## 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 Сершеев А.Д.

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(π/2), sрел= c\*sqrt(2-π/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** | 1.851 | 2.745 | 2.922 | 3.94 | 3.972 | 4.619 | 4.743 | 5.201 | 7.14 | 7.365 |
| **i** | **11** | **12** | **13** | **14** | **15** | **16** | **17** | **18** | **19** | **20** |
| **Xi** | 7.454 | 7.513 | 7.638 | 9.362 | 9.781 | 11.084 | 11.151 | 11.938 | 12.739 | 12.821 |
| **i** | **21** | **22** | **23** | **24** | **25** | **26** | **27** | **28** | **29** | **30** |
| **Xi** | 13.293 | 13.457 | 13.527 | 13.684 | 13.724 | 13.831 | 13.93 | 14.42 | 14.941 | 18.754 |

Условие сходимости: 𝐴 > (𝑛+ 1)/2



𝐴 = 19.392 > 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 | 39 |
| 𝑓𝑛(𝑚) | 3.995 | 3.027 | 2.559 | 2.255 | 2.035 | 1.863 | 1.725 | 1.609 | 1.51 |
| 𝑔(𝑚, 𝐴) | 2.585 | 2.38 | 2.205 | 2.054 | 1.922 | 1.806 | 1.704 | 1.612 | 1.53 |
| |𝑓𝑛(𝑚) − 𝑔(𝑚, 𝐴)| | 1.41 | 0.648 | 0.354 | 0.202 | 0.113 | 0.057 | 0.021 | 0.004 | 0.02 |

Минимум разности достигается при 𝑚 = 38.

Первоначальное количество ошибок 𝐵 = 𝑚 – 1 = 37.



K = 0.006

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 | 25.656 | 29.931 | 35.918 | 44.897 | 59.863 | 89.794 | 179.588 |

Время до полного завершения тестирования: 465.647

Полное время: 755

**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** | 0.386 | 2.079 | 2.508 | 3.279 | 4.687 | 5.7 | 7.632 | 8.677 | 9.121 | 10.198 |
| **i** | **11** | **12** | **13** | **14** | **15** | **16** | **17** | **18** | **19** | **20** |
| **X\_i** | 10.602 | 11.637 | 11.64 | 11.859 | 12.736 | 15.559 | 16.973 | 17.835 | 18.349 | 19.024 |
| **i** | **21** | **22** | **23** | **24** |  |  |  |  |  |  |
| **X\_i** | 19.251 | 19.525 | 19.656 | 19.782 |  |  |  |  |  |  |

Условие сходимости: 𝐴 > (𝑛+ 1)/2



𝐴 = 16.156 > 12.5

Были вычислены значения



Результаты представлены в таблице 5.

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

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| 𝑚 | 25 | 26 | 27 | 28 | 29 | 30 |
| 𝑓𝑛(𝑚) | 3.776 | 2.816 | 2.354 | 2.058 | 1.844 | 1.678 |
| 𝑔(𝑚, 𝐴) | 2.714 | 2.438 | 2.213 | 2.026 | 1.869 | 1.734 |
| |𝑓𝑛(𝑚) − 𝑔(𝑚, 𝐴)| | 1.062 | 0.378 | 0.141 | 0.032 | 0.025 | 0.055 |

Минимум разности достигается при 𝑚 = 29.

Первоначальное количество ошибок 𝐵 = 𝑚 – 1 = 28.



K = 0.007

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 | 27 | 28 |
| Xi | 37.286 | 49.715 | 74.572 | 149.145 |

Время до полного завершения тестирования: 310.718

Полное время: 589

**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** | 0.631 | 0.668 | 1.135 | 5.31 | 6.546 | 6.702 | 9.52 | 10.011 | 10.407 | 12.273 |
| **i** | **11** | **12** | **13** | **14** | **15** | **16** | **17** | **18** |  |  |
| **X\_i** | 13.612 | 14.476 | 14.742 | 15.413 | 15.809 | 18.563 | 18.826 | 19.467 |  |  |

Условие сходимости: 𝐴 > (𝑛+ 1)/2



𝐴 = 12,43 > 12.354

Были вычислены значения



Результаты представлены в таблице 8.

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

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 𝑚 | 19 | 20 | 21 | 22 |
| 𝑓𝑛(𝑚) | 3.495 | 2.548 | 2.098 | 1.812 |
| 𝑔(𝑚, 𝐴) | 2.709 | 2.354 | 2.082 | 1.866 |
| |𝑓𝑛(𝑚) − 𝑔(𝑚, 𝐴)| | 0.787 | 0.193 | 0.016 | 0.054 |

Минимум разности достигается при 𝑚 = 21.

Первоначальное количество ошибок 𝐵 = 𝑚 – 1 = 20.



K = 0.011

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 | 46.617 | 93.233 |

Время до полного завершения тестирования: 139.849

Полное время: 333

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** | 0.505 | 1.288 | 1.289 | 2.234 | 2.627 | 2.972 | 3.273 | 3.608 | 3.633 | 3.66 |
| **i** | **11** | **12** | **13** | **14** | **15** | **16** | **17** | **18** | **19** | **20** |
| **X\_i** | 3.674 | 3.762 | 4.205 | 4.539 | 4.868 | 4.985 | 5.564 | 5.706 | 5.776 | 6.638 |
| **i** | **21** | **22** | **23** | **24** | **25** | **26** | **27** | **28** | **29** | **30** |
| **X\_i** | 6.679 | 10.114 | 10.119 | 10.257 | 10.374 | 27.566 | 31.381 | 40.123 | 43.124 | 51.422 |

Условие сходимости: 𝐴 > (𝑛+ 1)/2



𝐴 = 23.882 > 15.5

Были вычислены значения



Результаты представлены в таблице 11.

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

|  |  |  |
| --- | --- | --- |
| 𝑚 | 31 | 32 |
| 𝑓𝑛(𝑚) | 3.995 | 3.027 |
| 𝑔(𝑚, 𝐴) | 4.215 | 3.696 |
| |𝑓𝑛(𝑚) − 𝑔(𝑚, 𝐴)| | 0.22 | 0.668 |

Минимум разности достигается при 𝑚 = 31.

Первоначальное количество ошибок 𝐵 = 𝑚 – 1 = 30.



K = 0.013

Условие 𝐵 > *n* не выполняется.

Полное время: 315

**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** | 0.095 | 0.497 | 0.787 | 0.818 | 1.439 | 1.556 | 1.671 | 1.926 | 2.259 | 2.339 |
| **i** | **11** | **12** | **13** | **14** | **15** | **16** | **17** | **18** | **19** | **20** |
| **X\_i** | 2.637 | 2.887 | 4.525 | 4.866 | 5.495 | 6.574 | 6.657 | 12.669 | 12.731 | 13.017 |
| **i** | **21** | **22** | **23** | **24** |  |  |  |  |  |  |
| **X\_i** | 18.185 | 19.767 | 24.332 | 36.021 |  |  |  |  |  |  |

Условие сходимости: 𝐴 > (𝑛+ 1)/2



𝐴 = 19.356 > 12.5

Были вычислены значения



Результаты представлены в таблице 13.

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

|  |  |  |
| --- | --- | --- |
| 𝑚 | 25 | 26 |
| 𝑓𝑛(𝑚) | 3.776 | 2.816 |
| 𝑔(𝑚, 𝐴) | 4.252 | 3.612 |
| |𝑓𝑛(𝑚) − 𝑔(𝑚, 𝐴)| | 0.476 | 0.796 |

Минимум разности достигается при 𝑚 = 25.

Первоначальное количество ошибок 𝐵 = 𝑚 – 1 = 24.



K = 0.0231

Условие 𝐵 > *n* не выполняется.

Полное время: 183

**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** | 1.925 | 1.943 | 2.5 | 3.541 | 3.949 | 3.991 | 4.701 | 5.502 | 7.421 | 10.765 |
| **i** | **11** | **12** | **13** | **14** | **15** | **16** | **17** | **18** |  |  |
| **X\_i** | 12.331 | 13.074 | 15.43 | 16.819 | 17.065 | 17.084 | 24.498 | 26.826 |  |  |

Условие сходимости: 𝐴 > (𝑛+ 1)/2



𝐴 = 13.095 > 9.5

Были вычислены значения



Результаты представлены в таблице 15.

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

|  |  |  |  |
| --- | --- | --- | --- |
| 𝑚 | 19 | 20 | 21 |
| 𝑓𝑛(𝑚) | 3.495 | 2.548 | 2.098 |
| 𝑔(𝑚, 𝐴) | 3.048 | 2.607 | 2.277 |
| |𝑓𝑛(𝑚) − 𝑔(𝑚, 𝐴)| | 0.447 | 0.059 | 0.179 |

Минимум разности достигается при 𝑚 = 20.

Первоначальное количество ошибок 𝐵 = 𝑚 – 1 = 19.



K = 0.0124

Было рассчитано среднее время обнаружения следующих ошибок.



Результат представлен в таблице 16.

Таблица 16 – Расчет времени обнаружения следующих ошибок для экспоненциального распределения (60%).

|  |  |
| --- | --- |
| i | 19 |
| Xi | 72.641 |

Время до полного завершения тестирования: 72.641

Полное время: 262

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 17.

Таблица 17 – Релеевское распределение, n = 30 (100%).

|  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **i** | **1** | **2** | **3** | **4** | **5** | **6** | **7** | **8** | **9** | **10** |
| **X\_i** | 3.736 | 3.771 | 4.385 | 4.816 | 5.083 | 6.065 | 6.426 | 7.364 | 7.799 | 8.939 |
| **i** | **11** | **12** | **13** | **14** | **15** | **16** | **17** | **18** | **19** | **20** |
| **X\_i** | 9.246 | 9.681 | 10.071 | 10.391 | 11.184 | 11.264 | 12.044 | 12.898 | 13.529 | 14.003 |
| **i** | **21** | **22** | **23** | **24** | **25** | **26** | **27** | **28** | **29** | **30** |
| **X\_i** | 14.192 | 14.661 | 15.647 | 16.116 | 16.359 | 16.6 | 17.59 | 19.549 | 20.085 | 20.43 |

Условие сходимости: 𝐴 > (𝑛+ 1)/2



𝐴 = 19.222 > 15.5

Были вычислены значения



Результаты представлены в таблице 18.

Таблица 18 – Расчёт значений функций для релеевского распределения (100%).

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| 𝑚 | 31 | 32 | 33 | 34 | 35 | 36 | 37 |
| 𝑓𝑛(𝑚) | 3.995 | 3.027 | 2.559 | 2.255 | 2.035 | 1.863 | 1.725 |
| 𝑔(𝑚, 𝐴) | 2.547 | 2.348 | 2.177 | 2.03 | 1.901 | 1.788 | 1.687 |
| |𝑓𝑛(𝑚) − 𝑔(𝑚, 𝐴)| | 1.448 | 0.68 | 0.381 | 0.225 | 0.134 | 0.075 | 0.037 |
| 𝑚 | 38 | 39 | 40 |  |  |  |  |
| 𝑓𝑛(𝑚) | 1.609 | 1.510 | 1.425 |  |  |  |  |
| 𝑔(𝑚, 𝐴) | 1.598 | 1.517 | 1.444 |  |  |  |  |
| |𝑓𝑛(𝑚) − 𝑔(𝑚, 𝐴)| | 0.011 | 0.007 | 0.019 |  |  |  |  |

Минимум разности достигается при 𝑚 = 39.

Первоначальное количество ошибок 𝐵 = 𝑚 – 1 = 38.



K = 0.004

Было рассчитано среднее время обнаружения следующих ошибок.



Результат представлен в таблице 19.

Таблица 19 – Расчет времени обнаружения следующих ошибок для релеевского распределения (100%).

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| i | 31 | 32 | 33 | 34 | 35 | 36 | 37 | 38 |
| Xi | 28.343 | 32.392 | 37.791 | 45.349 | 56.686 | 75.581 | 113.372 | 226.743 |

Время до полного завершения тестирования: 616.256

Полное время: 960

**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 20.

Таблица 20 – Релеевское распределение, n = 24 (80%).

|  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **i** | **1** | **2** | **3** | **4** | **5** | **6** | **7** | **8** | **9** | **10** |
| **X\_i** | 2.131 | 2.403 | 2.995 | 3.169 | 3.362 | 3.464 | 3.542 | 3.65 | 4.379 | 4.596 |
| **i** | **11** | **12** | **13** | **14** | **15** | **16** | **17** | **18** | **19** | **20** |
| **X\_i** | 4.699 | 7.573 | 8.123 | 9.216 | 10.891 | 10.991 | 12.106 | 12.312 | 12.694 | 12.711 |
| **i** | **21** | **22** | **23** | **24** |  |  |  |  |  |  |
| **X\_i** | 14.115 | 15.217 | 15.741 | 15.886 |  |  |  |  |  |  |

Условие сходимости: 𝐴 > (𝑛+ 1)/2



𝐴 = 16.428 > 12.5

Были вычислены значения



Результаты представлены в таблице 21.

Таблица 21 – Расчёт значений функций для релеевского распределения (80%).

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 𝑚 | 25 | 26 | 27 | 28 | 29 |
| 𝑓𝑛(𝑚) | 3.776 | 2.816 | 2.354 | 2.058 | 1.844 |
| 𝑔(𝑚, 𝐴) | 2.8 | 2.507 | 2.27 | 2.074 | 1.909 |
| |𝑓𝑛(𝑚) − 𝑔(𝑚, 𝐴)| | 0.976 | 0.309 | 0.084 | 0.016 | 0.065 |

Минимум разности достигается при 𝑚 = 28.

Первоначальное количество ошибок 𝐵 = 𝑚 – 1 = 27.



K = 0.011

Было рассчитано среднее время обнаружения следующих ошибок.



Результат представлен в таблице 22.

Таблица 22 – Расчет времени обнаружения следующих ошибок для релеевского распределения (80%).

|  |  |  |  |
| --- | --- | --- | --- |
| i | 25 | 26 | 27 |
| Xi | 31.497 | 47.245 | 94.49 |

Время до полного завершения тестирования: 173.232

Полное время: 369

**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 23.

Таблица 23 – Релеевское распределение, n = 18 (60%).

|  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **i** | **1** | **2** | **3** | **4** | **5** | **6** | **7** | **8** | **9** | **10** |
| **X\_i** | 1.58 | 3.393 | 4.18 | 4.729 | 6.07 | 8.553 | 9.583 | 9.827 | 10.526 | 10.647 |
| **i** | **11** | **12** | **13** | **14** | **15** | **16** | **17** | **18** |  |  |
| **X\_i** | 10.744 | 11.043 | 11.633 | 13.821 | 14.042 | 15.286 | 16.257 | 27.132 |  |  |

Условие сходимости: 𝐴 > (𝑛+ 1)/2



𝐴 = 12.08 > 9.5

Были вычислены значения



Результаты представлены в таблице 24.

Таблица 24 – Расчёт значений функций для релеевского распределения (60%).

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 𝑚 | 19 | 20 | 21 | 22 | 23 |
| 𝑓𝑛(𝑚) | 3.495 | 2.548 | 2.098 | 1.812 | 1.607 |
| 𝑔(𝑚, 𝐴) | 2.601 | 2.273 | 2.018 | 1.815 | 1.648 |
| |𝑓𝑛(𝑚) − 𝑔(𝑚, 𝐴)| | 0.894 | 0.275 | 0.08 | 0.003 | 0.04 |

Минимум разности достигается при 𝑚 = 22.

Первоначальное количество ошибок 𝐵 = 𝑚 – 1 = 21.



K = 0.01

Было рассчитано среднее время обнаружения следующих ошибок.



Результат представлен в таблице 25.

Таблица 25 – Расчет времени обнаружения следующих ошибок для релеевского распределения (60%).

|  |  |  |  |
| --- | --- | --- | --- |
| i | 19 | 20 | 21 |
| Xi | 34.727 | 52.09 | 104.18 |

Время до полного завершения тестирования: 190.997

Полное время: 380

1. *РЕЗУЛЬТАТЫ РАСЧЕТОВ*

В таблицах 26 и 27 представлены сводные результаты оценки первоначального числа ошибок и полного времени выполнения тестирования соответственно.

Таблица 26 – Оценка первоначального числа ошибок.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| n | Входные данные, % | Распределение | | |
| Равномерное | Экспоненциальное | Релеевское |
| 30 | 100 | 37 | 30 | 38 |
| 24 | 80 | 28 | 24 | 27 |
| 18 | 60 | 20 | 19 | 21 |

Таблица 27 – Оценка полного времени проведения тестирования

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| n | Входные данные, % | Распределение | | |
| Равномерное | Экспоненциальное | Релеевское |
| 30 | 100 | 755 | 315 | 960 |
| 24 | 80 | 589 | 183 | 369 |
| 18 | 60 | 333 | 262 | 380 |

Результаты при экспоненциальном распределении оказались ниже остальных, 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.