9.8/

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

T\_s table m1,0,2,6

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

10 GENERATE fn$puas

20 test LE q1,3,exit1

30 test L qx1,fn$expV,exit2

40 QUEUE 1

50 TRANSFER ALL,CHAN1,CHAN4,3

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

100 CHAN1 SEIZE zet1

110 assign 1,zet1

120 transfer ,come

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

200 chan2 SEIZE zet2

210 assign 1,zet2

220 transfer ,come

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

300 chan3 SEIZE zet3

310 assign 1,zet3

320 transfer ,come

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

400 chan4 SEIZE zet4

410 assign 1,zet4

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

510 come DEPART 1

520 advance fn$expM

530 release p1

540 tabulate T\_s

550 work terminate 1

560 exit1 terminate

570 exit2 terminate

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

START 1000

;end

В программе использованы следующие стандартные числовые атрибуты: q1 — длина первой очереди, qx1 — среднее время пребывания в первой очереди без нулевых входов, m1 — время пребывания транзакта в модели. Освобождение того или иного устройства блоком release выполнено на основе записи в первый параметр транзактов имени устройства, например, assign 1,zet4.

Программа реализует следующий алгоритм. Поступающие по пуассоновскому закону требования (транзакты в системах *GPSS*) попадают в очередь на обслуживание. Если в очереди меньше или равно 3 транзактам, то они ожидают начала обслуживания. В противном случае они покидают систему по метке exit1. Проверка длины очереди осуществляется блоком TEST в строке 20. Время ожидания соотносится со случайным временем ожидания "нетерпеливых" требований —— со значением функции распределения expV. Если оно меньше, то требования поступают в устройства обслуживания, в одно из свободных устройств, начиная с первого (реализуется блоком TRANSFER в режиме ALL ). Поступающим в устройства обслуживания транзактам в первый параметр записывается имя устройства ( zet1, … , zet4 ), после чего они отправляются по метке come в блок DEPART, который освобождает одно место в очереди под номером 1. Далее происходит обслуживание транзактов по экспоненциальному закону блоком ADVANCE в строке 520. Вывод транзакта из работающего устройства осуществляется блоком RELEASE по первому параметру, в котором записано имя устройства. Обслуженные транзакты выводятся из системы блоком TERMINATE с меткой work.

Для получения распределения времени пребывания в модели использован оператор таблицы table с меткой T_s. Сбор статистики для введенной таблицы осуществляется блоком TABULATE.

Результат выполнения программы система *GPSS*/PC оформляет в виде файла стандартного отчета, который имеет следующий вид:

GPSS/PC Report file REPORT.GPS. (V 2, # 37349) 01-08-2010 11:16:14 page 1

START\_TIME END\_TIME BLOCKS FACILITIES STORAGES FREE\_MEMORY

0 416 23 4 0 137072

LINE LOC BLOCK\_TYPE ENTRY\_COUNT CURRENT\_COUNT RETRY

10 1 GENERATE 14227 0 0

20 2 TEST 14227 0 0

30 3 TEST 13127 0 0

40 4 QUEUE 1002 0 0

50 5 TRANSFER 1288 1 0

100 CHAN1 SEIZE 317 0 0

110 7 ASSIGN 317 0 0

120 8 TRANSFER 317 0 0

200 CHAN2 SEIZE 265 0 0

210 10 ASSIGN 265 0 0

220 11 TRANSFER 265 0 0

300 CHAN3 SEIZE 232 0 0

310 13 ASSIGN 232 0 0

320 14 TRANSFER 232 0 0

400 CHAN4 SEIZE 187 0 0

410 16 ASSIGN 187 0 0

510 COME DEPART 1001 0 0

520 18 ADVANCE 1001 1 0

530 19 RELEASE 1000 0 0

540 20 TABULATE 1000 0 0

550 WORK TERMINATE 1000 0 0

560 EXIT1 TERMINATE 1100 0 0

570 EXIT2 TERMINATE 12125 0 0

FACILITY ENTRIES UTIL. AVE.\_TIME AVAILABLE OWNER PEND INTER RETRY DELAY

ZET1 317 0.776 1.02 1 0 0 0 1 0

ZET2 265 0.675 1.06 1 14161 0 0 1 0

ZET3 232 0.557 1.00 1 0 0 0 1 0

ZET4 187 0.507 1.13 1 0 0 0 1 0

QUEUE MAX CONT. ENTRIES ENTRIES(0) AVE.CONT. AVE.TIME AVE.(-0) RETRY

1 4 1 1002 765 0.72 0.30 1.27 0

TABLE MEAN STD.DEV. RETRY RANGE FREQUENCY CUM.%

T\_S 1.34 1.58 0

- 0 374 37.40

0 - 2 448 82.20

2 - 4 122 94.40

4 - 6 42 98.60

6 - 8 14 100.00

XACT\_GROUP GROUP\_SIZE RETRY

POSITION 0 0

По файлу стандартного отчета определим следующие операционные характеристики: среднее время пребывания в очереди и среднее время пребывания в модели (в системе). Эти характеристики обозначим Tq и Ts соответственно.

Значение Tq расположено в поле AVE.TIME статистики очереди. Оно равно Tq = 0.30.

Значение Ts есть среднее значение таблицы T_S, т. е. в поле MEAN. Оно равно Ts = 1.34.

Поскольку очередь имеет нулевые входы – поле ENTRIES(0), для расчета *средней длины очереди* Nq используем аналог формулы Литтла [[2]](https://www.intuit.ru/studies/professional_skill_improvements/1602/courses/479/literature#literature.2), т. е.

Nq = \lambda \cdot Tq = 3.52\cdot 0.30 = 1.0560.

Для определения среднего количества требований Ns в системе используем формулу Литтла:

Ns = \lambda \cdot Ts = 3.52\cdot 1.34 = 4.7168.

*Задание 4*

1. По данным файла стандартного отчета определите *вероятность отказа*, *относительную пропускную способность*, *абсолютную пропускную способность*.
2. Если требуется, то внесите изменения в программу для определения ве-роятности наличия очереди и вероятности загрузки всех каналов обслуживания.
3. Внесите изменения в программу для определения вероятностей состояния системы.
4. Полученные результаты сведите в таблицу, в которую также внесите операционные характеристики системы М/М/4/7 с ограниченным временем ожидания, полученные с помощью анализа в MATLAB.

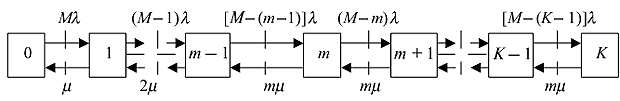
**2.3. Пример моделирования системы типа М/М/M/K/M**

Система M/M/m/K/M — это система с пуассоновским входящим пото-ком требований, с экспоненциальным законом обслуживания в m приборах, с допустимым числом требований в системе, не превышающим K, и с ограниченным числом источников нагрузки, которые создают поток из M требований. Общее число K требований в системе заключено в интервале m \le K \le M, где M — число требований, формируемых конечным числом источников нагрузки.

Предполагается, что требования, поступающие в систему, когда в ней уже имеется K требований, теряются и немедленно возвращаются в группу поступающих так, как будто бы они полностью обслужены. Для описанного функционирования системы и ее заданного буквенного обозначения можно определить ее параметры в соответствии с процессом размножения и гибели в следующем виде:

\lambda_{k}=
\left
\begin{cases}
\lambda(M-k), 0\le k \le K-1, \lambda=const;\\
0\mbox{  }\mbox{    в остальных случаях}\\
\end{cases}
\right.\\
\\
\\
\mu_{k}=
\left
\begin{cases}
k\mu, 0\le k \le m, \mu=const;\\
m\mu, m\le k \le K.\\
\end{cases}
\right.

Диаграмма интенсивностей переходов для системы M/M/m/K/M будет представлять собой конечный размеченный граф состояний, который показан на [рис. 2.5](https://www.intuit.ru/studies/professional_skill_improvements/1602/courses/479/lecture/21082?page=3#image.2.5).



**Рис. 2.5.**Граф состояний системы M/M/m/K/M

На [рис. 2.5](https://www.intuit.ru/studies/professional_skill_improvements/1602/courses/479/lecture/21082?page=3#image.2.5) вертикальными штриховыми линиями размечены границы между состояниями системы, с помощью которых можно найти стационарные вероятности состояний по следующему мнемоническому правилу: на *границе раздела* двух состояний размеченного графа *поток вероятности* слева от границы равен *потоку вероятности* справа от границы.

Для определения дифференциальных уравнений относительно вероятностей состояний следует каждое состояние описать воображаемой окружностью и далее применить мнемоническое правило составления дифференциальных уравнений Колмогорова: производная вероятности любого состояния равна сумме *потоков вероятности*, переводящих систему в это состояние, минус сумма всех *потоков вероятности*, выводящих систему из этого состояния [[4]](https://www.intuit.ru/studies/professional_skill_improvements/1602/courses/479/literature#literature.4).

Запишем следующее выражение для определения стационарных вероятностей состояний от 0 до m – 1:

p_{k}=p_{0}\rho^{k}C_{M}^{k},\mbox{  }0\le k\le m-1,

где:

C_{M}^{k}=\frac{M!}{k!(M-k)!} — *биномиальные коэффициенты*;

\rho = \lambda / \mu,

p_{0}=\frac{1}{\sum\limits_{k=0}^{m-1} \rho^{k}C_{M}^{k}+\sum\limits_{k=m}^{K} \rho^{k}C_{M}^{k}\frac{k!m^{m-k}}{m!}}  — вероятность нулевого состояния.

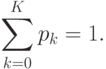
Расчет вероятностей состояний от m до K:

p_{m+r}=p_{0}\rho^{m+r}C_{M}^{m+r}\frac{(m+r)!}{m!m^{r}},\mbox{  }r=0,1,2,...,K-m.

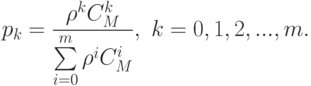
Если в последнем выражении сделать замену m+r=k, r=k-m, то получим

p_{k}=p_{0}\rho^{k}C_{M}^{k}\frac{k!m^{m-k}}{m!},\mbox{  }m\le k\le K.

Вероятность нулевого состояния была определена из нормировочного условия:



Рассмотрим случай, когда система M/M/m/K/M работает в режиме чистых потерь, т. е. когда параметры системы удовлетворяют условию M\ge K=m. Расчет вероятностей состояний будет определяться при K = m:



Распределение вероятностей в соответствии с последней формулой называется распределением ***Энгсета*** .

В соответствии с размеченным графом состояний и с помощью мнемонического правила составим следующие дифференциальные уравнения относительно вероятностей состояний системы:

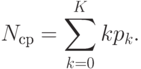
\frac{dP_{0}}{dt}=-M\lambda P_{0}+\mu P_{1};\\
\frac{dP_{1}}{dt}=M\lambda P_{0}-[(M-1)\lambda +\mu] P_{1}+2\mu P_{2};\\
\frac{dP_{2}}{dt}=(M-1)\lambda P_{1}-[(M-2)\lambda +2\mu] P_{2}+3\mu P_{3};\\
.............................\\
\frac{dP_{m}}{dt}=[M-(m-1)]\lambda P_{m-1}-[(M-m)\lambda +m\mu] P_{m}+m\mu P_{m+1};\\
.............................\\
\frac{dP_{K-1}}{dt}=[M-(K-2)]\lambda P_{K-2}-\{[M-(KI-1)]\lambda +m\mu\} P_{K-1}+m\mu P_{K};\\
\frac{dP_{K}}{dt}=[M-(K-1)]\lambda P_{K-1}-m\mu P_{K}.

Для *решения дифференциальных уравнений* следует задать начальные условия. Обычно используются естественные начальные условия, т. е.

P_{0}(0)=1,\qquad P_{j}(0)=0,\qquad j=1,2,...,K.

Для стационарного режима рассмотрим ряд операционных характеристик в достаточно общем виде.

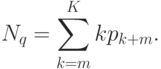
*Среднее число требований в системе*:



*Среднее время пребывания одного требования в системе* определим по формуле Литтла:

T=\frac{N_{ср}}{\lambda}.

*Средняя длина очереди*:



*Среднее время пребывания требования в очереди* определим по формуле Литтла:

T_{q}=\frac{N_{q}}{\lambda}.

*Вероятность отказа в обслуживании* соответствует вероятности того, что в системе находится K требований (максимально допустимое число):

p_{отк}=p_{K}=p_{0}\rho^{K}C_{M}^{K}\frac{K!m^{m-K}}{m!}.

*Относительная пропускная способность*:

Q=1-p_{отк}.

*Абсолютная пропускная способность*:

A=\lambda Q.

Таким образом, операционные характеристики рассчитываются по известным стационарным вероятностям и параметрам системы.

*Пример 3*. Для системы массового обслуживания М/М/4/7/9 постройте *переходные вероятности* состояний и определите для нее операционные характеристики при следующих параметрах: \lambda = 3.52 с^{–1}, \mu = 0.678 с^{–1}. Среда программирования — MATLAB.

Запишем с буквенными коэффициентами дифференциальные уравнения для системы М/М/4/7/9:

\frac{dP_{0}}{dt}=-M\lambda P_{0}+\mu P_{1},\\
\frac{dP_{1}}{dt}=M\lambda P_{0}-[(M-1)\lambda +\mu] P_{1}+2\mu P_{2},\\
\frac{dP_{2}}{dt}=(M-1)\lambda P_{1}-[(M-2)\lambda +2\mu] P_{2}+3\mu P_{3},\\
\frac{dP_{3}}{dt}=(M-2)\lambda P_{2}-[(M-3)\lambda +3\mu] P_{3}+4\mu P_{4},\\
\frac{dP_{4}}{dt}=(M-3)\lambda P_{3}-[(M-4)\lambda +4\mu] P_{4}+4\mu P_{5},\\
\frac{dP_{5}}{dt}=(M-4)\lambda P_{4}-[(M-5)\lambda +4\mu] P_{5}+4\mu P_{6},\\
\frac{dP_{6}}{dt}=(M-5)\lambda P_{5}-[(M-6)\lambda +4\mu] P_{6}+4\mu P_{7},\\
\frac{dP_{7}}{dt}=[M-(K-1)]\lambda P_{6}-4\mu P_{7},\\

где:

М = 9;
K = 7; 
\lambda = 3.52 с^{–1}; 
\mu = 0.678 с^{–1}.

Полученную систему представим в матричном виде:

\frac{dP}{dt}=AP,

где *матрица* А размера 8\times 8 составляется из коэффициентов при вероятностях в правой части уравнений.

Для *интегрирования дифференциальных уравнений* примем естественные граничные условия.

Программный код решения примера в MATLAB:

function MMmKM;

clc,close all

% Параметры системы M/M/4/7/9

L = 2.2;

M = 3.678;

m = 4;

K = 7;

N = 9;

% Матрица коэффициентов А

global A

A = [-N\*L,M,zeros(1,6);

N\*L,-[(N-1)\*L+M],2\*M,zeros(1,5);

zeros(1,1),(N-1)\*L,-[(N-2)\*L+2\*M],3\*M,zeros(1,4);

zeros(1,2),(N-2)\*L,-[(N-3)\*L+3\*M],4\*M,zeros(1,3);

zeros(1,3),(N-3)\*L,-[(N-4)\*L+4\*M],4\*M,zeros(1,2);

zeros(1,4),(N-4)\*L,-[(N-5)\*L+4\*M],4\*M,zeros(1,1);

zeros(1,5),(N-5)\*L,-[(N-6)\*L+4\*M],4\*M;

zeros(1,6),(N-6)\*L,-4\*M];

T = [0,2]; %% интервал интегрирования

P0 = [1,zeros(1,length(A)-1)]; %% начальные условия

[t,P] = ode23(@cmo,T,P0);

%% Построение диаграмм вероятностей состояний

% line(t,P,'linew',2) %% сплошные линии с различными цветами

line(t,P(:,1),'linew',2, 'color','r') %% Po

line(t,P(:,2), 'linew',2,'lines','--') %% P1

line(t,P(:,3), 'linew',2,'lines','-.') %% P2

line(t,P(:,4), 'linew',2,'lines',':') %% P3

line(t,P(:,5), 'marker','o', 'color', 'm') %% P4

line(t,P(:,6), 'marker','h', 'color','k') %% P5

line(t,P(:,7), 'marker','p','color','r') %% P6

line(t,P(:,8), 'marker','>') %% P7

grid on

Na = length(A) - 1;

arr = [0:Na]';

str = num2str(arr);

legend(strcat('\bf\itP\rm\bf\_', str, '(\itt\rm\bf)'));

title(sprintf('%sСистема M/M/%d/%d/%d; %s%g; %s%g',...

'\bf\fontsize{11}',m,K,N,'\lambda = ',L,'\mu = ',M));

xlabel('\bf\it\fontsize{12} - - - - - - - - t - - - - - - - -');

ylabel('\bf\fontsize{12}\itP\rm\bf(\itt\rm\bf)');

set(gca, 'fontweight','bold', 'fontsize',10)

Pcm = P(end,:); % Стационарные вероятности

fprintf('\n Стационарные вероятности системы M/M/%d/%d/%d:\n', m,K,N);

for J = 1 : length(A)

fprintf('\tP%d = %f\n', J-1, Pcm(J));

end

fprintf('\n ОПЕРАЦИОННЫЕ ХАРАКТЕРИСТИКИ СИСТЕМЫ M/M/%d/%d/%d:\n', m,K,N);

Pnot = P(end,end);

fprintf(' Вероятность отказа Pnot = %f\n', P(end,end));

Q = 1 - Pnot;

fprintf(' Относительная пропускная способность Q = %f\n', Q);

Ab = L\*Q;

fprintf(' Абсолютная пропускная способность A = %f\n', Ab);

Pq = sum(P(end, m+1:end));

fprintf(' Вероятность наличия очереди Pq = %f\n', Pq);

Ps = sum(P(end, m:end));

fprintf(' Вероятность загрузки всех каналов обслуживания Ps = %f\n', Ps);

Ns = [0:length(A)-1]\*P(end,:)';

fprintf(' Среднее количество требований в системе Ns = %f\n', Ns);

fprintf(' Среднее время пребывания требования в системе Ts = %f\n', Ns/L);

Nq = [0:(K-m)]\*P(end,m:K)';

fprintf(' Средняя длина очереди Nq = %f\n', Nq);

fprintf(' Среднее время пребывания требования в очереди Tq = %f\n', Nq/L);

function f = cmo(t,P)

% М-функция описания правых частей дифференциальных уравнений:

global A

f = A\*P;

Результат выполнения программы

Стационарные вероятности системы M/M/4/7/9:

P0 = 0.013187

P1 = 0.070956

P2 = 0.169841

P3 = 0.236851

P4 = 0.212652

P5 = 0.158847

P6 = 0.095048

P7 = 0.042619

ОПЕРАЦИОННЫЕ ХАРАКТЕРИСТИКИ СИСТЕМЫ M/M/4/7/9

Вероятность отказа: Pnot = 0.042619

Относительная пропускная способность: Q = 0.957381

Абсолютная пропускная способность: A = 2.106238

Вероятность наличия очереди: Pq = 0.509166

Вероятность загрузки всех каналов обслуживания: Ps = 0.746017

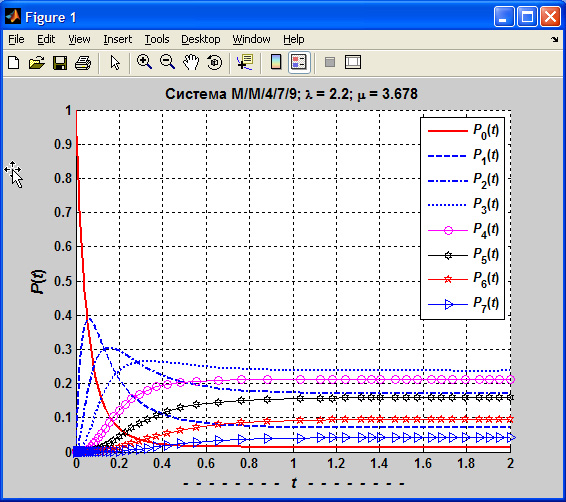
Среднее количество требований в системе: Ns = 3.634652

Среднее время пребывания требования в системе: Ts = 1.652115

Средняя длина очереди: Nq = 0.815489

Среднее время пребывания требования в очереди: Tq = 0.370677

Зависимости вероятностей состояний от времени показаны на [рис. 2.6](https://www.intuit.ru/studies/professional_skill_improvements/1602/courses/479/lecture/21082?page=4#image.2.6).



**Рис. 2.6.**Изменение вероятностей состояний во времени системы М/М/4/7/9

*Задание 5*

1. Рассчитайте стационарные вероятности по аналитическим формулам (см. теоретическую часть). Сравните результаты.
2. Постройте зависимость среднего времени пребывания в очереди от интенсивности входного потока, изменяя интенсивность от 0.Х до 5.Х, где Х — номер компьютера (1, 2, 3, ...), за которым выполняется лабораторная работа.
3. Напишите программу для анализа системы с параметрами, вводимыми с клавиатуры пользователем, т. е. \lambda, \mu, m, K, М. Предусмотрите также ввод интервала *интегрирования дифференциальных уравнений*.

**Контрольные вопросы**

1. Что называется простейшим пуассоновским потоком?
2. По какому закону распределены интервалы времени между требованиями в простейшем пуассоновском потоке?
3. Что такое символика Кендалла, применяемая для систем массового обслуживания?
4. Что определяет собой точка на *графике функции* распределения случайной величины?
5. Какие случайные величины в системе массового обслуживания являются дискретными, а какими непрерывными?
6. Что означает *несовместность событий*?
7. Каким условиям должны отвечать начальные условия при *решении дифференциальных уравнений* относительно вероятностей состояния системы массового обслуживания?
8. Какова длина очереди в системе M/M/m/K/M?
9. Через какой тип данных могут выражаться интенсивность входного *потока требований* и интенсивность обслуживания требований в системах массового обслуживания?

Во многих программных средах имеются генераторы случайных (псевдослучайных) чисел, которые обеспечивают формирование случайных чисел, равномерно распределенных в интервале [0; 1]. На основе этого распределения могут быть сформированы выборки случайных чисел с каким-либо другим законом распределения случайной величины. В некоторых случаях этого можно добиться на основе метода инверсии, или метода обратной функции. Суть метода обратной функции заключается в следующем. Пусть есть некоторая *функция* распределения F(t) случайной величины t, которая по определению является неотрицательной и не превосходящей значения +1. Если выбрать на оси ординат *графика функции* F(t) случайную точку r, то мы сможем получить *значение* величины t такое, что F(t) = r. Как известно, *функция* распределения есть *вероятность* того, что *случайная величина* t_0 не превысит своего некоторого значения t, т. е. F(t) = P(t_0 < t). На основе заданной функции распределения F(t) существует однозначное соответствие между значением аргумента функции и значением самой функции, что позволяет записать *выражение* F(t) = P(t0 < t) = P(R < F(t)), где R — *случайная величина*, равномерно распределенная в интервале [0; 1]. Используя соотношение f(t) = dF(t)/dt, можно записать функцию распределения случайной величины t:

|  |  |
| --- | --- |
| F(t)=P(t_{0} < t)=\int\limits_{0}^{t} f(\tau)d\tau =P(R<F(t))=\int\limits_{0}^{F(t)} f(r)dr. | ( 3.1) |

Таким образом, последовательность случайных чисел r_1, r_2, …, принадлежащая интервалу [0; 1], преобразуется в последовательность t_1, t_2, …, которая имеет заданную функцию распределения F(t) и, соответственно, функцию плотности f(t).

Метод обратной функции нельзя применять напрямую к непрерывным распределениям, для которых *функция* распределения не может быть выражена в квадратурах от соответствующей функции плотности. Типичными примерами такого рода являются *нормальное распределение*, гамма-распределение, логарифмически-*нормальное распределение*, а также *дискретное распределение* Пуассона. В таких случаях для получения выборок случайных чисел можно воспользоваться одним из следующих методов:

1. аппроксимацией непрерывной функции F(t) дискретной функцией распределения;
2. получением с помощью статистических соотношений необходимой информации на основе других распределений, имеющих простую аналитическую форму.

Например, в случае второго метода, *случайная величина*, подчиняющаяся распределению Эрланга k -го порядка (или гамма-распределению с целочисленным параметром), представляет собой сумму экспоненциально распределенных случайных величин, а время между наступлениями событий в пуассоновском процессе также распределено экспоненциально.

**Практическая часть**

**1. Формирование выборки случайных чисел с равномерным распределением в заданном интервале**

Равномерно распределенная случайная величина в интервале [a; b] имеет функцию плотности

f(t)=\frac{1}{b-a}.

Определим функцию распределения в соответствии с (1.1) и приравняем случайному числу R, равномерно распределенному в интервале [0; 1]:

|  |  |
| --- | --- |
| F(t)=R=\int\limits_{a}^{t} \frac{d\tau}{b-a}=\left.\frac{\tau}{b-a}\right|_{a}^{t}=\frac{t}{b-a}-\frac{a}{b-a}=\frac{t-a}{b-a};\Rightarrow t=a+(b-a)R. | ( 3.2) |

Если требуется сформировать выборку из N случайных чисел из интервала [a; b], то выражение (3.2) перепишем в виде

|  |  |
| --- | --- |
| t_{i}=a+(b-a)R_{i},\mbox{  }i=\overline{1,N}. | ( 3.3) |

В системе MATLAB случайные числа, равномерно распределенные в интервале [0; 1], формируются встроенной функцией rand (см. help rand ).

Рассмотрим следующий пример формирования выборки случайных чисел в системе MATLAB.

*Пример 1*. Сформируйте 20 случайных чисел с равномерным распределением из интервала [2; 5].

Программный код решения примера (возможно, в командном окне MATLAB):

a = 2; b = 5; N = 20;

t20 = a + (b - a)\*rand(N,1)

*Задание 1*

1. Рассчитайте теоретическое значение математического ожидания случайной величины, равномерно распределенной в интервале [2; 5]. Рассчитайте среднее значение величины t20 при трехкратном обращении к программе. Сравните результаты.
2. Постройте функцию распределения случайной величины из интервала по сформированной выборке. Интервал [a; b] примите следующим (в соответствии с номером компьютера, за которым выполняется лабораторная работа):
3. №1: [–1; 1.1]; №2: [–0.2; 2.2]; №3: [–3; 3.3]; №4: [–4.4; 6.6]; №5: [–5, 5.5];

№6: [–6.6, –3.3]; №7: [–7, 8]; №8: [–8, 8.8]; №9: [–9, 9]; №10: [–10.1, 10.1].

**2. Формирование выборки случайных чисел с экспоненциальным распределением**

Функция плотности f(t) экспоненциального распределения случайной величины и функция распределения F(t) даются формулами

f(t)=\lambda e^{-\lambda t},\qquad F(t)=1-e^{-\lambda t},

где \lambda — параметр экспоненциального распределения, имеющий размерность с^{-1}.

В соответствии с методом обратной функции запишем откуда найдем t:

|  |  |
| --- | --- |
| t=-\frac{1}{\lambda}\ell n (1-R). | ( 3.4) |

Известно, что функция случайной величины R имеет такое же распределение, что и сама случайная величина R. Поэтому (3.4) представим в виде

|  |  |
| --- | --- |
| t_{i}=-\frac{1}{\lambda}\ell n (R_{i}),\qquad i=\overline{1,N}. | ( 3.5) |

Рассмотрим пример формирования выборки случайных чисел с экспоненциальным распределением в MATLAB.

*Пример 2*. Сформируйте N = 20 случайных чисел с экспоненциальным распределением по методу обратной функции.

Программный код решения примера:

L = 0.25; N = 20;

texp20 = -1/L\*log(rand(N,1))

Если воспользоваться встроенной функцией exprnd (см. help exprnd ) системы MATLAB, то будем иметь следующий программный код:

L = 0.25; N = 20;

texp20M = exprnd(1/L,N,1)

Следует обратить внимание, что аргументом функции exprnd является величина математического ожидания ( М = 1/\lambda ) экспоненциального распределения.

Для приведенных программ результаты их выполнения не приводятся, так как они будут различными при каждом выполнении программы.

В системе MATLAB *эмпирическую функцию* распределения можно построить с помощью cdfplot (см. help cdfplot ). Тогда программа формирования и построения функции распределения примет следующий вид:

clear,clc

L = 0.25; N = 20;

texp20 = –1/L\*log(rand(N,1));

cdfplot(texp20), %% специальная графическая функция

title('\bf Функция экспоненциального распределения');

xlabel('\bf - - - - - - - t - - - - - - - ');

ylabel('\bf F(t)');

*Примечание*. Моделирование случайной величины с экспоненциальным распределением и заданным параметром можно произвести, взяв за основу моделирование случайной величины с экспоненциальным распределением и параметром, равным единице [[21]](https://www.intuit.ru/studies/professional_skill_improvements/1602/courses/479/literature#literature.21). В этом случае поступают следующим образом:

* генерируют значения случайной величины, распределенной по экспоненциальному закону с параметром \lambda = 1 ;
* находят произведение полученного значения и математического ожидания случайной величины, у которой \lambda \ne 1. Математическое ожидание экспоненциально распределенной величины обратно пропорционально параметру.

*Задание 2*

1. Постройте *эмпирическую функцию* плотности экспоненциального распределения по сформированному массиву случайных чисел.
2. Постройте *эмпирическую функцию* распределения по сформированному массиву случайных чисел без применения функции cdfplot.
3. Сравните выборку случайных чисел, распределенных по экспоненциальному закону с параметром \lambda = 0.25, и выборку чисел, полученных для \lambda = 1, с последующим преобразованием — умножением математического ожидания на массив сформированных чисел, для которых \lambda = 0.25.
4. Сформируйте выборку случайных чисел с экспоненциальным распределением по формуле (3.5) с помощью операторов цикла.

**3. Формирование нормально распределенных случайных чисел по методу бокса и маллера**

В соответствии с методом Бокса–Маллера случайные числа формируются парами с математическим ожиданием, близким к значению m = 0, и средним квадратическим отклонением, близким к единице, т. е. s = 1:

z_{1}=-2\ell nR_{1}\cos(2\pi R_{2});\qquad z_{2}=-2\ell nR_{1}\sin(2\pi R_{2}),

где R_{1}, R_{2} — случайные числа из интервала (0; 1) с равномерным распределением.

Выборка случайных чисел, сформированная по методу Бокса–Маллера, может центрироваться с целью получения параметров *нормального закона* практически равным своим теоретическим значениям стандартного *нормального закона*, т. е. m = 0, s = 1.

*Пример 3*. Сформируйте N = 500 случайных чисел в соответствии со стандартным *нормальным законом*.

Программный код решения примера:

clear, clc

N = 500;

z1 = -2\*log(rand(N,1)).\*cos(2\*pi\*rand(N,1));

z2 = -2\*log(rand(N,1)).\*sin(2\*pi\*rand(N,1));

z = sort([z1;z2]);

%-------------- Средняя величина---------------------

mz = sum(z)/(2\*N);

fprintf('\n\t m = %g\n',mz)

%-------------- Стандартное отклонение ------------

stz = sqrt(sum((z-mz).^2)/(length(z)-1));

fprintf('\t s = %g\n', stz)

%--------------- Нормализованные данные ----------

zz = (z-mz)/stz;

mzN = sum(zz)/(2\*N);

fprintf('\t m = %g\n', mzN)

stzN = sqrt(sum((zz-mzN).^2)/(2\*N-1));

fprintf('\t s = %g\n', stzN)

*Задание 3*

1. В программе примените циклические операции расчета средних значений и средних квадратических отклонений.
2. Рассчитайте параметры *нормального закона* для сформированной выборки случайных чисел с помощью функций mean и std, сравните результаты.
3. Постройте функцию плотности и функцию распределения для нормализованных данных. Сравните с функциями для данных, получаемых с помощью встроенной функции randn.
4. Напишите программу формирования выборки случайных чисел, распределенных по *нормальному закону* в соответствии с методом Марсальи–Брея по следующему алгоритму:
   1. Генерируются два равномерно распределенных случайных числа R1, R2 из интервала [0; 1].
   2. Формируются два соотношения V1= –1 + 2R1, V2= –1 + 2R2.
   3. Составляется сумма S: S = V1^{2} + V2^{2}.
   4. Если S \ge 1, то пп. 1-3 повторяются.
   5. Если S < 1, то вычисляется первая пара случайных чисел t1, t2:

t1=V1\sqrt{\frac{-2\ell nS}{S}},\qquad t2=V2\sqrt{\frac{-2\ell nS}{S}}

Для выборки случайных чисел, сформированных по методу Марсальи–Брея, рассчитать параметры *нормального закона*, т. е. математическое ожидание (среднее) и среднее квадратическое отклонение (*стандартное отклонение*).

*Примечание*. Для формирования выборки X нормально распределенных случайных чисел с произвольными параметрами m_{x} и s_{x} следует воспользоваться формулой

|  |  |
| --- | --- |
| X=m_x+s_xZ, | ( 3.6) |

где Z — выборка случайных чисел, распределенных по стандартному *нормальному закону*, т. е. когда m = 0, s = 1.

**4. Формирование выборки случайных чисел, соответствующей нормальному распределению, с помощью центральной предельной теоремы**

Известно [[20]](https://www.intuit.ru/studies/professional_skill_improvements/1602/courses/479/literature#literature.20), что сумма n одинаково распределенных независимых случайных величин стремится к нормально распределенной величине при бесконечном увеличении n. Например, можно воспользоваться случайными числами, равномерно распределенными в интервале [0; 1]. В соответствии с центральной предельной теоремой сформированное случайное число является асимптотически нормальной величиной со средним m = n/2 и дисперсией s^2 = n/12. В случае, когда используется какое-либо другое распределение с заданным математическим ожиданием m_x и средним квадратическим отклонением s_x, расчетное математическое ожидание будет равно m = n\cdot m_x и среднее квадратическое отклонение s = s_x\cdot n^{1/2}.

Когда для формирования нормально распределенных случайных чисел суммируются равномерно распределенные случайные величины из интервала [0; 1], то приемлемые результаты в смысле параметров стандартного *нормального закона* дает формула

|  |  |
| --- | --- |
| X=\sum\limits_{i=1}^{12}r_i-6, | ( 3.7) |

где r — равномерно распределенное случайное число из интервала [0; 1].

*Пример 4*. Сформируйте 5000 нормально распределенных чисел в соответствии с центральной предельной теоремой на основе суммирования случайных чисел, равномерно распределенных в интервале [0, 1].

Программный код решения примера:

clear, clc

N = 5000;

for J = 1 : N

X(J) = sum(rand(12,1))-6;

end

X = sort(X);

m = mean(X);

s = std(X);

fprintf('\n\t m = %g\n', m)

fprintf('\t s = %g', s)

*Задание 4*

1. Постройте функции плотности и распределения для случайных величин, сформированных по вышеприведенной программе.
2. Получите выборку случайных чисел со стандартным нормальным распределением при суммировании случайных чисел, распределенных:
   * по экспоненциальному закону,
   * по закону Эрланга 3-го порядка.

При необходимости произведите центрирование полученных случайных величин.

**5. Моделирование нормального закона распределения в системе GPSS/PC**

В системе *GPSS*/PC, как и в других версиях среды *GPSS*, необходимо обеспечить неотрицательность значений интервалов поступления требований (транзактов) в систему и их задержки (обслуживания). Область определения случайной величины для *нормального закона* представляет собой всю *числовую ось*. Поэтому, чтобы обеспечить по возможности только неотрицательные значения случайной величины, необходимо выполнить требование m \ge 5s, где m — математическое ожидание, а s – среднеквадратическое отклонение нормально распределенной случайной величины. Во многих системах программирования имеются генераторы случайных чисел со стандартным *нормальным законом* распределения, для которого m = 0, s = 1. Но переход к *нормальному закону* с другими параметрами можно произвести по формуле (3.6). Поэтому в системах *GPSS* можно использовать стандартный нормальный закон распределения с последующим преобразованием по формуле (3.6).

*Пример 5*. Выполните моделирование одноканальной системы обслуживания, для которой известно, что поступление требований подчиняется равномерному закону из интервала [10; 16] мин., время обслуживания подчиняется *нормальному закону* с параметрами m = 18 мин., s = 3 мин. Моделирование произведите по обслуживанию 1000 требований (транзактов).

Для решения примера сначала сформируем в среде MATLAB функцию стандартного *нормального закона* для области определения от 0 до 30 мин. Затем данные функции распределения и ее аргумента экспортируем в систему *GPSS*/PC.

MATLAB-программа формирования массива данных для стандартной функции нормального распределения:

X = linspace(-3,3.2, 49);

F = normcdf(X);

%------ Преобразование нормального закона -------------

m = 18; s = 3;

xx = m + s\*X;

F2 = [0,F];

X2 = [0, xx];

%%% Заголовок и имя функции для GPSS/PC

fid = fopen('norm.txt', 'w'); %% запись в текстовый файл

fprintf(fid, 'nor function RN6,C50\r\n')

%%% Цикл форматированного вывода данных для GPSS/PC

for J = 1 : length(X2)

if mod(J, 3)

fprintf(fid, '%g,%g/', F2(J), X2(J));

else

fprintf(fid, '\r\n');

fprintf(fid, '%g,%g/', F2(J), X2(J));

end

end

fprintf(fid,';\r\n------------------------------\r\n');

fclose(fid);

Получаемые результаты здесь не приводятся, они будут использованы в *GPSS*-программе после их экспортирования.

*Примечание*. В системе *GPSS*/PC массив данных для функции распределения должен начинаться с нулевых значений, поэтому в программе предусмотрено прибавление нулей к формируемому массиву.

Программный код решения примера в *GPSS*/PC:

simulate

nor function RN6,C50

0,0/0.0013499,9/

0.00204696,9.3875/0.00305642,9.775/0.00449413,10.1625/

0.00650796,10.55/0.00928214,10.9375/0.0130406,11.325/

0.0180485,11.7125/0.0246108,12.1/0.0330681,12.4875/

0.0437873,12.875/0.0571489,13.2625/0.0735293,13.65/

0.0932785,14.0375/0.116696,14.425/0.144004,14.8125/

0.175324,15.2/0.21065,15.5875/0.249838,15.975/

0.29259,16.3625/0.338461,16.75/0.386865,17.1375/

0.437097,17.525/0.488366,17.9125/0.539828,18.3/

0.59063,18.6875/0.639953,19.075/0.687048,19.4625/

0.731273,19.85/0.772116,20.2375/0.809213,20.625/

0.842351,21.0125/0.871463,21.4/0.896616,21.7875/

0.917988,22.175/0.935849,22.5625/0.950529,22.95/

0.962394,23.3375/0.971826,23.725/0.9792,24.1125/

0.98487,24.5/0.989157,24.8875/0.992346,25.275/

0.994678,25.6625/0.996355,26.05/0.997542,26.4375/

0.998368,26.825/0.998933,27.2125/0.999313,27.6/

;------------------------------

tab1 table mp1,5,2,15

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

10 generate 13,3

20 seize 1

30 mark 1

40 advance fn$nor

50 tabulate tab1

60 release 1

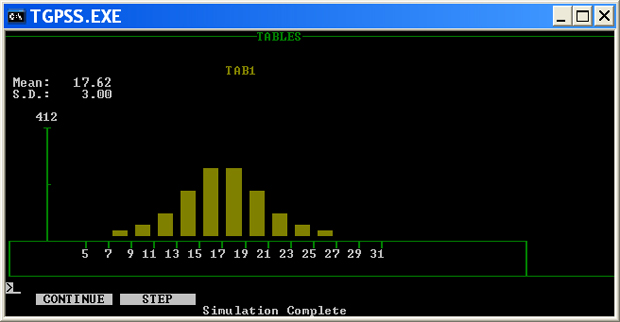
70 terminate 1

start 1000

;end

В программе предусмотрено табулирование времени пребывания транзактов в устройстве обслуживания по стандартному числовому атрибуту mp1, где цифра 1 относится к номеру параметра, через который производится отметка времени блока mark.

На [рис. 3.1](https://www.intuit.ru/studies/professional_skill_improvements/1602/courses/479/lecture/21084?page=2#image.3.1) приведена гистограмма распределения времени обслуживания. Эта гистограмма может быть получена после запуска программы на исполнение при закомментированном операторе end, и далее с помощью комбинации клавиш Alt+T можно перейти в окно просмотра таблиц.



**Рис. 3.1.**Гистограмма распределения времени обслуживания

Как видно из [рис. 3.1](https://www.intuit.ru/studies/professional_skill_improvements/1602/courses/479/lecture/21084?page=2#image.3.1), гистограмма имеет форму, близкую к функции плотности *нормального закона*. Среднее значение времени распределения обслуживания требований в устройстве равно 17.62 ( Mean: 17.62 ), *стандартное отклонение* равно 3.00 ( S.D.: 3.00 ), что достаточно близко к условию примера.

*Задание 5*

1. По файлу стандартного отчета постройте функцию распределения времени обработки транзактов в устройстве. Сравните с той же функцией, построенной в среде MATLAB.
2. Проанализируйте результаты для разных номеров генераторов случайных чисел в зависимости от номера компьютера: RN1, RN2, … .
3. Проанализируйте результаты при различных значениях счетчика завершений, т. е. положите start\ 200, start\ 300 и т. д. в зависимости от номера компьютера, за которым выполняется лабораторная работа (1, 2, 3, ...). Графический образ получаемой гистограммы внесите в отчет лабораторной работы.

**6. Формирование выборки случайных чисел, распределенных по закону эрланга k-го порядка**

Случайная величина, подчиняющаяся распределению Эрланга с параметрами k (порядок распределения Эрланга) и \lambda, представляет собой сумму n независимых, одинаково распределенных случайных величин с параметром \lambda (интенсивность, с^{–1} ) [[20]](https://www.intuit.ru/studies/professional_skill_improvements/1602/courses/479/literature#literature.20). Это означает, что можно получить выборку, соответствующую распределению Эрланга, при помощи многократного применения метода инверсии (метода обратной функции).

*Пример 6*. Сформируйте выборку объема N = 1000 случайных чисел Т с распределением Эрланга 3-го порядка ( k = 3 ) и параметром \lambda = 1.25.

Программный код решения примера:

L = 1.25; k = 3;

N = 1000;

for I = 1 : N

T(I) = 0;

for J = 1 : k

T(I) = T(I) - 1/L\*log(rand);

end

end

T2 = sort(T);

*Задание 6*

1. Упростите программный код, используя матричные возможности системы MATLAB.
2. Постройте функцию плотности и функцию распределения по аналитическим формулам для распределения Эрланга [[3]](https://www.intuit.ru/studies/professional_skill_improvements/1602/courses/479/literature#literature.3):



1. Выполните предыдущий пункт с помощью встроенных функций gampdf и gamcdf. Сравнить результаты.

**7. Моделирование процесса обслуживания по закону эрланга k-го порядка в системе GPSS/PC**

Для моделирования распределения Эрланга k -го порядка используется экспоненциальная функция распределения, которая может оставаться неизменной при изменении порядка потока Эрланга.

*Пример 7*. Смоделируйте одноканальную систему массового обслуживания, в которую требования поступают по равномерному закону через 18 \pm 2 мин., а обслуживание осуществляется по закону Эрланга 3-го порядка с параметром \ lambda = 0.25. Произведите обслуживание 5600 требований.

Для решения примера 7 сначала сформируем функцию экспоненциального распределения случайной величины с параметром \lambda = 1, а затем выполним пересчет для параметра \lambda = 0.25. Для этого может быть использована следующая программа в системе MATLAB:

clear, clc

L = 0.25;

t = linspace(0, 9, 50);

F = 1 - exp(-t);

t2 = 1/L\*t;

fid = fopen('erlang.txt', 'w'); %% запись в текстовый файл

fprintf(fid, '\nexp function RN22,C50\n');

for J = 1 : length(F)

if mod(J,3)

fprintf(fid, '%g,%g/', F(J), t2(J));

else

fprintf(fid, '\r\n');

fprintf(fid, '%g,%g/', F(J), t2(J));

end

end

fprintf(fid,'\r\n;------------------------------\r\n');

fclose(fid);

Программный код решения примера в системе *GPSS*/PC:

simulate

exp function RN22,C50

0,0/0.167792,0.734694/

0.307431,1.46939/0.423639,2.20408/0.520348,2.93878/

0.60083,3.67347/0.667808,4.40816/0.723547,5.14286/

0.769934,5.87755/0.808537,6.61224/0.840663,7.34694/

0.867399,8.08163/0.889648,8.81633/0.908164,9.55102/

0.923574,10.2857/0.936397,11.0204/0.947069,11.7551/

0.955951,12.4898/0.963342,13.2245/0.969493,13.9592/

0.974612,14.6939/0.978872,15.4286/0.982417,16.1633/

0.985367,16.898/0.987822,17.6327/0.989866,18.3673/

0.991566,19.102/0.992981,19.8367/0.994159,20.5714/

0.995139,21.3061/0.995955,22.0408/0.996633,22.7755/

0.997198,23.5102/0.997668,24.2449/0.99806,24.9796/

0.998385,25.7143/0.998656,26.449/0.998882,27.1837/

0.999069,27.9184/0.999225,28.6531/0.999355,29.3878/

0.999464,30.1224/0.999554,30.8571/0.999628,31.5918/

0.999691,32.3265/0.999743,33.0612/0.999786,33.7959/

0.999822,34.5306/0.999852,35.2653/0.999877,36/

;------------------------------

tab1 table mp1,0,3,50

\*\*\*\*\*\*\*\* Erlang distribution \*\*\*\*\*\*\*

10 generate 18,2

20 assign 1,1

30 seize 1

40 mark 1

50 advance 1,fn$exp

60 advance 1,fn$exp

70 advance 1,fn$exp

80 tabulate tab1

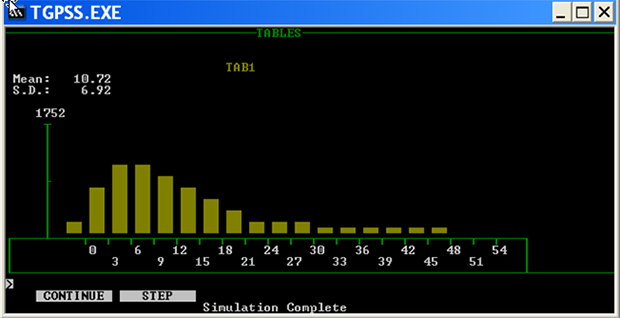
90 release 1

100 terminate 1

start 5600

;end

В программе обслуживание по закону Эрланга 3-го порядка осуществляется с помощью трехкратной временной задержки транзактов на время, распределенное по экспоненциальному закону с заданным параметром. Временная задержка производится блоками advance (строки 50, 60, 70). Контролируемая отметка времени осуществляется блоком mark по 1-му параметру и табулированием стандартного числового атрибута mp1 с помощью оператора table (с меткой tab1). Таким образом, реализация обслуживания транзактов (требований) по закону Эрланга k -го порядка может быть запрограммирована с помощью k блоков advance в теле устройства, т. е. между блоками seize и release. Результат выполнения программы показан на [рис. 3.2](https://www.intuit.ru/studies/professional_skill_improvements/1602/courses/479/lecture/21084?page=3#image.3.2), который дает наглядное представление о форме распределения времени обслуживания, и в нем приводятся числовые данные распределения – среднее значение, равное 10.72, и *стандартное отклонение*, равное 6.92.



**Рис. 3.2.**Распределение времени обслуживания по закону Эрланга

Рассчитаем величину параметра закона Эрланга 3-го порядка по данным рис. 2 или из файла стандартного отчета. Как известно [[2]](https://www.intuit.ru/studies/professional_skill_improvements/1602/courses/479/literature#literature.2), математическое ожидание и параметр закона Эрланга k -го порядка связаны между собой формулой

M=\frac{k}{\lambda} или M=\frac{k+1}{\lambda}

Применим первую формулу, считая, что среднее М равно 10.72:

\lambda=\frac{k}{M}=\frac{3}{10.72}=0,279851\approx 0.28

Получен результат одного порядка по сравнению с заданным условием.

*Задание 7*

1. По файлу стандартного отчета постройте в MATLAB функцию распределения времени обслуживания транзактов в устройстве.
2. В соответствии с номером компьютера, за которым выполняется лабораторная работа, примите следующие числа обработки транзактов (оператор start ): № 1: 1000; № 2: 2000 и т. д. Проанализируйте результаты, рассчитайте параметр закона Эрланга 3-го порядка.

**Контрольные вопросы**

1. Что из себя представляет график функции плотности равномерного закона распределения случайных величин?
2. Как формируется поток Эрланга k -го порядка?
3. Чему соответствует область определения функции распределения случайных величин экспоненциального закона?
4. Чему соответствует область определения функции распределения случайных величин равномерного закона?
5. Какая связь между стандартным отклонением и дисперсией случайных величин?
6. Что собой отображает гистограмма случайных величин?
7. Какая связь между функцией плотности и функцией распределения случайных непрерывных величин?
8. Что является параметром в экспоненциальном законе распределения случайных величин? Какой функциональный смысл имеет параметр экспоненциального распределения случайной величины?
9. Назовите допустимую область определения функции распределения потока Эрланга k -го порядка.