МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РОССИЙСКОЙ ФЕДЕРАЦИИ
федеральное государственное автономное образовательное учреждение высшего образования
«САНКТ-ПЕТЕРБУРГСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ АЭРОКОСМИЧЕСКОГО ПРИБОРОСТРОЕНИЯ»

КАФЕДРА № 52

ОТЧЕТ
ЗАЩИЩЕН С ОЦЕНКОЙ

ПРЕПОДАВАТЕЛЬ

кандидат технических наук, доцент		Н.В. Марковская
должность, уч. степень, звание	подпись, дата	инициалы, фамилия

ОТЧЕТ О ЛАБОРАТОРНОЙ РАБОТЕ № 3

ИССЛЕДОВАНИЕ ИНТЕНСИВОНСТИ ОТКАЗОВ ДЛЯ НЕВОССТАНАВЛИВАЕМЫХ СИСТЕМ

по курсу: НАДЕЖНОСТЬ ИНФОКОММУНИКАЦИОННЫХ СИСТЕМ

РАБОТУ ВЫПОЛНИЛ

СТУДЕНТ ГР. №	5912		Д.Р. Андаев
		подпись, дата	инициалы, фамилия

Санкт-Петербург 2022

1. Цель работы:

Промоделировать процесс функционирования невосстанавливаемой системы.

2. Моделирование первого периода жизни (период приработки):

Для первого периода жизни рассмотрим n = 30000 систем, k = 2 подмножества, первая интенсивность отказов , вторая интенсивность отказа , вероятность попадания в первое множество , вероятность попадания во второе множество .

Описание моделирования: множество n разбиваем на k подмножества. Попадание экземпляра во множество j характеризуется вероятностью . При этом выполняется следующее равенство: . Подмножество j характеризуется своей интенсивностью отказов . Предполагается, что каждый экземпляр системы состоит из единственного элемента и в течение всего времени функционирования системы (вплоть до момента отказа), относящейся к подмножеству j, интенсивность отказов остается величиной постоянной.

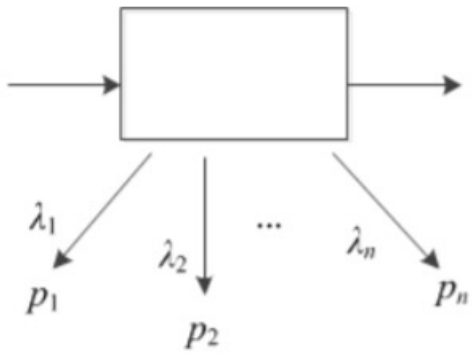

Рисунок 1. Схема периода приработки.

Для i-ой системы первоначально надо определить, к какому подмножеству она относится. Для этого используется распределение . После того, как номер j помножества определен, необходимо сгенерировать значение , как случайной величины, распределенной по экспоненициальному закону с параметром , который определеяется по номеру подмножества.

Для построения графиков R(t) и воспользовались формулами:

,

,

где – это число систем, остающихся работоспособными в момент времени t и соответственно, , – рассматриваемый интервал.

Кроме того, построили теоретическе графики R(t) и при помощи выведенных формул:

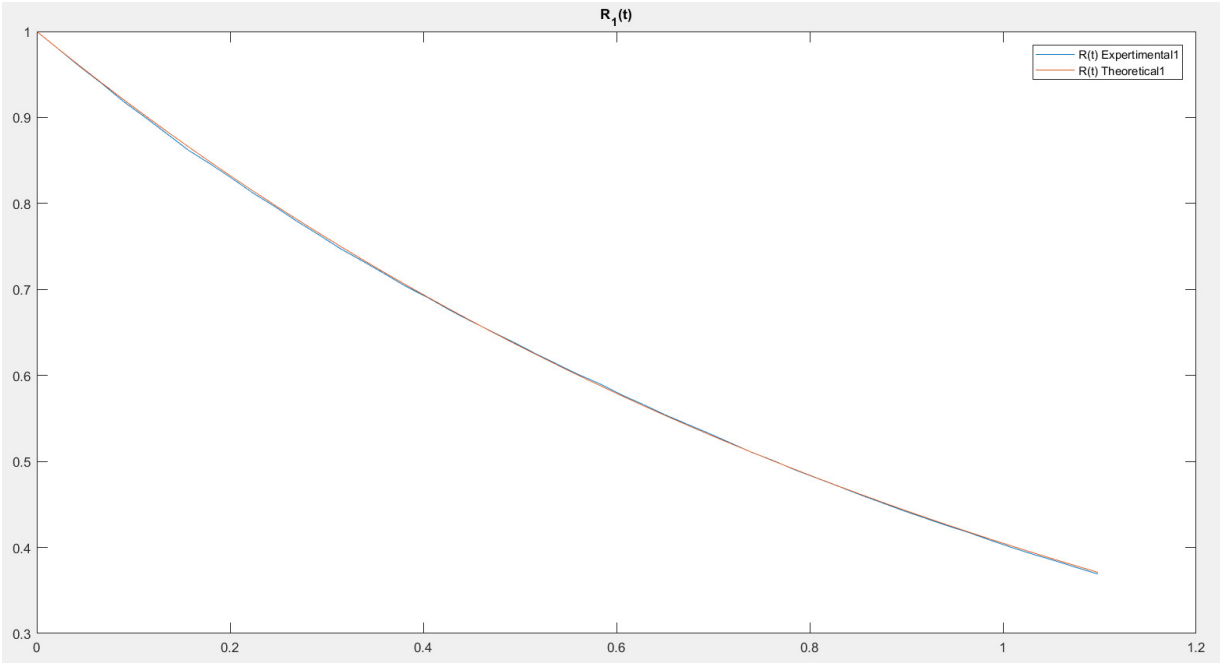

Рисунок 2. Графики R(t) для первого периода жизни.

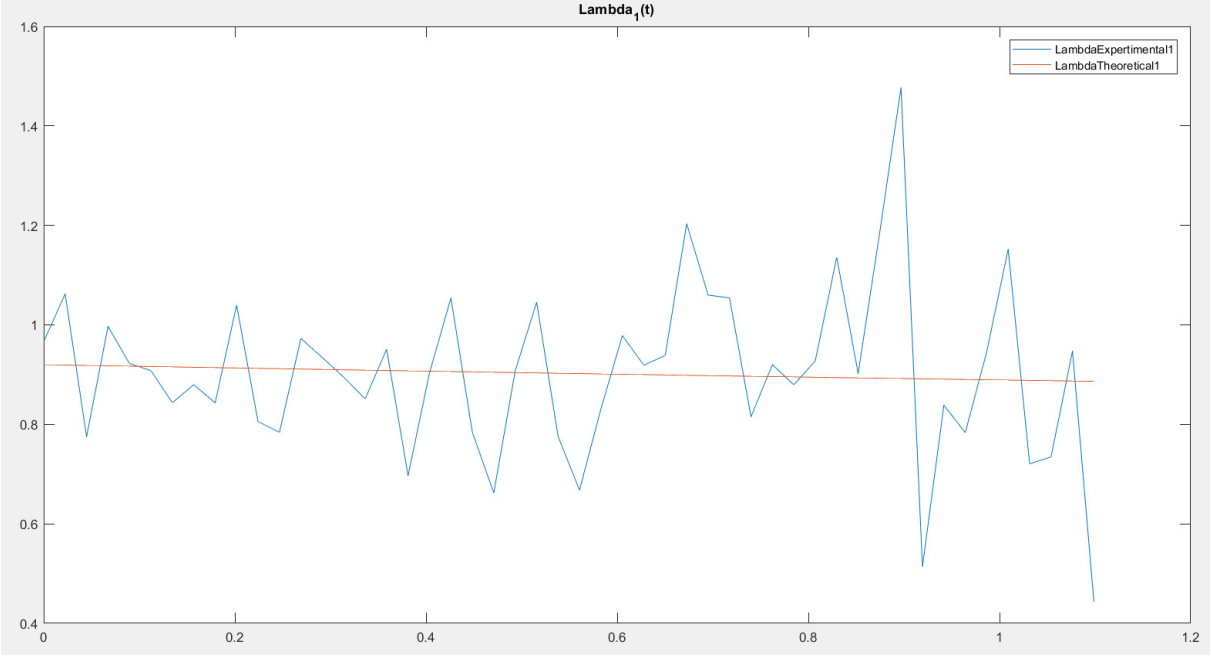

Рисунок 3. Графики (t) для первого периода жизни при .

Полученный график показывает, что функция изменения интенсивности отказов в первый период жизни – убывающая функция. При близких друг к другу значениях первое и второе подмножества систем будут иметь довольно близкое время безотказной работы. Если увеличивать разницу этих величин, то одно множество будет иметь меньшее время безотказной работы, чем другое. А так как мы рассматриваем промежуток, равный среднему времени работы всех систем, то функция изменения отказов будет убывать, так как системы из одного из множеств закончат свою работу намного раньше, чем системы из другого.

Также графики экспериментального и теоретиечского расчетов величины R(t) совпали, а экспериментальный график колеблется около теоретического.

3. Моделирование второго периода жизни (период нормального функционирования):

Для второго периода жизни рассмотрим m = 30000 систем, n = 2 последовательно соединенных элемента в системе, первая интенсивность отказов , вторая интенсивность отказа .

Описание моделирования: в качестве модели системы используется схема с последовательным соединением n элементов. Элемент i характеризуется интенсивностью отказов , которая является постоянной величиной.

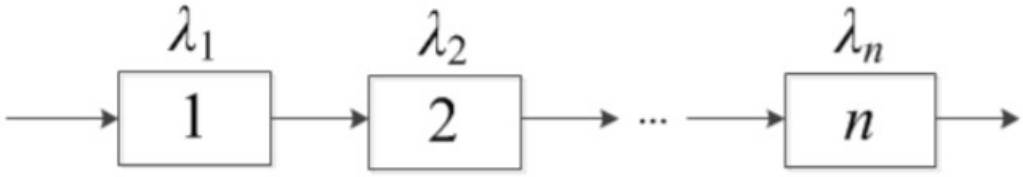

Рисунок 4. Схема периода нормального функционирования.

Пусть исследуемая система состоит из n элементов. Для j-ого элемента системы вычисляется время безотказной работы , как случайной величины, распределенной по экспоненциальному закону с параметром . Значение времени безотказной работ i-ой системы вычисляется по следующей формуле:

.

Для построения графиков R(t) и воспользовались формулами:

,

,

где – это число систем, остающихся

работоспособными в момент времени t и соответственно, ,
– рассматриваемый интервал.

Кроме того, построили теоретическе графики R(t) и при помощи выведенных формул:

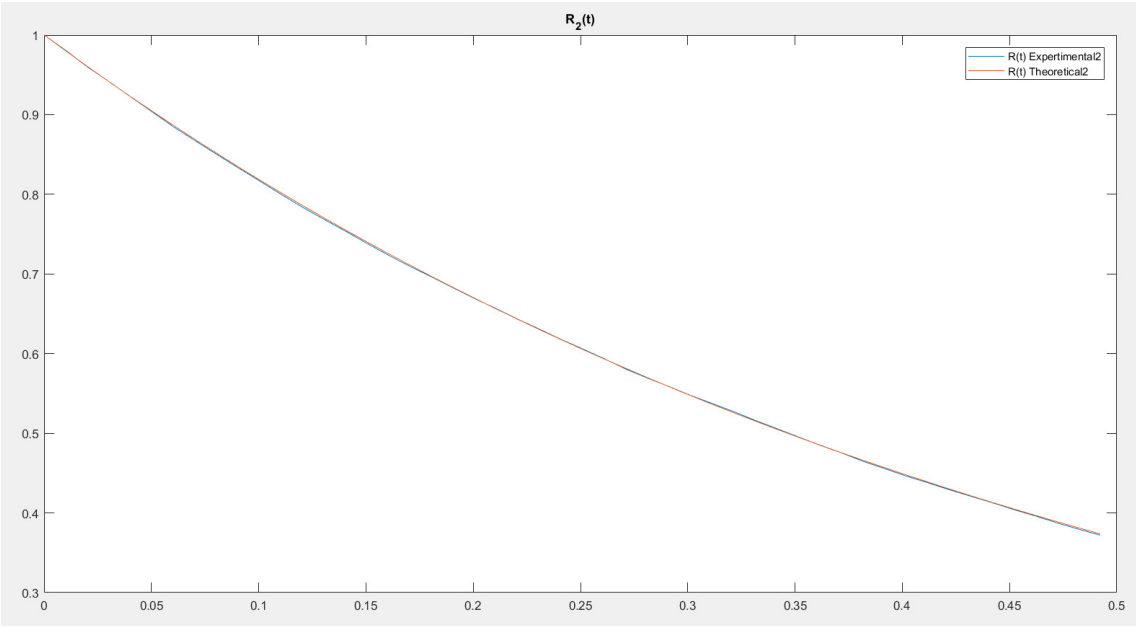

Рисунок 5. Графики R(t) для второго периода жизни.

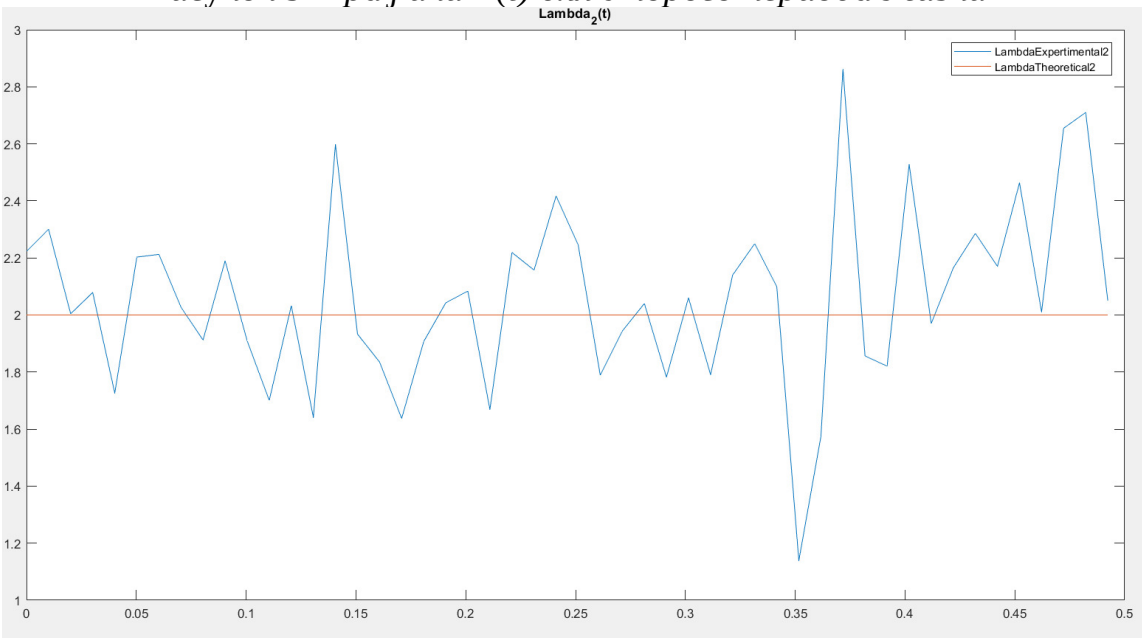

Рисунок 6. Графики (t) для второго периода жизни при .

Исходя из первого графика, можно сделать вывод о том, что функция надежности – невозрастающая функция. Исходя из второго графика функция изменения интенсивности отказов во второй период жизни является константой и высчитывается по формуле

Также графики экспериментального и теоретиечского расчетов величины R(t) совпали, а экспериментальный график колеблется около теоретического.

4. Моделирование третьего периода жизни (период старения):

Для третьего периода жизни рассмотрим m = 30000 систем, n = 2 параллельно соединенных элемента в системе, первая интенсивность отказов , вторая интенсивность отказа .

Описание моделирования: в качестве модели системы используется схема с параллельным соединением n элементов. Элемент i характеризуется интенсивностью отказов , которая является постоянной величиной.

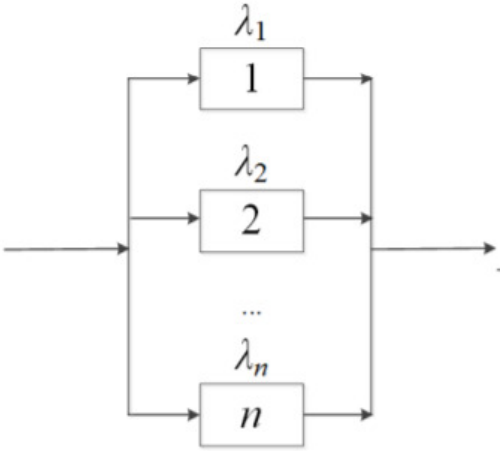

Рисунок 7. Схема периода старения.

Пусть исследуемая система состоит из n элементов. Для i-ого элемента системы вычисляется время безотказной работы , как случайной величины, распределенной по экспоненциальному закону с параметром . Значение времени безотказной работ i-ой системы вычисляется по следующей формуле:

.

Для построения графиков R(t) и воспользовались формулами:

,

,

где – это число систем, остающихся

работоспособными в момент времени t и соответственно, ,
– рассматриваемый интервал.

Кроме того, построили теоретическе графики R(t) и при помощи выведенных формул:

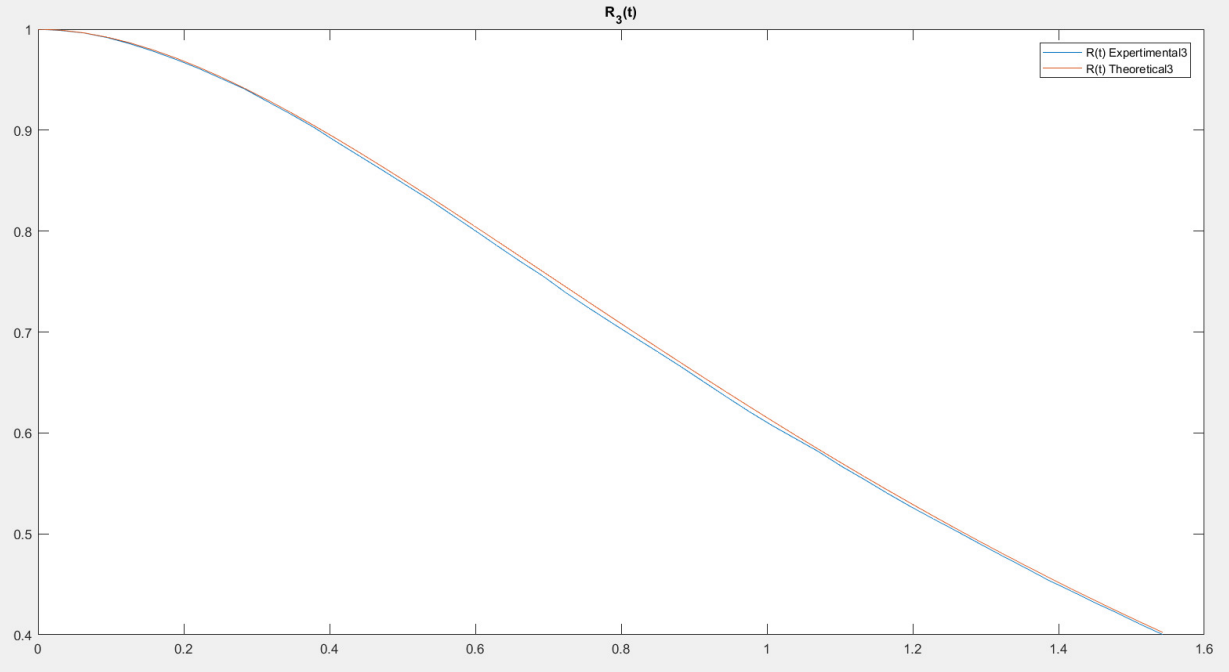

Рисунок 8. Графики R(t) для третьего периода жизни.

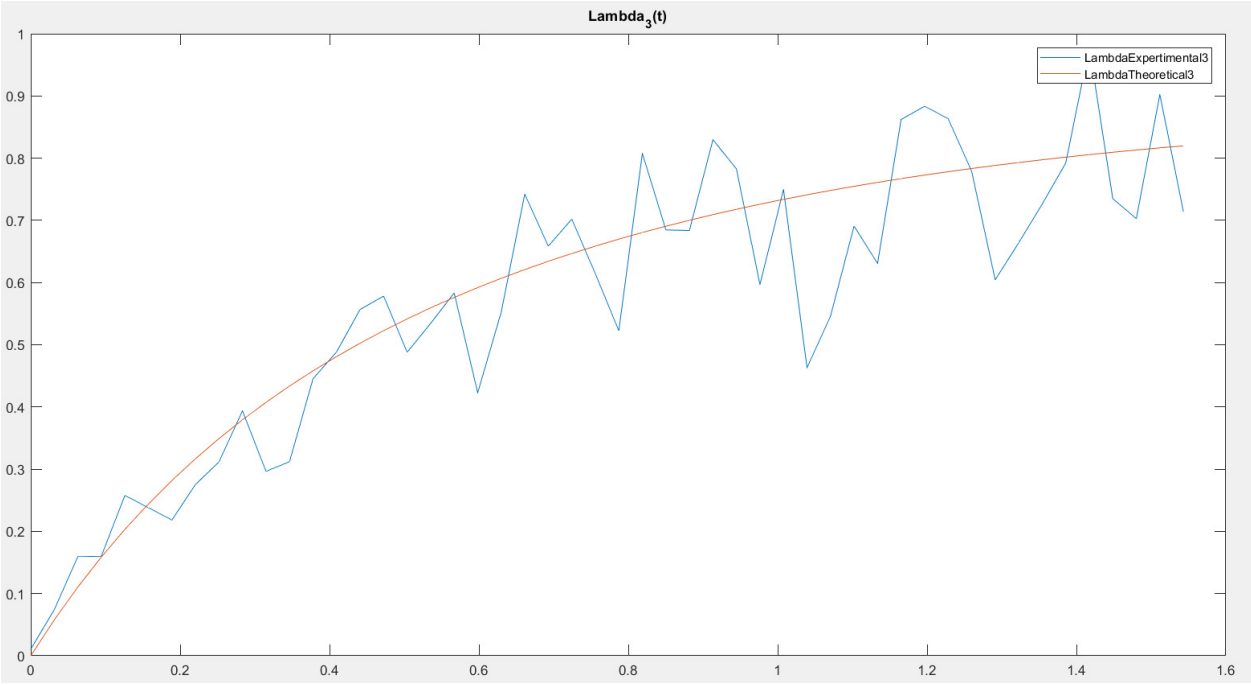

Рисунок 9. Графики (t) для третьего периода жизни при .

Исходя из первого графика, можно сделать вывод о том, что функция надежности – невозрастающая функция. Исходя из второго графика функция изменения интенсивности отказов в третий период жизни является возрастающей, так как со временем число систем, вышедших из строя, увеличивается.

Также графики экспериментального и теоретиечского расчетов величины R(t) совпали, а экспериментальный график колеблется около теоретического.

5. Вывод.

В ходе выполнения работы, промоделировали работу трех этапов жизни невосстанавливаемых систем. Из полученных графиков функции изменения интенсивности можно сделать следующие выводы:

- В начале этапа приработки многие системы выходят из строя по различным причинам (изначальные ошибки в системах). С приближением к второму периоду интенсивность выхода систем из строя уменьшается
- В определенный момент интенсивность выхода систем из строя приходит к постоянному значению и начинается период нормального функционирования. В это время основными причинами отказов систем являются случайные перегрузки.
- Для последнего периода – периода старения – характерно резкое повышение интенсивности отказов, так как наступает предельное состояние работы основной части систем.

6. Приложение с кодом:

Java:

```
package com.company;

import java.io.FileWriter;
import java.io.IOException;
import java.util.ArrayList;
import java.util.List;

import static java.lang.Math.*;

public class Main {

    public static void main(String[] args) {
        int n = 30000;
        int k = 2;
        double l1 = 0.8;
```

```
        double l2 = 1.2;
        double p1 = 0.7;
        double p2 = 0.3;
        ArrayList <Double> Ti1 = new ArrayList<>();
        ArrayList <Double> Ti2 = new ArrayList<>();
        ArrayList <Double> Ti3 = new ArrayList<>();
        ArrayList <Double> lambda = new ArrayList<>();
        ArrayList <Double> Rt = new ArrayList<>();
        ArrayList <Double> RtT = new ArrayList<>();
        ArrayList <Double> lambdaT = new ArrayList<>();

        for (int i = 0; i < (int) (p1*n); i++) {
            Ti1.add(-log(random())/l1);
        }
        for (int i = 0; i < n - (int) (p1*n); i++) {
            Ti1.add(-log(random())/l2);
        }
        double Tm = 0;
        for (int i = 0; i < n; i++) {
            Tm += Ti1.get(i);
        }
        Tm /= n;
        double tStep = Tm/50;
        for (double i = 0; i < Tm; i += tStep) {
            int nt = countNt(Ti1,i);
            int ntdelta = countNt(Ti1,i + tStep * 0.1);
            lambda.add((nt-ntdelta)/(nt * tStep * 0.1));
            Rt.add ( (double) nt/n);
            RtT.add (exp(-l1*i)*p1+exp(-l2*i)*p2);
                 lambdaT.add(-(-l1*exp(-l1*i)*p1-l2 *  exp(-
l2*i)*p2)/(exp(-l1*i)*p1+exp(-l2*i)*p2));
        }
        save(lambda,"lambda1.txt", tStep);
        save(Rt, "Rt1.txt",tStep);
        save(RtT, "Rt1T.txt",tStep);
        save(lambdaT, "lambda1T.txt",tStep);
        lambda.clear();
        Rt.clear();
        RtT.clear();
        lambdaT.clear();


        for (int i = 0; i < n; i++) {
            double T1 = -log(random())/l1;
            double  T2 = -log(random())/l2;
            Ti2.add(min(T1,T2));
```

```
        }
        Tm = 0;
        for (int i = 0; i < n; i++) {
            Tm += Ti2.get(i);
        }
        Tm = Tm / n;
        tStep = Tm/50;
        for (double i = 0; i < Tm; i += tStep) {
            int nt = countNt(Ti2,i);
            int ntdelta = countNt(Ti2,i + tStep * 0.1);
            lambda.add((nt-ntdelta)/(nt * tStep * 0.1));
            Rt.add ( (double) nt/n);
            RtT.add (exp(i*(-l1-l2)));
            lambdaT.add(l1+l2);
        }
        save(lambda,"lambda2.txt", tStep);
        save(Rt, "Rt2.txt",tStep);
        save(RtT, "Rt2T.txt",tStep);
        save(lambdaT, "lambda2T.txt",tStep);
        lambda.clear();
        Rt.clear();
        RtT.clear();
        lambdaT.clear();


        for (int i = 0; i < n; i++) {
            double T1 = -log(random())/l1;
            double  T2 = -log(random())/l2;
            Ti3.add(max(T1,T2));
        }
        Tm = 0;
        for (int i = 0; i < n; i++) {
            Tm += Ti3.get(i);
        }
        Tm = Tm / n;
        tStep = Tm/50;
        for (double i = 0; i < Tm; i += tStep) {
            int nt = countNt(Ti3,i);
            int ntdelta = countNt(Ti3,i + tStep * 0.1);
            lambda.add((nt-ntdelta)/(nt * tStep * 0.1));
            Rt.add ( (double) nt/n);
             RtT.add (exp(-l1*i)+exp(-l2*i)-exp(-l1*i)*exp(-
l2*i));
                        lambdaT.add(-(-l1*exp(-l1*i)-l2*exp(-
l2*i)+exp(-l1*i-l2*i)*      (l1+l2))/(exp(-l1*i)+exp(-l2*i)-
exp(-l1*i)*exp(-l2*i)));
```

```
        }
        save(lambda,"lambda3.txt", tStep);
        save(Rt, "Rt3.txt",tStep);
        save(RtT, "Rt3T.txt",tStep);
        save(lambdaT, "lambda3T.txt",tStep);
    }

    public static int countNt (ArrayList<Double> Ti, double
t) {
        int count = 0;
        for (double i = 0; i < Ti.size(); i++) {
            if (t < Ti.get( (int) i)) {
                count++;
            }
        }
        return count;
    }

       public static void save(List<Double> list, String
filepath, double step) {
        try {
                                FileWriter fileWriter = new
FileWriter(filepath);
            for (int i = 0; i < list.size(); ++i) {
                        fileWriter.write(i * step + "    " +
list.get(i) +"\n");
                fileWriter.flush();
            }
        } catch (IOException exception) {
            exception.printStackTrace();
        }
    }
}
```

Matlab:

```
figure(1)

C = load('Rt1.txt');
x = C(:,1);
y = C(:,2);
plot(x, y);
hold on
B = load('Rt1T.txt');
x = B(:,1);
y = B(:,2);
plot(x, y);
```

```
legend({'R(t) Expertimental1', 'R(t)
Theoretical1'},'Location','northeast')
title('R_1(t)');
hold on

figure(2)

C = load('lambda1.txt');
x = C(:,1);
y = C(:,2);
plot(x, y);
hold on
B = load('lambda1T.txt');
x = B(:,1);
y = B(:,2);
plot(x, y);
legend({'LambdaExpertimental1',
'LambdaTheoretical1'},'Location','northeast')
title('Lambda_1(t)');
hold on

figure(3)

C = load('Rt2.txt');
x = C(:,1);
y = C(:,2);
plot(x, y);
hold on
B = load('Rt2T.txt');
x = B(:,1);
y = B(:,2);
plot(x, y);
legend({'R(t) Expertimental2', 'R(t)
Theoretical2'},'Location','northeast')
title('R_2(t)');
hold on

figure(4)

C = load('lambda2.txt');
x = C(:,1);
y = C(:,2);
plot(x, y);
hold on
B = load('lambda2T.txt');
x = B(:,1);
y = B(:,2);
plot(x, y);
legend({'LambdaExpertimental2',
```

```
'LambdaTheoretical2'},'Location','northeast')
title('Lambda_2(t)');
hold on

figure(5)

C = load('Rt3.txt');
x = C(:,1);
y = C(:,2);
plot(x, y);
hold on
B = load('Rt3T.txt');
x = B(:,1);
y = B(:,2);
plot(x, y);
legend({'R(t) Expertimental3', 'R(t)
Theoretical3'},'Location','northeast')
title('R_3(t)');
hold on

figure(6)

C = load('lambda3.txt');
x = C(:,1);
y = C(:,2);
plot(x, y);
hold on
B = load('lambda3T.txt');
x = B(:,1);
y = B(:,2);
plot(x, y);
legend({'LambdaExpertimental3',
'LambdaTheoretical3'},'Location','northeast')
title('Lambda_3(t)');
hold on
```