

**ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ АВТОНОМНОЕ
ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ
«Национальный исследовательский университет ИТМО»**

Факультет безопасности информационных технологий

Дисциплина:

«Теория надежности»

Лабораторные работы №1

Вариант 27

Выполнил:

Студент гр. N33472

Шарифов Ф. Р.

Проверил:

Кузнецов А. Ю.

Санкт-Петербург

2021 г.

Цель работы:

- изучение методов определения основных показателей надежности изделий на основе экспериментальных данных;
- разработка программного обеспечения обработки статистических данных по отказам изделий в процессе испытаний на надежность или в процессе их эксплуатации.

Ход работы:

Используемые формулы:

$$\text{Вероятность безотказной работы } p = 1 - \frac{r}{N} \quad (1)$$

$$\text{Частота отказов } a_i = \frac{\Delta r_i}{N * \Delta t} \quad (2)$$

$$\text{Интенсивность отказов } \lambda(t) = \frac{\Delta r_i}{(N - r(i-1)) \Delta t} \quad (3)$$

$$\text{Средняя наработка до отказа: } T = \frac{1}{N} \sum_{k=1}^N t_k \quad (4)$$

Δr_i - количество изделий, отказавших в период Δt_i .

Общее числа отказов наблюдаемых изделий в интервале времени $(0, t_{i-1})$ вычисляется с помощью соотношения вида:

$$r(i-1) = \sum_{j=0}^{i-1} \Delta r(j)$$

В процессе испытаний находилось $N=100$ однотипных изделий, работающих до первого отказа. Моменты времени выхода из строя вследствие наступления отказов всех рассматриваемых изделий обозначим через $t_1, t_2, \dots, t_k, \dots, t_{N-1}, t_N$. Разделим весь диапазон времени безотказной работы всех N изделий на n интервалов времени.

Для определения показателей надежности была написана программа на языке C++, следовательно ниже представлено алгоритм, для расчета показателей надежности:

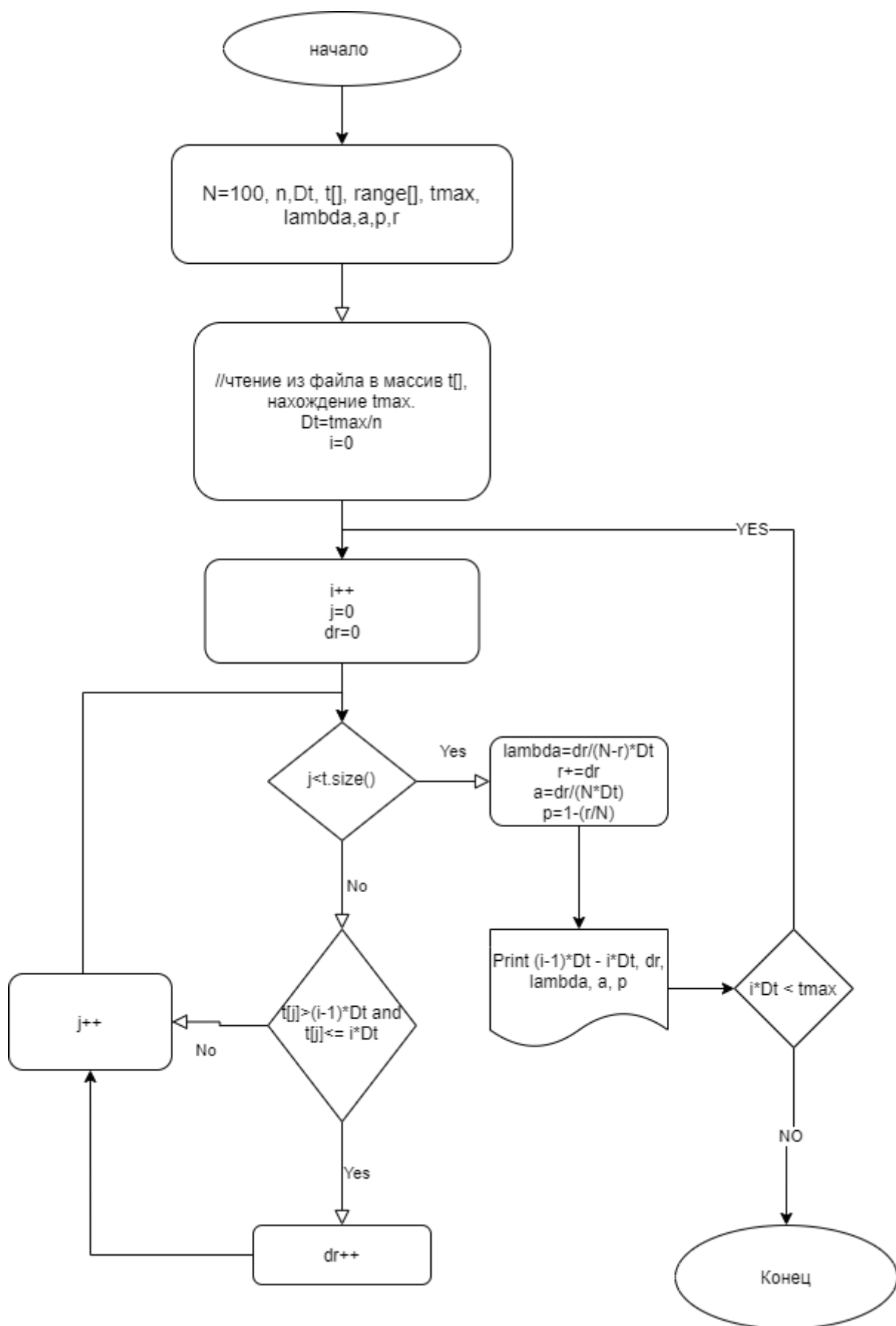


Рисунок 1. Блок-схема

Данный алгоритм, получает на вход файл с данными и n . Далее разделяет весь диапазон времени без отказной работы, на n частей, и вычисляет показатели надежности используя формулы, для всех n диапазонов.

Пример работы программы при n : 5, 10, 20

Интервалы	Кол-во отказов	Интенсивность отказов	Плотность распределения	Функция надежности	Контроль
(t_i, t_{i-1})	Δr_i	λ^*_{i-1}	a^*_{i-1}	p^*_{i-1}	$\lambda^*_{i-1} p^*_{i-1} - a^*_{i-1}$
0 - 203,612	20	0,000982	0,000982	0,8	
203,612 - 407,224	22	0,001351	0,00108	0,58	
407,224 - 610,836	9	0,000762	0,000442	0,49	
610,836 - 814,448	23	0,002305	0,00113	0,26	
814,448 - 1018,06	26	0,004911	0,001277	0	

Таблица 1. При $n = 5$

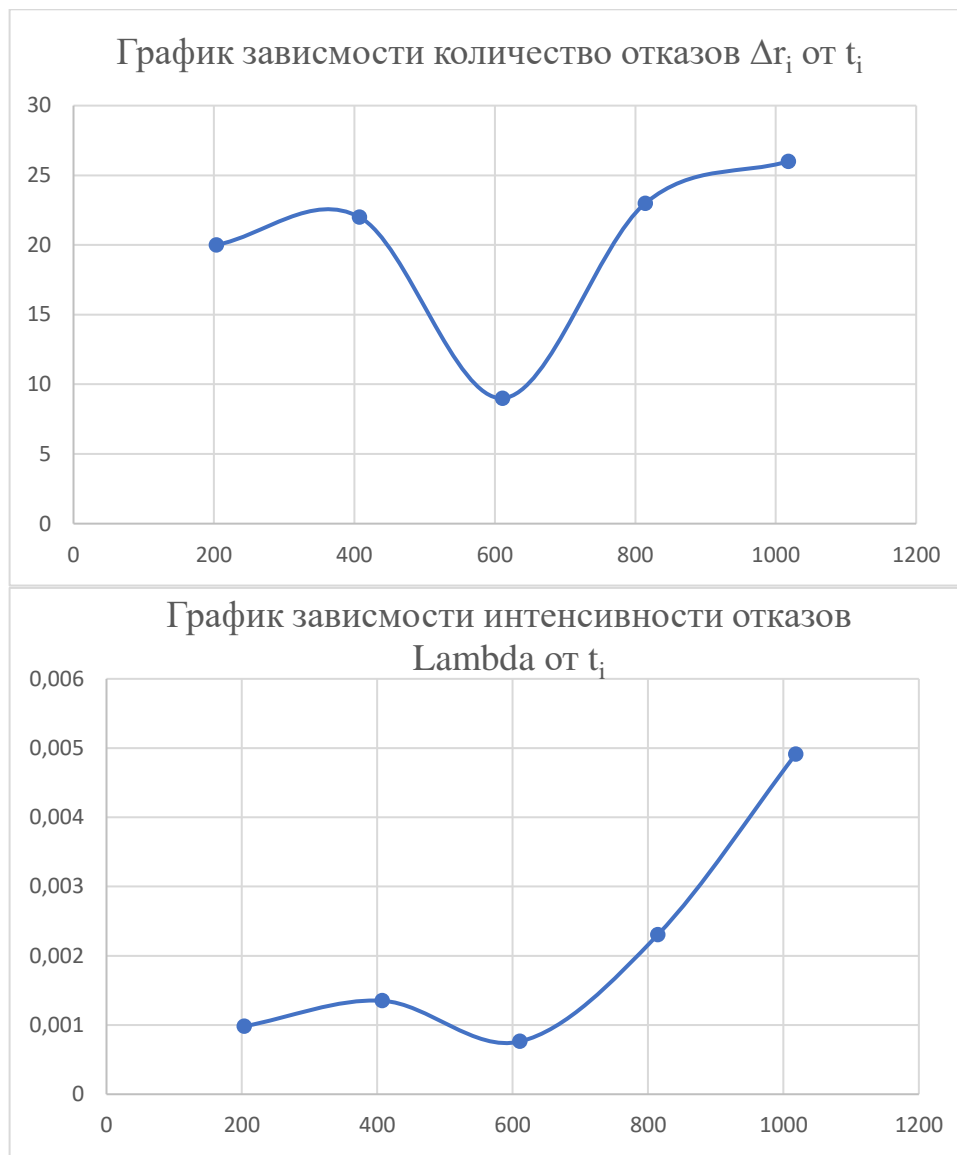
Интервалы	Кол-во отказов	Интенсивность отказов	Плотность распределения	Функция надежности	Контроль
(t_i, t_{i-1})	Δr_i	λ^*_{i-1}	a^*_{i-1}	p^*_{i-1}	$\lambda^*_{i-1} p^*_{i-1} - a^*_{i-1}$
0 - 101,806	5	0,000491	0,000491	0,95	
101,806 - 203,612	15	0,001551	0,001473	0,8	
203,612 - 305,418	12	0,001473	0,001179	0,68	
305,418 - 407,224	10	0,001445	0,000982	0,58	
407,224 - 509,03	3	0,000508	0,000295	0,55	
509,03 - 610,836	6	0,001072	0,000589	0,49	
610,836 - 712,642	15	0,003007	0,001473	0,34	
712,642 - 814,448	8	0,002311	0,000786	0,26	
814,448 - 916,254	17	0,006422	0,00167	0,09	
916,254 - 1018,06	9	0,009823	0,000884	0	

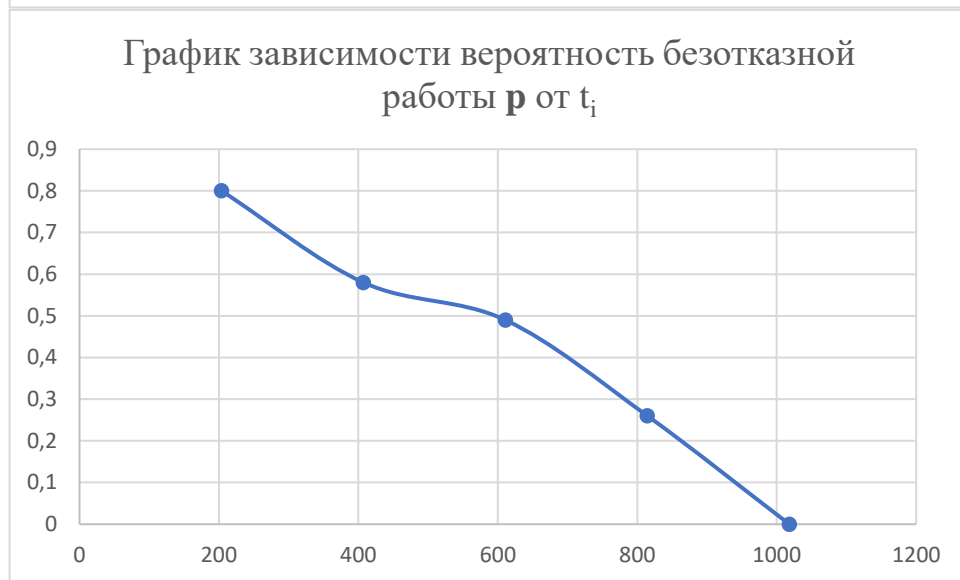
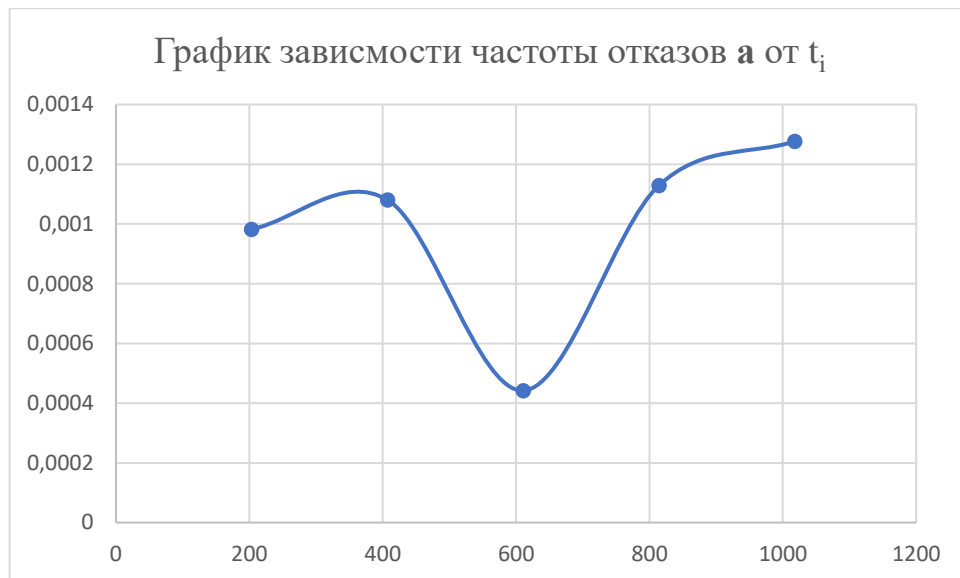
Таблица 2. При $n = 10$

Интервалы	Кол-во отказов	Интенсивность отказов	Плотность распределения	Функция надежности	Контроль
(t_i, t_{i-1})	Δr_i	λ^*_{i-1}	a^*_{i-1}	p^*_{i-1}	$\lambda^*_{i-1} p^*_{i-1} - a^*_{i-1}$
0 - 50,903	0	0	0	1	
50,903 - 101,806	5	0,000982	0,000982	0,95	
101,806 - 152,709	5	0,001034	0,000982	0,9	
152,709 - 203,612	10	0,002183	0,001965	0,8	
203,612 - 254,515	8	0,001965	0,001572	0,72	
254,515 - 305,418	4	0,001091	0,000786	0,68	
305,418 - 356,321	4	0,001156	0,000786	0,64	
356,321 - 407,224	6	0,001842	0,001179	0,58	
407,224 - 458,127	0	0	0	0,58	
458,127 - 509,03	3	0,001016	0,000589	0,55	
509,03 - 559,933	2	0,000714	0,000393	0,53	
559,933 - 610,836	4	0,001483	0,000786	0,49	
610,836 - 661,739	3	0,001203	0,000589	0,46	
661,739 - 712,642	12	0,005125	0,002357	0,34	
712,642 - 763,545	5	0,002889	0,000982	0,29	
763,545 - 814,448	3	0,002032	0,000589	0,26	
814,448 - 865,351	8	0,006045	0,001572	0,18	
865,351 - 916,254	9	0,009823	0,001768	0,09	
916,254 - 967,157	1	0,002183	0,000196	0,08	
967,157 - 1018,06	8	0,019645	0,001572	0	

Таблица 2. При $n = 20$

Графики функции, $\Delta r_i(t_i)$, $\lambda^*_i(t_i)$, $a^*_i(t_i)$, $p^*_i(t_i)$, при $n = 5$





Графики функции, $\Delta r_i(t_i)$, $\lambda^*_i(t_i)$, $a^*_i(t_i)$, $p^*_i(t_i)$, при $n = 10$.



График зависимости интенсивности отказов
 Λ от t_i

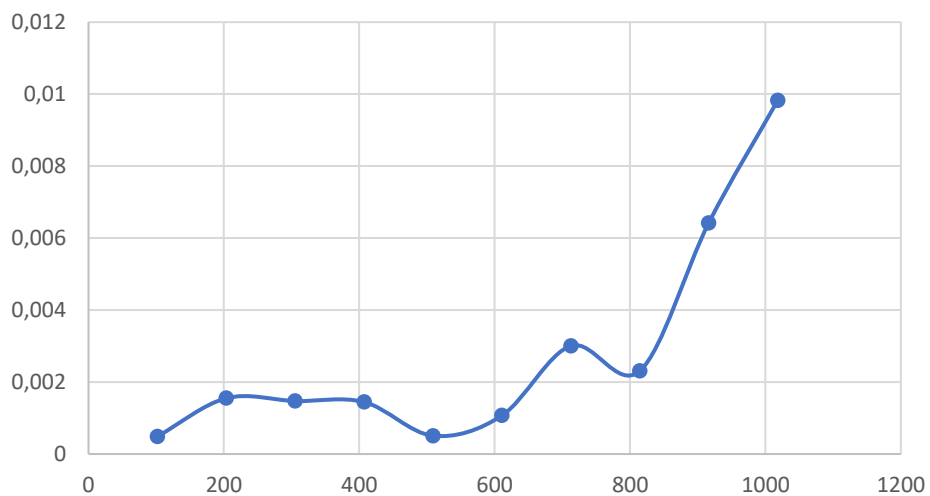


График зависимости частоты отказов a от t_i

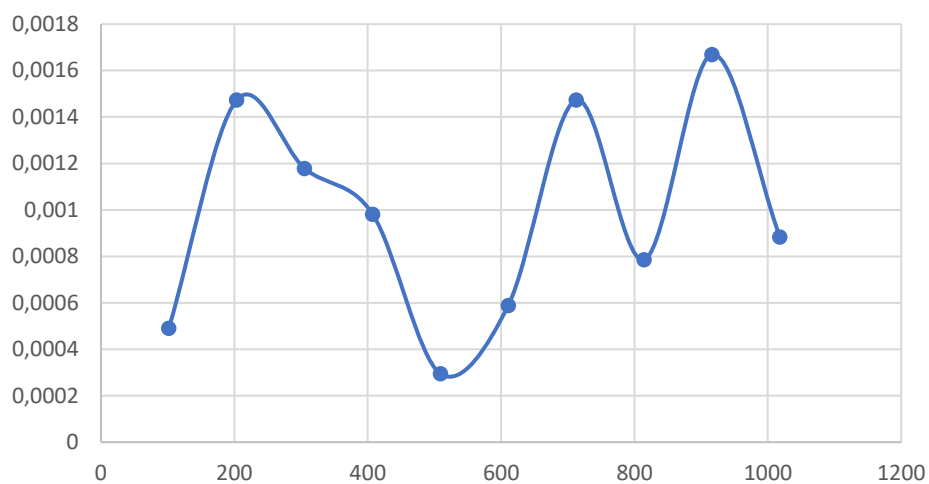
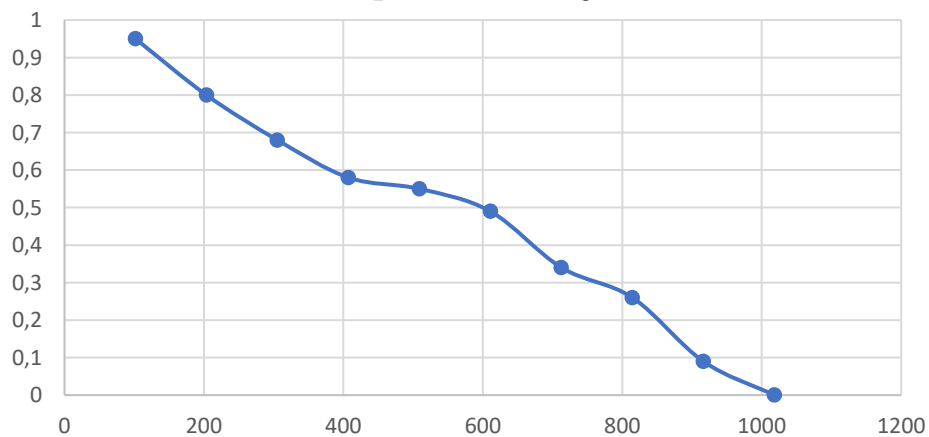
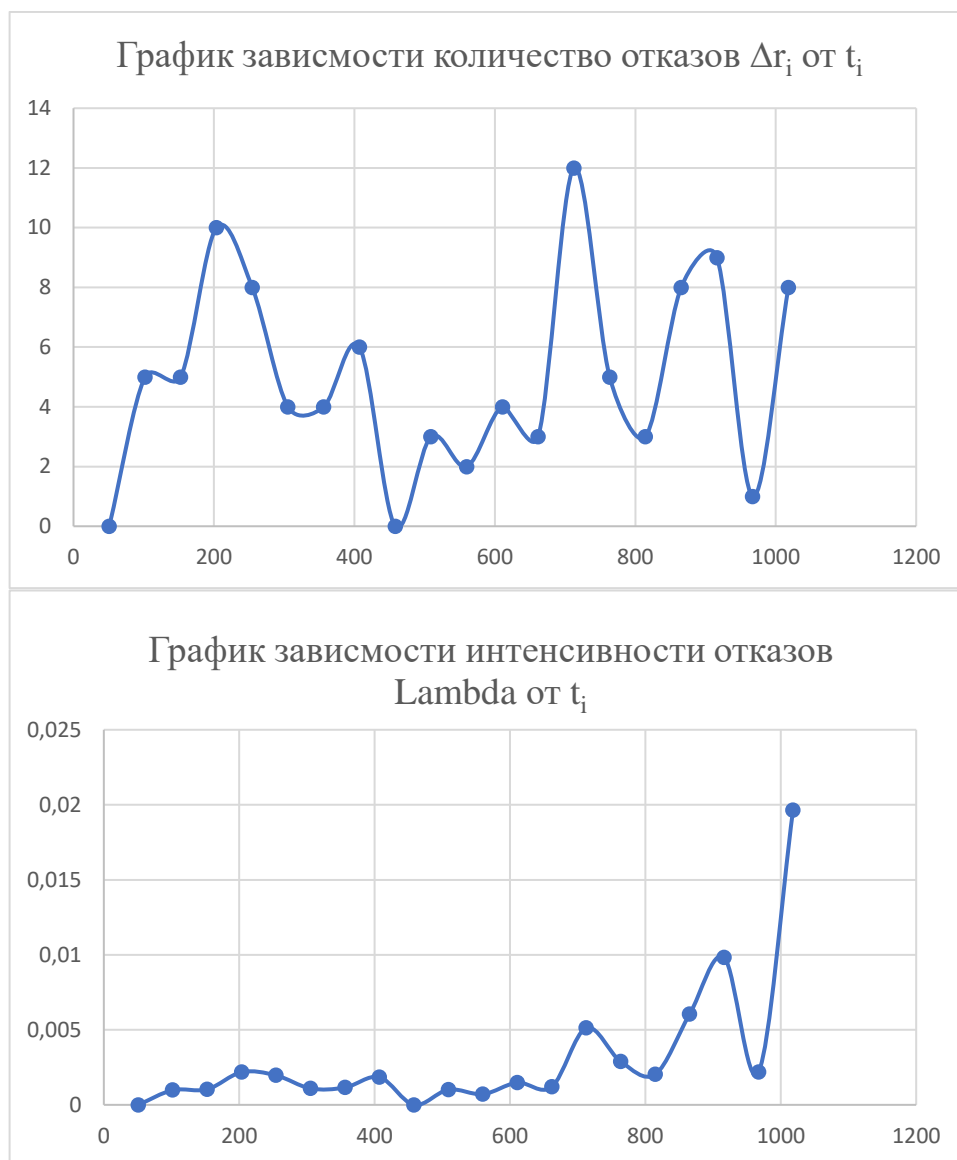
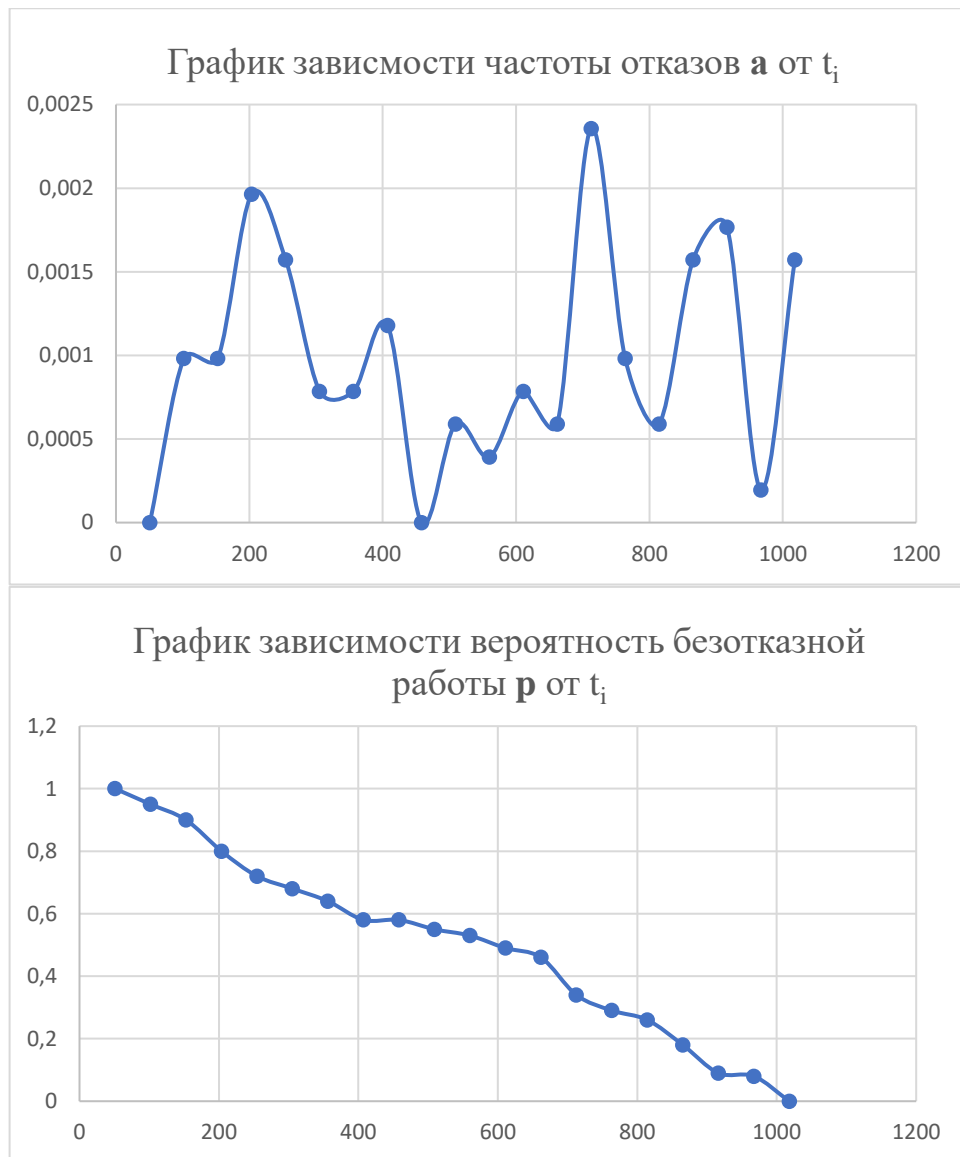


График зависимости вероятность безотказной
работы p от t_i



Графики функции, $\Delta r_i(t_i)$, $\lambda^*_i(t_i)$, $a^*_i(t_i)$, $p^*_i(t_i)$, при $n = 20$





Вывод

В ходе Лабораторной работы были определены основные показатели надежности, такие как частота отказов, вероятность безотказной работы, интенсивность отказов и средняя наработка на отказ. И еще были построены графики четырех функций $\Delta r_i(t_i)$, $\lambda^*_i(t_i)$, $a^*_i(t_i)$, $p^*_i(t_i)$, с различными интервалами, $n=\{5,10,20\}$.

Код программы:

```
#include <iostream>
#include <fstream>
#include <string>
#include <stdlib.h>
```

```

#include <vector>

#include <bits/stdc++.h>


using namespace std;
vector <double> t; // Array for product uptime.
vector <double> range;
double t_max=0.0, Dt, M;
int N=100, n;
// Replacing commas with points
void coma_to_point( string &s)
{
    int k=0;
    int i=0;
    while (k!= -1)
    {
        k=s.find(',');
        if (k!= -1)
            s.replace(k,1, ".");
    }
}

//Adding to array and finding Max element.
void Add_to_Array(string text)
{
    int pos=0;
    text+= " ";
    double summ=0.0;//
    while(pos != -1)
    {
        pos = text.find(" ");
        if (pos != -1)
        {

```

```

        double temp1;
        string temp;
        temp=text.substr(0,pos+1);
        temp1 = atof(temp.c_str());
        t.push_back(temp1);
        summ+=temp1;
        if (temp1 > t_max) t_max = temp1;
        text.replace(0,pos+1,"");
    }
}

```

M=summ/N;// Mean time between failures

```

}
void parsing(string file)
{
    ifstream fff(file.c_str());
    while (!fff.eof())
    {
        string data;
        getline(fff,data);
        coma_to_point(data);
        Add_to_Array(data);

    }
    fff.close();
}
void show(int* mass, int n)
{
    for (int i=0; i<n;i++)
        cout<<mass[i]<<" ";
}

```

```

        cout<<"\n";
    }
void computation()
{
    //range table
    ofstream file("DataInrvl20.csv");
    file <<"ti-ti_1 " <<"Dr " <<"Lambda " <<"*a " <<"*p \n";
    double counter=0.0;
    while (counter< t_max)
    {
        counter+=Dt;
        range.push_back(counter);
    }
    int r=0;
    for (int i=1;i<range.size();i++)
    {
        int dr=0;
        double lambda, a;
        long double p;
        for (int j=0;j<t.size();j++)
        {
            if(t[j]>range[i-1] && t[j]<= range[i])
                dr++;
        }
        lambda= dr/(double(N-r)*Dt);
        r+= dr;
        a=dr/(N*Dt);
        p=1-(double(r)/double(N));
        cout<<"(" <<setw(7)<<range[i-1]<<"-" <<setw(7)<<range[i]<<" )";

cout<<setw(3)<<dr<<setw(15)<<lambda<<setw(15)<<a<<setw(15)<<p<<setw(15)<<lambda*p
-a<<"\n";

        file<<range[i]<<" "; <<dr<<" "; <<lambda<<" "; <<a<<" "; <<p<<"\n";
    }
}

```

```
}
```

```
}
```

```
int main()
```

```
{
```

```
    cout<<"Number of intervals:";
```

```
    cin>>n;
```

```
    parsing("L1_out.txt");
```

```
    range.push_back(0.0);
```

```
    sort(t.begin() , t.end());
```

```
    Dt = t_max/n;
```

```
    cout<<" Dt = "<<Dt<<"\n";
```

```
    computation();
```

```
}
```