## МИНОБРНАУКИ РОССИИ

**САНКТ-ПЕТЕРБУРГСКИЙ ГОСУДАРСТВЕННЫЙ ЭЛЕКТРОТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ**

## «ЛЭТИ» ИМ. В.И. УЛЬЯНОВА (ЛЕНИНА)

**Кафедра МО ЭВМ**

## ОТЧЕТ

**по лабораторной работе №5**

## по дисциплине «Качество и метрология программного обеспечения»

**Тема: «Оценка параметров надежности программ по временным моделям обнаружения ошибок»**

Студентка гр. 8304 Мельникова О.А.

Преподаватель Ефремов М.А.

Санкт-Петербург 2022

## Задание

Выполнить исследование показателей надежности программ, характеризуемых моделью обнаружения ошибок Джелинского-Моранды, для различных законов распределения времен обнаружения отказов и различного числа используемых для анализа данных. Для проведения исследования требуется:

1. Сгенерировать массивы данных {Xi}, где Xi – случайное значение *интервала между соседними (i-1)–ой и i–ой ошибками* ( i=[1,30], также смотри примечание в п.3), в соответствии с:

А) равномерным законом распределения в интервале [0,20]; при этом cредний интервал между ошибками будет mравн = 10, СКО sравн = 20/(2\*sqrt(3)) = 5.8 .

Б) экспоненциальным законом распределения

W(y) = b\*exp(-b\*y), y>=0, c параметром b=0.1 и соответственно mэксп=sэксп= 1/b=10.

Значения случайной величины Y с экспоненциальным законом распределения с параметром «b» можно получить по значениям случайной величины t, равномерно распределенной в интервале [0,1], по формуле [1]: Y = -ln(t) / b

В) релеевским законом распределения

W(y) = (y/c^2)\*exp(-y^2/(2\*c^2)), y>=0, c параметром c=8.0 и соответственно mрел = c\*sqrt(□/2), sрел= c\*sqrt(2-□/2).

Значения случайной величины Y с релеевским законом распределения с параметром «с» можно получить по значениям случайной величины t, равномерно распределенной в интервале [0,1], по формуле [1]: Y = с \* sqrt(-2\*ln(t)).

1. Каждый из 3-х массивов {Xi} интервалов времени между соседними ошибками упорядочить по возрастанию.
2. Для каждого из 3-х массивов {Xi} оценить значение первоначального числа ошибок в программе B. При этом для каждого закона использовать 100%, 80% и 60% входных данных (то есть в массивах

{Хi} использовать n = 30, 24 и 18 элементов).

*Примечание*: для каждого значения n следует генерировать и сортировать новые массивы.

1. Если B>n, оценить значения средних времен Xj , j=n+1,n+2…, n+k до обнаружения k<= 5 следующих ошибок и общее время на выполнение тестирования.
2. Результаты вычислений представить в виде двух таблиц, одна из которых содержит оценки первоначального числа ошибок, а другая – оценки полных времен проведения тестирования - для разных законов распределения времен между отказами и разного числа используемых данных.

Сравнить и объяснить результаты, полученные для различных законов распределения времени между соседними отказами и различного числа используемых для анализа данных.

## Ход работы

1. *РАВНОМЕРНОЕ РАСПРЕДЕЛЕНИЕ*

**100% входных данных:**

Был сгенерирован массив из 30-ти элементов, равномерно распределённых на интервале [0, 20].

Генерация происходила с помощью функции np.random.uniform(0, 20, 30). Массив был упорядочен по возрастанию. Результаты представлены в таблице 1.

Таблица 1 – Равномерное распределение, n = 30 (100%).

|  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **i** | **1** | **2** | **3** | **4** | **5** | **6** | **7** | **8** | **9** | **10** |
| **Xi** | 0.264 | 0.499 | 2.328 | 2.607 | 4.842 | 5.998 | 6.205 | 7.115 | 8.044 | 8.584 |
| **i** | **11** | **12** | **13** | **14** | **15** | **16** | **17** | **18** | **19** | **20** |
| **Xi** | 8.931 | 9.949 | 11.186 | 11.777 | 12.авг | 12.361 | 12.664 | 12.785 | 12.793 | 14.113 |
| **i** | **21** | **22** | **23** | **24** | **25** | **26** | **27** | **28** | **29** | **30** |
| **Xi** | 14.521 | 14.608 | 14.808 | 15.338 | 15.566 | 15.875 | 16.905 | 17.028 | 19.674 | 19.799 |

Условие сходимости: 𝐴 > (𝑛+ 1)/2



𝐴 = 19,59 > 15.5

Были вычислены значения



Результаты представлены в таблице 2.

Таблица 2 – Расчёт значений функций для равномерного распределения (100%).

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 𝑚 | 31 | 32 | 33 | 34 | 35 | 36 | 37 | 38 |
| 𝑓𝑛(𝑚) | 3.99499 | 3.02725 | 2.5585 | 2.25546 | 2.03488 | 1.86345 | 1.72456 | 1.60873 |
| 𝑔(𝑚, 𝐴) | 2.62914 | 2.41729 | 2.23704 | 2.0818 | 1.94671 | 1.82809 | 1.72309 | 1.6295 |
| |𝑓𝑛(𝑚) − 𝑔(𝑚, 𝐴)| | 1.36585 | 0.609953 | 0.321456 | 0.173661 | 0.0881624 | 0.0353594 | 0.00146921 | 0.0207685 |

Минимум разности достигается при 𝑚 = 37.

Первоначальное количество ошибок 𝐵 = 𝑚 – 1 = 36.



K = 0.0052

Было рассчитано среднее время обнаружения следующих ошибок.



Результат представлен в таблице 3.

Таблица 3 – Расчет времени обнаружения следующих ошибок для равномерного распределения (100%).

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| i | 31 | 32 | 33 | 34 | 35 | 36 |
| Xi | 31.8466 | 38.2159 | 47.7698 | 63.6931 | 95.5397 | 191.079 |

Время до полного завершения тестирования 468.14

Полное время: 797

**80% входных данных:**

Был сгенерирован массив из 24-х элементов, равномерно распределённых на интервале [0, 20]. Генерация происходила с помощью функции np.random.uniform(0, 20, 24). Массив был упорядочен по возрастанию. Результаты представлены в таблице 4.

Таблица 4 – Равномерное распределение, n = 24 (80%).

|  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **i** | **1** | **2** | **3** | **4** | **5** | **6** | **7** | **8** | **9** | **10** |
| **X\_i** | 0.11 | 0.975 | 1.146 | 1.247 | 2.494 | 3.607 | 3.681 | 3.718 | 4.887 | 4.979 |
| **i** | **11** | **12** | **13** | **14** | **15** | **16** | **17** | **18** | **19** | **20** |
| **X\_i** | 6.283 | 6.849 | 7.292 | 8.524 | 8.527 | 8.538 | 8.947 | 13.354 | 13.703 | 14.041 |
| **i** | **21** | **22** | **23** | **24** |  |  |  |  |  |  |
| **X\_i** | 15.221 | 15.611 | 17.303 | 17.421 |  |  |  |  |  |  |

Условие сходимости: 𝐴 > (𝑛+ 1)/2



𝐴 = 17,17 > 12.5

Были вычислены значения



Результаты представлены в таблице 5.

Таблица 5 – Расчёт значений функций для равномерного распределения (80%).

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 𝑚 | 25 | 26 | 27 | 28 |
| 𝑓𝑛(𝑚) | 3.77596 | 2.81596 | 2.35442 | 2.05812 |
| 𝑔(𝑚, 𝐴) | 3.06681 | 2.71933 | 2.44257 | 2.21694 |
| |𝑓𝑛(𝑚) − 𝑔(𝑚, 𝐴)| | 0.709145 | 0.0966308 | 0.0881514 | 0.158821 |

Минимум разности достигается при 𝑚 = 27.

Первоначальное количество ошибок 𝐵 = 𝑚 – 1 = 26.



K = 0.01296

Было рассчитано среднее время обнаружения следующих ошибок.



Результат представлен в таблице 6.

Таблица 6 – Расчет времени обнаружения следующих ошибок для равномерного распределения (80%).

|  |  |  |
| --- | --- | --- |
| i | 25 | 26 |
| Xi | 38.5778 | 77.1556 |

Время до полного завершения тестирования 115.73

Полное время: 304

**60% входных данных:**

Был сгенерирован массив из 18-ти элементов, равномерно распределённых на интервале [0, 20]. Генерация происходила с помощью функции np.random.uniform(0, 20, 18). Массив был упорядочен по возрастанию. Результаты представлены в таблице 1.

Таблица 7 – Равномерное распределение, n = 18 (60%).

|  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **i** | **1** | **2** | **3** | **4** | **5** | **6** | **7** | **8** | **9** | **10** |
| **X\_i** | 0.28 | 2.459 | 3.266 | 3.571 | 4.009 | 5.174 | 6.918 | 8.569 | 9.098 | 10.297 |
| **i** | **11** | **12** | **13** | **14** | **15** | **16** | **17** | **18** |  |  |
| **X\_i** | 10.842 | 12.389 | 12.876 | 13.261 | 14.322 | 15.571 | 16.161 | 19.901 |  |  |

Условие сходимости: 𝐴 > (𝑛+ 1)/2



𝐴 = 12,43 > 9.5

Были вычислены значения



Результаты представлены в таблице 8.

Таблица 8 – Расчёт значений функций для равномерного распределения (60%).

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 𝑚 | 19 | 20 | 21 | 22 |
| 𝑓𝑛(𝑚) | 3.49511 | 2.54774 | 2.09774 | 1.81203 |
| 𝑔(𝑚, 𝐴) | 2.7394 | 2.37756 | 2.10016 | 1.88073 |
| |𝑓𝑛(𝑚) − 𝑔(𝑚, 𝐴)| | 0.755704 | 0.170175 | 0.0024211 | 0.0687006 |

Минимум разности достигается при 𝑚 = 21.

Первоначальное количество ошибок 𝐵 = 𝑚 – 1 = 20.



K = 0.0124

Было рассчитано среднее время обнаружения следующих ошибок.



Результат представлен в таблице 9.

Таблица 9 – Расчет времени обнаружения следующих ошибок для равномерного распределения (60%).

|  |  |  |
| --- | --- | --- |
| i | 19 | 20 |
| Xi | 40.2264 | 80.4529 |

Время до полного завершения тестирования 120.68

Полное время: 290

1. *ЭКСПОНЕНЦИАЛЬНЫЙ ЗАКОН РАСПРЕДЕЛЕНИЯ*

**100% входных данных:**

Был сгенерирован массив из 30-ти элементов, распределённых по закону W(y) = b\*exp(-b\*y),y>=0, c параметром b=0.1.

Генерация происходила с помощью функции np.random.exponential(10, 30).

Массив был упорядочен по возрастанию.

Результаты представлены в таблице 10.

Таблица 10 – Экспоненциальное распределение, n = 30 (100%).

|  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **i** | **1** | **2** | **3** | **4** | **5** | **6** | **7** | **8** | **9** | **10** |
| **X\_i** | 0.237 | 0.259 | 0.429 | 1.029 | 1.063 | 1.457 | 1.494 | 2.187 | 4.362 | 4.38 |
| **i** | **11** | **12** | **13** | **14** | **15** | **16** | **17** | **18** | **19** | **20** |
| **X\_i** | 4.551 | 4.779 | 6.242 | 6.315 | 6.562 | 7.278 | 7.94 | 8.908 | 9.35 | 10.359 |
| **i** | **21** | **22** | **23** | **24** | **25** | **26** | **27** | **28** | **29** | **30** |
| **X\_i** | 10.838 | 15.106 | 16.648 | 16.817 | 22.294 | 23.658 | 24.081 | 28.527 | 32.21 | 41.616 |

Условие сходимости: 𝐴 > (𝑛+ 1)/2



𝐴 = 23,22 > 15.5

Были вычислены значения



Результаты представлены в таблице 11.

Таблица 11 – Расчёт значений функций для экспоненциального распределения (100%).

|  |  |  |
| --- | --- | --- |
| 𝑚 | 31 | 32 |
| 𝑓𝑛(𝑚) | 3.99499 | 3.02725 |
| 𝑔(𝑚, 𝐴) | 3.85693 | 3.41755 |
| |𝑓𝑛(𝑚) − 𝑔(𝑚, 𝐴)| | 0.13806 | 0.39031 |

Минимум разности достигается при 𝑚 = 31.

Первоначальное количество ошибок 𝐵 = 𝑚 – 1 = 30.



K = 0.01202

Условие 𝐵 > *n* не выполняется.

Полное время: 321 дней

**80% входных данных:**

Был сгенерирован массив из 30-ти элементов, распределённых по закону W(y) = b\*exp(-b\*y),y>=0, c параметром b=0.1.

Генерация происходила с помощью функции np.random.exponential(10, 24).

Массив был упорядочен по возрастанию.

Результаты представлены в таблице 12.

Таблица 12 – Экспоненциальное распределение, n = 24 (80%).

|  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **i** | **1** | **2** | **3** | **4** | **5** | **6** | **7** | **8** | **9** | **10** |
| **X\_i** | 0.368 | 0.705 | 1.556 | 2.051 | 2.52 | 2.785 | 3.272 | 3.554 | 3.824 | 4.403 |
| **i** | **11** | **12** | **13** | **14** | **15** | **16** | **17** | **18** | **19** | **20** |
| **X\_i** | 6.183 | 8.461 | 8.581 | 9.335 | 10.911 | 11.7 | 12.4 | 13.063 | 14.008 | 15.714 |
| **i** | **21** | **22** | **23** | **24** |  |  |  |  |  |  |
| **X\_i** | 24.596 | 24.624 | 29.722 | 59.825 |  |  |  |  |  |  |

Условие сходимости: 𝐴 > (𝑛+ 1)/2



𝐴 = 18, 75 > 12.5

Были вычислены значения



Результаты представлены в таблице 13.

Таблица 13 – Расчёт значений функций для экспоненциального распределения (80%).

|  |  |  |
| --- | --- | --- |
| 𝑚 | 25 | 26 |
| 𝑓𝑛(𝑚) | 3.77596 | 2.81596 |
| 𝑔(𝑚, 𝐴) | 3.83976 | 3.31017 |
| |𝑓𝑛(𝑚) − 𝑔(𝑚, 𝐴)| | 0.0638004 | 0.494207 |

Минимум разности достигается при 𝑚 = 25.

Первоначальное количество ошибок 𝐵 = 𝑚 – 1 = 24.



K = 0.014005

Условие 𝐵 > *n* не выполняется.

Полное время: 274

**60% входных данных:**

Был сгенерирован массив из 30-ти элементов, распределённых по закону W(y) = b\*exp(-b\*y),y>=0, c параметром b=0.1.

Генерация происходила с помощью функции np.random.exponential(10, 18).

Массив был упорядочен по возрастанию.

Результаты представлены в таблице 14.

Таблица 14 – Экспоненциальное распределение, n = 18 (60%).

|  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **i** | **1** | **2** | **3** | **4** | **5** | **6** | **7** | **8** | **9** | **10** |
| **X\_i** | 0.847 | 0.848 | 0.899 | 1.76 | 2.456 | 4.568 | 4.822 | 6.748 | 7.109 | 9.407 |
| **i** | **11** | **12** | **13** | **14** | **15** | **16** | **17** | **18** |  |  |
| **X\_i** | 12.647 | 13.785 | 14.311 | 14.444 | 17.311 | 18.101 | 18.402 | 50.914 |  |  |

Условие сходимости: 𝐴 > (𝑛+ 1)/2



𝐴 = 13,89 > 9.5

Были вычислены значения



Результаты представлены в таблице 15.

Таблица 15 – Расчёт значений функций для экспоненциального распределения (60%).

|  |  |  |
| --- | --- | --- |
| 𝑚 | 19 | 20 |
| 𝑓𝑛(𝑚) | 3.49511 | 2.54774 |
| 𝑔(𝑚, 𝐴) | 3.52163 | 2.94538 |
| |𝑓𝑛(𝑚) − 𝑔(𝑚, 𝐴)| | 0.0265178 | 0.397636 |

Минимум разности достигается при 𝑚 = 19.

Первоначальное количество ошибок 𝐵 = 𝑚 – 1 = 18.



K = 0.0124

Условие 𝐵 > *n* не выполняется.

Полное время: 199

1. *РЕЛЕЕВСКИЙ ЗАКОН РАСПРЕДЕЛЕНИЯ*

**100% входных данных:**

Был сгенерирован массив из 30-ти элементов, распределённых по закону W(y) = (y/c^2)\*exp(-y^2/(2\*c^2)), y>=0, c параметром c=8.0.

Генерация происходила с помощью функции np.random.rayleigh(8, 30).

Массив был упорядочен по возрастанию.

Результаты представлены в таблице 16.

Таблица 16 – Релеевское распределение, n = 30 (100%).

|  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **i** | **1** | **2** | **3** | **4** | **5** | **6** | **7** | **8** | **9** | **10** |
| **X\_i** | 2.923 | 3.493 | 4.785 | 5.072 | 5.644 | 7.214 | 7.708 | 7.795 | 8.005 | 8.107 |
| **i** | **11** | **12** | **13** | **14** | **15** | **16** | **17** | **18** | **19** | **20** |
| **X\_i** | 8.253 | 8.892 | 9.327 | 9.426 | 9.5 | 9.544 | 9.56 | 9.95 | 10.605 | 10.945 |
| **i** | **21** | **22** | **23** | **24** | **25** | **26** | **27** | **28** | **29** | **30** |
| **X\_i** | 11.33 | 12.129 | 12.199 | 12.435 | 13.501 | 13.688 | 14.157 | 15.478 | 15.629 | 17.669 |

Условие сходимости: 𝐴 > (𝑛+ 1)/2



𝐴 = 18,54 > 15.5

Были вычислены значения



Результаты представлены в таблице 17.

Таблица 17 – Расчёт значений функций для релеевского распределения (100%).

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| 𝑚 | 31 | 32 | 33 | 34 | 35 | 36 | 37 |
| 𝑓𝑛(𝑚) | 3.99499 | 3.02725 | 2.5585 | 2.25546 | 2.03488 | 1.86345 | 1.72456 |
| 𝑔(𝑚, 𝐴) | 2.40831 | 2.22935 | 2.07514 | 1.94089 | 1.82295 | 1.71852 | 1.62541 |
| |𝑓𝑛(𝑚) − 𝑔(𝑚, 𝐴)| | 1.58668 | 0.797899 | 0.483356 | 0.314579 | 0.211929 | 0.144926 | 0.0991474 |
| 𝑚 | 38 | 39 | 40 | 41 | 42 | 43 | 44 |
| 𝑓𝑛(𝑚) | 1.60873 | 1.51004 | 1.42457 | 1.34957 | 1.28306 | 1.22353 | 1.16986 |
| 𝑔(𝑚, 𝐴) | 1.54187 | 1.4665 | 1.39815 | 1.33589 | 1.27894 | 1.22665 | 1.17846 |
| |𝑓𝑛(𝑚) − 𝑔(𝑚, 𝐴)| | 0.0668565 | 0.0435442 | 0.0264206 | 0.0136801 | 0.00411242 | 0.00311754 | 0.00859938 |

Минимум разности достигается при 𝑚 = 43.

Первоначальное количество ошибок 𝐵 = 𝑚 – 1 = 42.



K = 0.004

Было рассчитано среднее время обнаружения следующих ошибок.



Результат представлен в таблице 18.

Таблица 18 – Расчет времени обнаружения следующих ошибок для релеевского распределения (100%).

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| i | 31 | 32 | 33 | 34 | 35 | 36 |
| Xi | 20.0385 | 21.8602 | 24.0462 | 26.718 | 30.0578 | 34.3518 |
| i | 37 | 38 | 39 | 40 | 41 | 42 |
| Xi | 40.077 | 48.0925 | 60.1156 | 80.1541 | 120.231 | 240.462 |

Время до полного завершения тестирования 746.2

Полное время: 1041.2

**80% входных данных:**

Был сгенерирован массив из 24-х элементов, распределённых по закону W(y) = (y/c^2)\*exp(-y^2/(2\*c^2)), y>=0, c параметром c=8.0.

Генерация происходила с помощью функции np.random.rayleigh(8, 24).

Массив был упорядочен по возрастанию. Результаты представлены в таблице 19.

Таблица 19 – Релеевское распределение, n = 24 (80%).

|  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **i** | **1** | **2** | **3** | **4** | **5** | **6** | **7** | **8** | **9** | **10** |
| **X\_i** | 1.817 | 2.739 | 2.939 | 3.509 | 3.774 | 5.193 | 5.41 | 5.502 | 6.179 | 8.191 |
| **i** | **11** | **12** | **13** | **14** | **15** | **16** | **17** | **18** | **19** | **20** |
| **X\_i** | 8.211 | 8.468 | 8.682 | 9.887 | 10.563 | 10.845 | 11.313 | 11.667 | 12.659 | 14.853 |
| **i** | **21** | **22** | **23** | **24** |  |  |  |  |  |  |
| **X\_i** | 15.378 | 15.51 | 17.061 | 22.242 |  |  |  |  |  |  |

Условие сходимости: 𝐴 > (𝑛+ 1)/2



𝐴 = 16,196 > 12.5

Были вычислены значения



Результаты представлены в таблице 20.

Таблица 20 – Расчёт значений функций для релеевского распределения (80%).

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 𝑚 | 25 | 26 | 27 | 28 | 29 |
| 𝑓𝑛(𝑚) | 3.77596 | 2.81596 | 2.35442 | 2.05812 | 1.84384 |
| 𝑔(𝑚, 𝐴) | 2.72596 | 2.44792 | 2.22135 | 2.03317 | 1.87438 |
| |𝑓𝑛(𝑚) − 𝑔(𝑚, 𝐴)| | 1.05 | 0.368038 | 0.13307 | 0.0249561 | 0.030541 |

Минимум разности достигается при 𝑚 = 28.

Первоначальное количество ошибок 𝐵 = 𝑚 – 1 = 27.



K = 0.009

Было рассчитано среднее время обнаружения следующих ошибок.



Результат представлен в таблице 21.

Таблица 21 – Расчет времени обнаружения следующих ошибок для релеевского распределения (80%).

|  |  |  |  |
| --- | --- | --- | --- |
| i | 25 | 26 | 27 |
| Xi | 36.4935 | 54.7402 | 109.48 |

Время до полного завершения тестирования 200,7

Полное время: 423,3

**60% входных данных:**

Был сгенерирован массив из 18-ти элементов, распределённых по закону W(y) = (y/c^2)\*exp(-y^2/(2\*c^2)), y>=0, c параметром c=8.0.

Генерация происходила с помощью функции np.random.rayleigh(8, 18).

Массив был упорядочен по возрастанию. Результаты представлены в таблице 22.

Таблица 22 – Релеевское распределение, n = 18 (60%).

|  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **i** | **1** | **2** | **3** | **4** | **5** | **6** | **7** | **8** | **9** | **10** |
| **X\_i** | 0.686 | 3.234 | 4.062 | 4.286 | 4.679 | 8.321 | 8.573 | 9.609 | 9.728 | 9.822 |
| **i** | **11** | **12** | **13** | **14** | **15** | **16** | **17** | **18** |  |  |
| **X\_i** | 9.975 | 10.29 | 10.477 | 12.35 | 14.882 | 15.394 | 18.022 | 19.017 |  |  |

Условие сходимости: 𝐴 > (𝑛+ 1)/2



𝐴 = 12,07 > 9.5

Были вычислены значения



Результаты представлены в таблице 23.

Таблица 23 – Расчёт значений функций для релеевского распределения (60%).

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 𝑚 | 19 | 20 | 21 | 22 |
| 𝑓𝑛(𝑚) | 3.49511 | 2.54774 | 2.09774 | 1.81203 |
| 𝑔(𝑚, 𝐴) | 2.59731 | 2.26979 | 2.01562 | 1.81264 |
| |𝑓𝑛(𝑚) − 𝑔(𝑚, 𝐴)| | 0.897797 | 0.277949 | 0.0821175 | 0.000618722 |

Минимум разности достигается при 𝑚 = 22.

Первоначальное количество ошибок 𝐵 = 𝑚 – 1 = 21.



K = 0.0105

Было рассчитано среднее время обнаружения следующих ошибок.



Результат представлен в таблице 24.

Таблица 24 – Расчет времени обнаружения следующих ошибок для релеевского распределения (60%).

|  |  |  |  |
| --- | --- | --- | --- |
| i | 19 | 20 | 21 |
| Xi | 31.8884 | 47.8326 | 95.6652 |

Время до полного завершения тестирования 175,39

Полное время: 348.79

1. *РЕЗУЛЬТАТЫ РАСЧЕТОВ*

В таблицах 25 и 26 представлены сводные результаты оценки первоначального числа ошибок и полного времени выполнения тестирования соответственно.

Таблица 25 – Оценка первоначального числа ошибок.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| n | Входные данные, % | Распределение | | |
| Равномерное | Экспоненциальное | Релеевское |
| 30 | 100 | 36 | 30 | 42 |
| 24 | 80 | 26 | 24 | 27 |
| 18 | 60 | 20 | 18 | 21 |

Таблица 26 – Оценка полного времени проведения тестирования

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| n | Входные данные, % | Распределение | | |
| Равномерное | Экспоненциальное | Релеевское |
| 30 | 100 | 797 | 320 | 1041 |
| 24 | 80 | 304 | 274 | 423 |
| 18 | 60 | 289 | 199 | 348 |

Результаты при экспоненциальном распределении оказались ниже, других, что связано с тем, что модель Джелинского-Моранды основана на том, что время до следующего отказа программы распределено экспоненциально. По сравнению с равномерным распределением, релеевское оказывается хуже.

**Выводы.**

В результате выполнения данной лабораторной работы было выполнено исследование показателей надежности программ, характеризуемых моделью обнаружения ошибок Джелински-Морданы, для различных законов распределения времен обнаружения отказов и различного числа используемых для анализа данных.