Министерство образования и науки Кыргызской Республики Кыргызско-Германский институт прикладной информатики

**СРСП**

**по теме “Аппроксимация и интерполяции, методы решения ОДУ, математическая статистики.”**

**Вариант №5**

Выполнил:

Студент группы AIN-1-22

Бердикожоев Санжар\_\_\_\_\_\_\_\_\_\_\_\_\_

Проверила:

Доцент КГИПИ

Картанова А.Д.\_\_\_\_\_\_\_\_\_\_\_\_\_

Бишкек 2023

**Оглавление**

[**Цель работы:** 3](#_Toc153057529)

[**Задание:** 3](#_Toc153057530)

[**Инструменты:** 3](#_Toc153057531)

[**1.** **Методы аппроксимации** 4](#_Toc153057532)

[**1.1** **Метод линейной аппроксимации** 6](#_Toc153057533)

[**Описание метода** 6](#_Toc153057534)

[**1.1.1 Исходный код** 7](#_Toc153057535)

[**1.1.2 Результаты в программе** 7](#_Toc153057536)

[**1.1.3 Результаты в таблице Excel** 8](#_Toc153057537)

[**1.2 Метод квадратичной аппроксимации** 10](#_Toc153057538)

[**Описание метода** 10](#_Toc153057539)

[**1.2.2 Результаты в программе** 11](#_Toc153057540)

[**1.2.3 Результаты в таблице Excel** 11](#_Toc153057541)

[**1.3 Метод наименьших квадратов** 13](#_Toc153057542)

[**Описание метода** 13](#_Toc153057543)

[**1.3.1 Исходный код** 14](#_Toc153057544)

[**1.3.2 Результаты в программе** 14](#_Toc153057545)

[15](#_Toc153057546)

[**1.3.3 Результаты в таблице Excel** 15](#_Toc153057547)

[**2.** **Методы интерполяции** 18](#_Toc153057548)

[**2.1 Интерполяция полиномом Лагранжа** 18](#_Toc153057549)

[19](#_Toc153057550)

[**2.1.1 Исходный код** 19](#_Toc153057551)

[**2.1.2 Результаты в программе:** 20](#_Toc153057552)

[**2.1.3 Результаты в таблице Excel:** 20](#_Toc153057553)

[**2.2 Интерполяция полиномом Ньютона** 20](#_Toc153057554)

[**2.1.1 Исходный код:** 21](#_Toc153057555)

[**2.1.2 Результаты в программе:** 22](#_Toc153057556)

[**2.1.3 Результаты в таблице Excel:** 22](#_Toc153057557)

[**3.** **ОДУ и задачи Коши** 22](#_Toc153057558)

[**3.1 Метод Эйлера:** 23](#_Toc153057559)

[**3.1.1 Исходный код:** 23](#_Toc153057560)

[**3.1.2 Результаты программы:** 24](#_Toc153057561)

[**3.1.3 Результаты в таблице Excel:** 24](#_Toc153057562)

[**3.2 Модифицированный метод Эйлера:** 24](#_Toc153057563)

[**3.2.1 Исходный код:** 25](#_Toc153057564)

[**3.2.2 Результаты программы:** 25](#_Toc153057565)

[**3.2.3 Результаты в таблице Excel:** 25](#_Toc153057566)

[**3.3 Метод Рунге-Кутта:** 26](#_Toc153057567)

[**Классический метод Рунге — Кутты четвёртого порядка** 26](#_Toc153057568)

[**3.3.1 Исходный код:** 27](#_Toc153057569)

[**3.3.2 Результаты программы:** 27](#_Toc153057570)

[**3.3.3 Результаты в таблице Excel:** 28](#_Toc153057571)

[**4** **Мат статистика** 30](#_Toc153057572)

[**4.1** **Исходный код:** 31](#_Toc153057573)

[**4.2 Результаты программы** 32](#_Toc153057574)

[**4.3 Таблицы в Excel** 32](#_Toc153057575)

# **Цель работы:**

- Изучение методов аппроксимации и интерполяции.

- Изучение методов решения задачи Коши (методы решения ОДУ).

- Изучение методов решения задач мат. статистики.

# **Задание:**

1. Реализация и тестирование методов аппроксимации и интерполяции. Нахождения с помощью этих методов интерполирующей и аппроксимирующей функций.
2. Реализация и тестирование методов решения задачи Коши.
3. Реализация и тестирование методов решения задач мат. статистики.

# **Инструменты:**

Для выполнения вышеперечисленных заданий в частности использовались:

1. Язык программирования Python для реализации методов аппроксимации и интерполяции, методов решения ОДУ и методов решения задач мат. статистики.
2. Программа Microsoft Excel для аналитического решения

нахождения аппроксимирующей и интерполирующей функций, решения задачи Коши, решения задачи мат. статистики.

1. Программа Microsoft Word для написания и оформления СРСП.

# **Методы аппроксимации**

Даны 4 задачи: 2 для экономических специальностей и 2 для инженерных.

**- Экономическая задача №1**

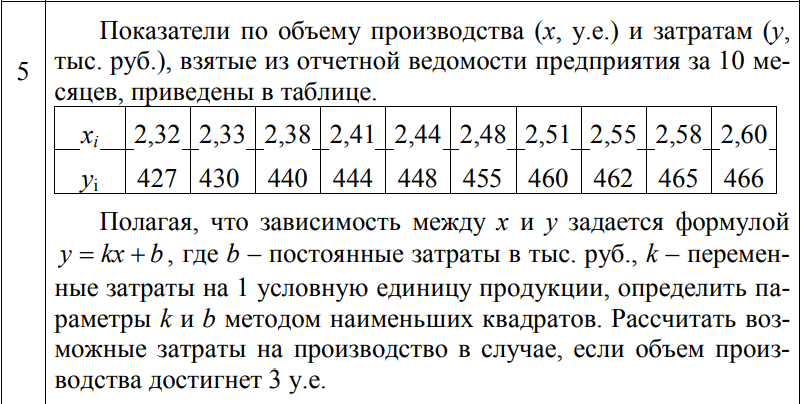


Таблица:

|  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  | 2.32 | 2.33 | 2.38 | 2.41 | 2.44 | 2.48 | 2.51 | 2.55 | 2.58 | 2.60 |
|  | 427 | 430 | 440 | 444 | 448 | 455 | 460 | 462 | 465 | 466 |

**- Экономическая задача №2**

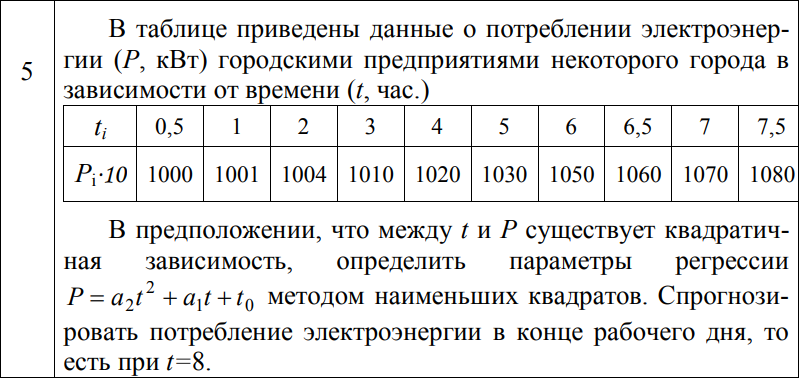


Таблица:

|  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  | 0.5 | 1 | 2 | 3 | 4 | 5 | 6 | 6.5 | 7 | 7.5 |
|  | 1000 | 1001 | 1004 | 1010 | 1020 | 1030 | 1050 | 1060 | 1070 | 1080 |

**- Инженерная задача №1**

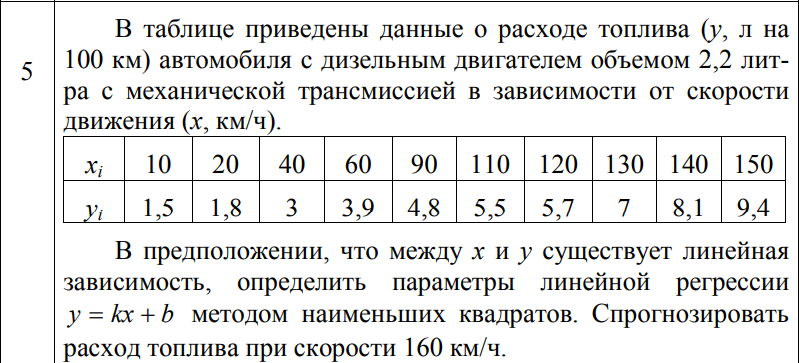


Таблица:

|  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  | 10 | 20 | 40 | 60 | 90 | 110 | 120 | 130 | 140 | 150 |
|  | 1.5 | 1.8 | 3 | 3.9 | 4.8 | 5.5 | 5.7 | 7 | 8.1 | 9.4 |

**- Инженерная задача №2**

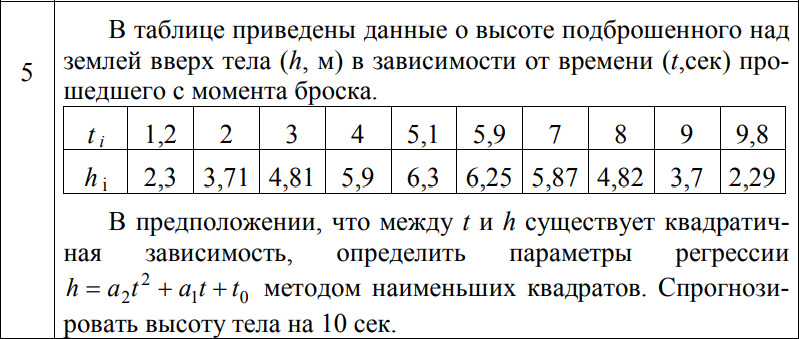


Таблица:

|  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  | 1.2 | 2 | 3 | 4 | 5.1 | 5.9 | 7 | 8 | 9 | 9.8 |
|  | 2.3 | 3.71 | 4.81 | 5.9 | 6.3 | 6.25 | 5.87 | 4.82 | 3.7 | 2.29 |

Необходимо найти аппроксимирующую функцию с помощью методов линейной, квадратичной аппроксимации, а также метода наименьших квадратов.

Для решения на ЯП Python представим таблицы в виде списков.

ecoX\_1: list = [2.32, 2.33, 2.38, 2.41, 2.44, 2.48, 2.51, 2.55, 2.58, 2.60]  
ecoY\_1: list = [427, 430, 440, 444, 448, 455, 460, 462, 465, 466]

ecoX\_2: list = [0.5, 1, 2, 3, 4, 5, 6, 6.5, 7, 7.5]  
ecoY\_2: list = [1000, 1001, 1004, 1010, 1020, 1030, 1050, 1060, 1070, 1080]

engX\_1: list = [10, 20, 40, 60, 90, 110, 120, 130, 140, 150]  
engY\_1: list = [1.5, 1.8, 3, 3.9, 4.8, 5.5, 5.7, 7, 8.1, 9.4]

engX\_2: list = [1.2, 2, 3, 4, 5.1, 5.9, 7, 8, 9, 9.8]  
engY\_2: list = [2.3, 3.71, 4.81, 5.9, 6.3, 6.25, 5.87, 4.82, 3.7, 2.29]

# **Метод линейной аппроксимации**

# **Описание метода**

**Линейное приближение** (линейная аппроксимация) — [приближение](https://ru.wikipedia.org/wiki/%D0%9F%D1%80%D0%B8%D0%B1%D0%BB%D0%B8%D0%B6%D0%B5%D0%BD%D0%B8%D0%B5_(%D0%BC%D0%B0%D1%82%D0%B5%D0%BC%D0%B0%D1%82%D0%B8%D0%BA%D0%B0))

произвольной функции [линейной функцией](https://ru.wikipedia.org/wiki/%D0%9B%D0%B8%D0%BD%D0%B5%D0%B9%D0%BD%D0%B0%D1%8F_%D1%84%D1%83%D0%BD%D0%BA%D1%86%D0%B8%D1%8F). Применяется для [приближённых расчётов](https://ru.wikipedia.org/wiki/%D0%9F%D1%80%D0%B8%D0%B1%D0%BB%D0%B8%D0%B6%D1%91%D0%BD%D0%BD%D1%8B%D0%B5_%D0%B2%D1%8B%D1%87%D0%B8%D1%81%D0%BB%D0%B5%D0%BD%D0%B8%D1%8F), в [методе конечных разностей](https://ru.wikipedia.org/wiki/%D0%9C%D0%B5%D1%82%D0%BE%D0%B4_%D0%BA%D0%BE%D0%BD%D0%B5%D1%87%D0%BD%D1%8B%D1%85_%D1%80%D0%B0%D0%B7%D0%BD%D0%BE%D1%81%D1%82%D0%B5%D0%B9) для решения [дифференциальных уравнений](https://ru.wikipedia.org/wiki/%D0%94%D0%B8%D1%84%D1%84%D0%B5%D1%80%D0%B5%D0%BD%D1%86%D0%B8%D0%B0%D0%BB%D1%8C%D0%BD%D1%8B%D0%B5_%D1%83%D1%80%D0%B0%D0%B2%D0%BD%D0%B5%D0%BD%D0%B8%D1%8F).

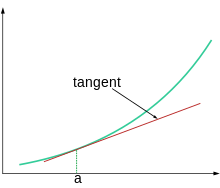
[](https://commons.wikimedia.org/wiki/File:TangentGraphic2.svg?uselang=ru)

График ��∗(�)— касательная в точке (�,�(�))для [непрерывно дифференцируемой](https://ru.wikipedia.org/wiki/%D0%9D%D0%B5%D0%BF%D1%80%D0%B5%D1%80%D1%8B%D0%B2%D0%BD%D0%BE_%D0%B4%D0%B8%D1%84%D1%84%D0%B5%D1%80%D0%B5%D0%BD%D1%86%D0%B8%D1%80%D1%83%D0%B5%D0%BC%D0%B0%D1%8F_%D1%84%D1%83%D0%BD%D0%BA%D1%86%D0%B8%D1%8F) в окрестности точки �[функции](https://ru.wikipedia.org/wiki/%D0%A4%D1%83%D0%BD%D0%BA%D1%86%D0%B8%D1%8F_(%D0%BC%D0%B0%D1%82%D0%B5%D0%BC%D0%B0%D1%82%D0%B8%D0%BA%D0%B0)) вещественной переменной �(�)линейное приближение определяется как:

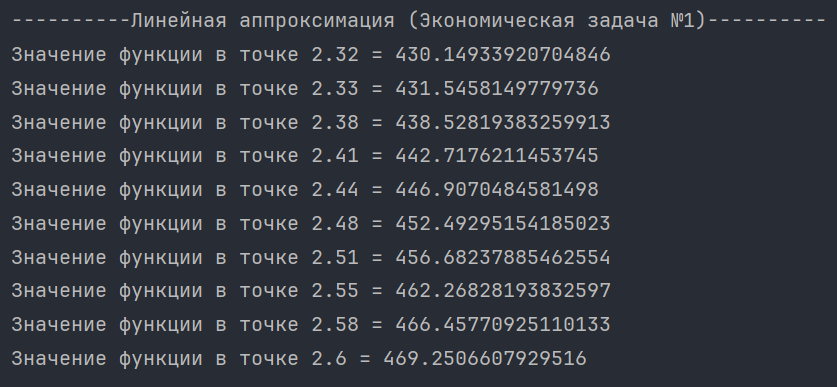
Определение получается из равенства из [теоремы Тейлора](https://ru.wikipedia.org/wiki/%D0%A2%D0%B5%D0%BE%D1%80%D0%B5%D0%BC%D0%B0_%D0%A2%D0%B5%D0%B9%D0%BB%D0%BE%D1%80%D0%B0) �(�)=�(�)+�′(�)(�−�)+�2 игнорированием [остаточного члена](https://ru.wikipedia.org/wiki/%D0%9E%D1%81%D1%82%D0%B0%D1%82%D0%BE%D1%87%D0%BD%D1%8B%D0%B9_%D1%87%D0%BB%D0%B5%D0%BD) �2(�)=�(|�−�|). Поскольку в ближайшей окрестности точки � значения этой функции близки к значениям �(�), её можно использовать как замену значений  �(�) в приближённых вычислениях. При этом в общем случае погрешность возрастает при удалении от � и равна �2. [График функции](https://ru.wikipedia.org/wiki/%D0%93%D1%80%D0%B0%D1%84%D0%B8%D0%BA_%D1%84%D1%83%D0%BD%D0%BA%D1%86%D0%B8%D0%B8) ��∗(�) — [касательная](https://ru.wikipedia.org/wiki/%D0%9A%D0%B0%D1%81%D0%B0%D1%82%D0%B5%D0%BB%D1%8C%D0%BD%D0%B0%D1%8F) к графику  �(�) в точке �.

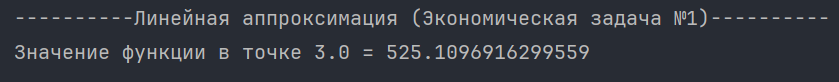
# **1.1.1 Исходный код**

*def* linear\_approximation(x\_values: list[float], y\_values: list[float], x: float):  
 n = len(x\_values)  
  
 mean\_x = sum(x\_values) / n  
 mean\_y = sum(y\_values) / n  
  
 a = sum((x\_values[i] - mean\_x) \* (y\_values[i] - mean\_y) *for* i *in* range(n)) / sum(  
 (x\_values[i] - mean\_x) \*\* 2 *for* i *in* range(n))  
 b = mean\_y - a \* mean\_x  
  
 y\_approximated = a \* x + b  
  
 *return* a, b, y\_approximated

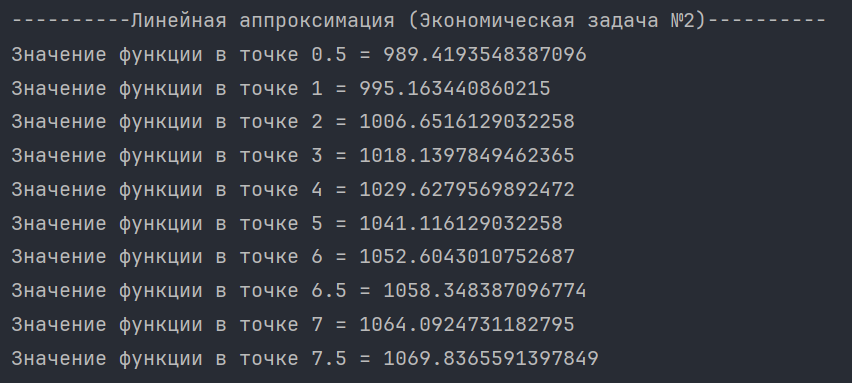
# **1.1.2 Результаты в программе**

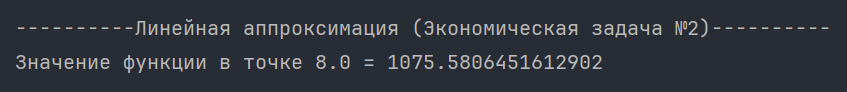
**- Экономическая задача №1**



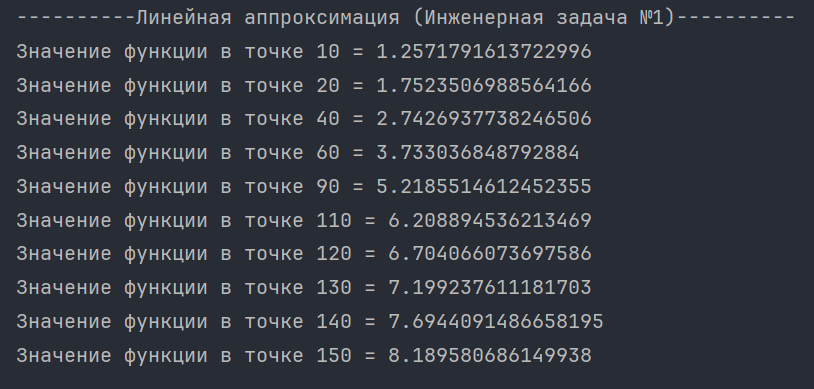


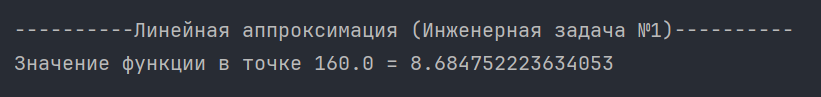
**- Экономическая задача №2**



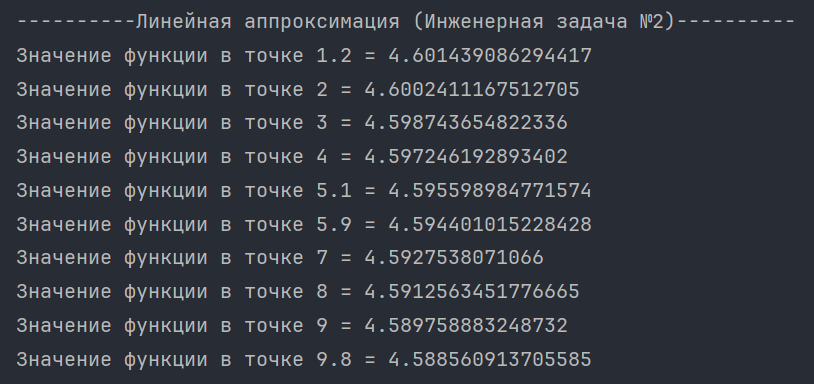


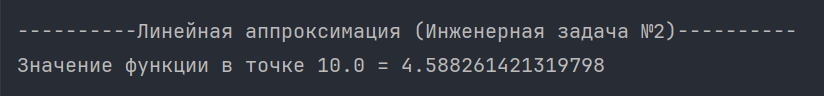
**- Инженерная задача №1**





**- Инженерная задача №2**





# **1.1.3 Результаты в таблице Excel**

**- Экономическая задача №1**

|  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Метод линейной аппроксимации | | | | | | | | | | | | |
| Экономическая задача №1 | | | | | | | | | | | | |
| i | x(i) | y(i) | x(i)^2 | x(i)\*y(i) | x\_ | y\_ | x\_^2 | x\_\*y\_ | a | b | x | y(x) |
| 1 | 2,32 | 427 | 5,38 | 990,64 | -0,14 | -22,7 | 2,9224 | 3,178 | 0,352156 | 448,8337 | 2,32 | 449,6507 |
| 2 | 2,33 | 430 | 5,43 | 1001,9 | -0,13 | -19,7 | 2,9689 | 2,561 |  |  | 2,33 | 449,6542 |
| 3 | 2,38 | 440 | 5,66 | 1047,2 | -0,08 | -9,7 | 3,2044 | 0,776 | x0 | y(x0) | 2,38 | 449,6718 |
| 4 | 2,41 | 444 | 5,81 | 1070,04 | -0,05 | -5,7 | 3,3481 | 0,285 | 3 | 449,8902 | 2,41 | 449,6824 |
| 5 | 2,44 | 448 | 5,95 | 1093,12 | -0,02 | -1,7 | 3,4936 | 0,034 |  |  | 2,44 | 449,693 |
| 6 | 2,48 | 455 | 6,15 | 1128,4 | 0,02 | 5,3 | 3,6904 | 0,106 |  |  | 2,48 | 449,707 |
| 7 | 2,51 | 460 | 6,30 | 1154,6 | 0,05 | 10,3 | 3,8401 | 0,515 |  |  | 2,51 | 449,7176 |
| 8 | 2,55 | 462 | 6,50 | 1178,1 | 0,09 | 12,3 | 4,0425 | 1,107 |  |  | 2,55 | 449,7317 |
| 9 | 2,58 | 465 | 6,66 | 1199,7 | 0,12 | 15,3 | 4,1964 | 1,836 |  |  | 2,58 | 449,7423 |
| 10 | 2,6 | 466 | 6,76 | 1211,6 | 0,14 | 16,3 | 4,3 | 2,282 |  |  | 2,6 | 449,7493 |
| sum | 24,6 | 4497 | 60,61 | 11075,3 | 0 | 1,13687E-13 | 36,0068 | 12,68 |  |  | 24,6 | 4497 |

**- Экономическая задача №2**

|  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Метод линейной аппроксимации | | | | | | | | | | | | |
| Экономическая задача №2 | | | | | | | | | | | | |
| i | x(i) | y(i) | x(i)^2 | x(i)\*y(i) | x\_ | y\_ | x\_^2 | x\_\*y\_ | a | b | x | y(x) |
| 1 | 0,5 | 1000 | 0,25 | 500 | -3,75 | -32,5 | -4 | 121,875 | 3,402548 | 1018,039 | 0,5 | 1019,74 |
| 2 | 1 | 1001 | 1,00 | 1001 | -3,25 | -31,5 | -3,25 | 102,375 |  |  | 1 | 1021,442 |
| 3 | 2 | 1004 | 4,00 | 2008 | -2,25 | -28,5 | -0,25 | 64,125 | x0 | y(x0) | 2 | 1024,844 |
| 4 | 3 | 1010 | 9,00 | 3030 | -1,25 | -22,5 | 4,75 | 28,125 | 8 | 1045,26 | 3 | 1028,247 |
| 5 | 4 | 1020 | 16,00 | 4080 | -0,25 | -12,5 | 11,75 | 3,125 |  |  | 4 | 1031,649 |
| 6 | 5 | 1030 | 25,00 | 5150 | 0,75 | -2,5 | 20,75 | -1,875 |  |  | 5 | 1035,052 |
| 7 | 6 | 1050 | 36,00 | 6300 | 1,75 | 17,5 | 31,75 | 30,625 |  |  | 6 | 1038,454 |
| 8 | 6,5 | 1060 | 42,25 | 6890 | 2,25 | 27,5 | 38 | 61,875 |  |  | 6,5 | 1040,156 |
| 9 | 7 | 1070 | 49,00 | 7490 | 2,75 | 37,5 | 44,75 | 103,125 |  |  | 7 | 1041,857 |
| 10 | 7,5 | 1080 | 56,25 | 8100 | 3,25 | 47,5 | 52 | 154,375 |  |  | 7,5 | 1043,558 |
| sum | 42,5 | 10325 | 238,75 | 44549 | 0 | 0 | 196,25 | 667,75 |  |  | 42,5 | 10325 |

**- Инженерная задача №1**

|  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Метод линейной аппроксимации | | | | | | | | | | | | |
| Инженерная задача №1 | | | | | | | | | | | | |
| i | x(i) | y(i) | x(i)^2 | x(i)\*y(i) | x\_ | y\_ | x\_^2 | x\_\*y\_ | a | b | x | y(x) |
| 1 | 10 | 1,5 | 100,00 | 15 | -77 | -3,57 | 13 | 274,89 | 0,011877 | 4,03666 | 10 | 4,155434 |
| 2 | 20 | 1,8 | 400,00 | 36 | -67 | -3,27 | 313 | 219,09 |  |  | 20 | 4,274209 |
| 3 | 40 | 3 | 1600,00 | 120 | -47 | -2,07 | 1513 | 97,29 | x0 | y(x0) | 40 | 4,511759 |
| 4 | 60 | 3,9 | 3600,00 | 234 | -27 | -1,17 | 3513 | 31,59 | 160 | 5,937056 | 60 | 4,749308 |
| 5 | 90 | 4,8 | 8100,00 | 432 | 3 | -0,27 | 8013 | -0,81 |  |  | 90 | 5,105632 |
| 6 | 110 | 5,5 | 12100,00 | 605 | 23 | 0,43 | 12013 | 9,89 |  |  | 110 | 5,343182 |
| 7 | 120 | 5,7 | 14400,00 | 684 | 33 | 0,63 | 14313 | 20,79 |  |  | 120 | 5,461957 |
| 8 | 130 | 7 | 16900,00 | 910 | 43 | 1,93 | 16813 | 82,99 |  |  | 130 | 5,580731 |
| 9 | 140 | 8,1 | 19600,00 | 1134 | 53 | 3,03 | 19513 | 160,59 |  |  | 140 | 5,699506 |
| 10 | 150 | 9,4 | 22500,00 | 1410 | 63 | 4,33 | 22413 | 272,79 |  |  | 150 | 5,818281 |
| sum | 870 | 50,7 | 99300,00 | 5580 | 0 | 0 | 98430 | 1169,1 |  |  | 870 | 50,7 |

**- Инженерная задача №2**

|  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Метод линейной аппроксимации | | | | | | | | | | | | |
| Инженерная задача №2 | | | | | | | | | | | | |
| i | x(i) | y(i) | x(i)^2 | x(i)\*y(i) | x\_ | y\_ | x\_^2 | x\_\*y\_ | a | b | x | y(x) |
| 1 | 1,2 | 2,3 | 1,44 | 2,76 | -4,3 | -2,295 | -4,06 | 9,8685 | -0,00036 | 4,596989 | 1,2 | 4,596555 |
| 2 | 2 | 3,71 | 4,00 | 7,42 | -3,5 | -0,885 | -1,5 | 3,0975 |  |  | 2 | 4,596266 |
| 3 | 3 | 4,81 | 9,00 | 14,43 | -2,5 | 0,215 | 3,5 | -0,5375 | x0 | y(x0) | 3 | 4,595904 |
| 4 | 4 | 5,9 | 16,00 | 23,6 | -1,5 | 1,305 | 10,5 | -1,9575 | 10 | 4,593373 | 4 | 4,595542 |
| 5 | 5,1 | 6,3 | 26,01 | 32,13 | -0,4 | 1,705 | 20,51 | -0,682 |  |  | 5,1 | 4,595145 |
| 6 | 5,9 | 6,25 | 34,81 | 36,875 | 0,4 | 1,655 | 29,31 | 0,662 |  |  | 5,9 | 4,594855 |
| 7 | 7 | 5,87 | 49,00 | 41,09 | 1,5 | 1,275 | 43,5 | 1,9125 |  |  | 7 | 4,594458 |
| 8 | 8 | 4,82 | 64,00 | 38,56 | 2,5 | 0,225 | 58,5 | 0,5625 |  |  | 8 | 4,594096 |
| 9 | 9 | 3,7 | 81,00 | 33,3 | 3,5 | -0,895 | 75,5 | -3,1325 |  |  | 9 | 4,593734 |
| 10 | 9,8 | 2,29 | 96,04 | 22,442 | 4,3 | -2,305 | 90,54 | -9,9115 |  |  | 9,8 | 4,593445 |
| sum | 55 | 45,95 | 381,30 | 252,607 | 0 | -6,2E-15 | 326,3 | -0,118 |  |  | 55 | 45,95 |

**График**

**- Экономическая задача №1**

**- Экономическая задача №2**

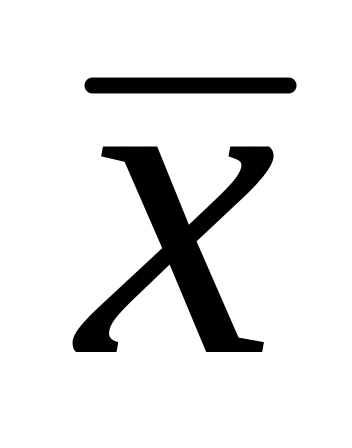
**- Инженерная задача №1**

**- Инженерная задача №2**

# **1.2 Метод квадратичной аппроксимации**

# **Описание метода**

Метод основан на предположении о том, что в ограниченном интервале можно аппроксимировать функцию квадратичным полиномом, который используется для оценивания координаты оптимума. Оценка оптимального значения рассчитывается по формуле:

  = (x2+ x1)/2 - (a1/2a2).

Предполагается, что заданы x1, x2, x3, и известны значения функции в этих точках f1, f2, f3, а аппроксимирующая функция

g(x) = a0+ a1(x - x1) + a2(x - x1)(x - x2)

совпадает с f(x) в трех указанных точках.

Коэффициенты полинома определяются уравнениями

a0= f1; a1= (f2- f1)/(x2- x1); a2= 1/(x3- x2)⋅[(f3- f1)/(x3- x1) - (f2- f1)(x2- x1)].

Для унимодальных функций https://studfile.net/html/2706/633/html_bQeq0B5Nmt.hoJr/htmlconvd-v_tiwU_html_7a5632e39f74b834.gif оказывается приемлемой для оценки оптимума x\*. Алгоритм метода квадратичной аппроксимации.

**Шаг 1.** Задать x1, x2, x3, и вычислить значения функции в этих точках f(х1), f(х2), f(х3).

**Шаг 2.** Рассчитать a0= f(х1); a1= (f(х2)- f(х1))/(x2- x1); a2= 1/(x3- x2)⋅

[(f(х3)- f(х1))/(x3- x1) - (f(х2)- f(х1))(x2- x1)].

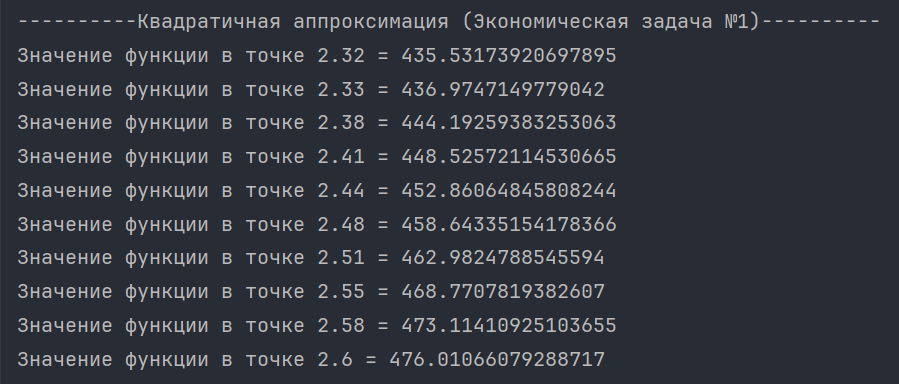
**Шаг 3.** Вычислить оптимальное решение:  = (x2+ x1)/2 - (a1/2a2).

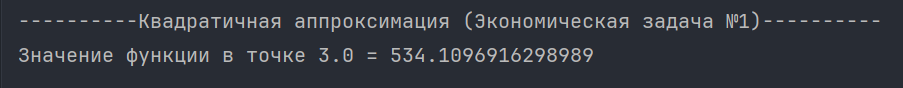
**1.2.1 Исходный код**

*def* quadratic\_approximation(x\_values: list[float], y\_values: list[float], x: float):  
 n = len(x\_values)  
  
 sum\_x = sum(x\_values)  
 sum\_x\_squared = sum(x\_i \*\* 2 *for* x\_i *in* x\_values)  
 sum\_y = sum(y\_values)  
 sum\_x\_y = sum(x\_i \* y\_i *for* x\_i, y\_i *in* zip(x\_values, y\_values))  
  
 a\_denominator = n \* sum\_x\_squared - sum\_x \*\* 2  
 b\_numerator = n \* sum\_x\_y - sum\_x \* sum\_y  
 b\_denominator = n \* sum\_x\_squared - sum\_x \*\* 2  
 c\_numerator = sum\_y \* sum\_x\_squared - sum\_x \* sum\_x\_y  
 c\_denominator = n \* sum\_x\_squared - sum\_x \*\* 2  
  
 a = a\_denominator / b\_denominator  
 b = b\_numerator / b\_denominator  
 c = c\_numerator / c\_denominator  
  
 y\_approximated = a \* x\*\*2 + b \* x + c  
  
 *return* a, b, c, y\_approximated

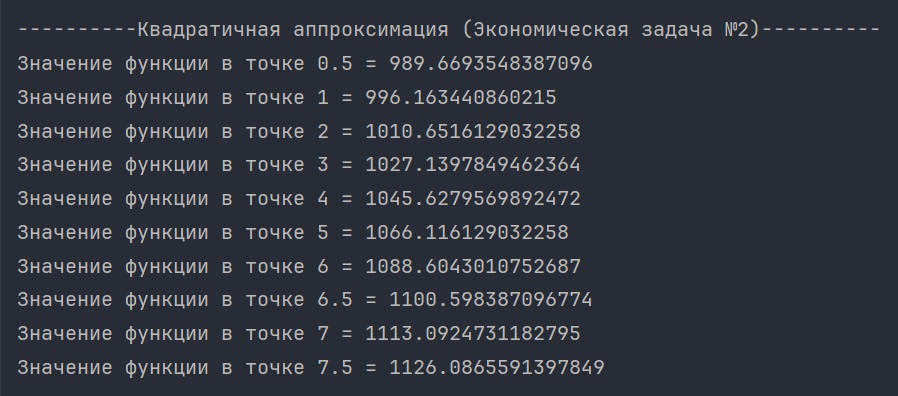
# **1.2.2 Результаты в программе**

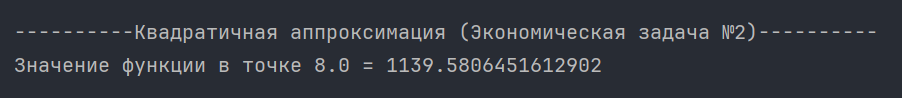
**- Экономическая задача №1**



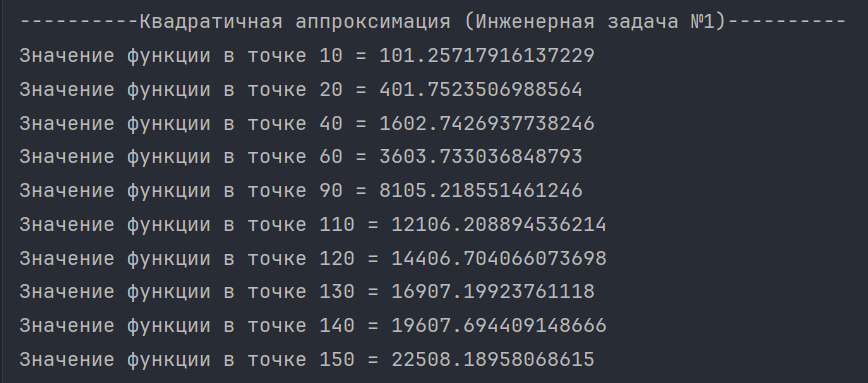


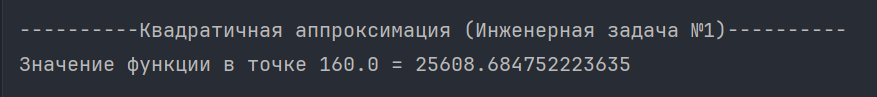
**- Экономическая задача №2**



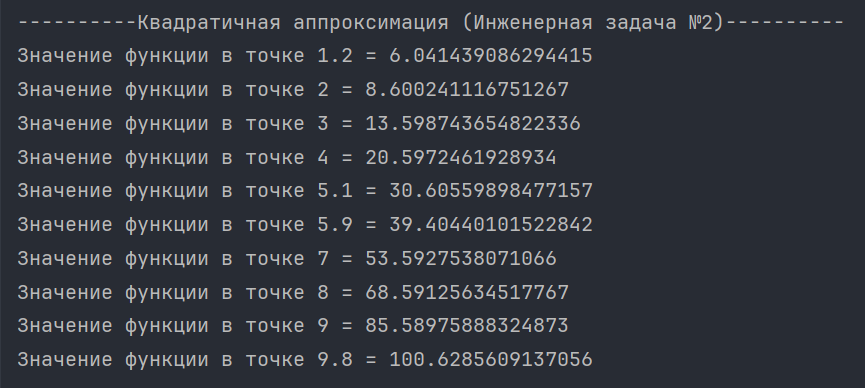


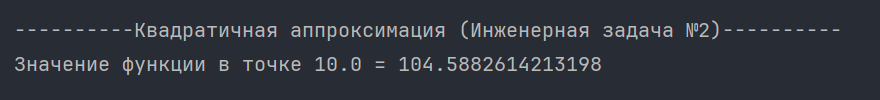
**- Инженерная задача №1**





**- Инженерная задача №2**





# **1.2.3 Результаты в таблице Excel**

**- Экономическая задача №1**

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Метод квадратичной аппроксимации | | | | | | | | | |
| Экономическая задача №1 | | | | | | | | | |
| i | x(i) | y(i) | x(i)^2 | x(i)\*y(i) | denom | b\_num | c\_num | x | y(x) |
| 1 | 2,32 | 427 | 5,38 | 990,64 | 0,908 | 126,8 | 96,3996 | 2,32 | 435,5317 |
| 2 | 2,33 | 430 | 5,43 | 1001,9 |  |  |  | 2,33 | 436,9747 |
| 3 | 2,38 | 440 | 5,66 | 1047,2 | a | b | c | 2,38 | 444,1926 |
| 4 | 2,41 | 444 | 5,81 | 1070,04 | 1 | 139,6475771 | 106,167 | 2,41 | 448,5257 |
| 5 | 2,44 | 448 | 5,95 | 1093,12 |  |  |  | 2,44 | 452,8606 |
| 6 | 2,48 | 455 | 6,15 | 1128,4 |  | x0 | y(x0) | 2,48 | 458,6434 |
| 7 | 2,51 | 460 | 6,30 | 1154,6 |  | 3 | 534,1097 | 2,51 | 462,9825 |
| 8 | 2,55 | 462 | 6,50 | 1178,1 |  |  |  | 2,55 | 468,7708 |
| 9 | 2,58 | 465 | 6,66 | 1199,7 |  |  |  | 2,58 | 473,1141 |
| 10 | 2,6 | 466 | 6,76 | 1211,6 |  |  |  | 2,6 | 476,0107 |
| sum | 24,6 | 4497 | 60,61 | 11075,3 |  |  |  | 24,6 | 4557,607 |

**- Экономическая задача №2**

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Метод квадратичной аппроксимации | | | | | | | | | |
| Экономическая задача №2 | | | | | | | | | |
| i | x(i) | y(i) | x(i)^2 | x(i)\*y(i) | denom | b\_num | c\_num | x | y(x) |
| 1 | 0,5 | 1000 | 0,25 | 500 | 581,25 | 6677,5 | 571761,3 | 0,5 | 989,6694 |
| 2 | 1 | 1001 | 1,00 | 1001 |  |  |  | 1 | 996,1634 |
| 3 | 2 | 1004 | 4,00 | 2008 | a | b | c | 2 | 1010,652 |
| 4 | 3 | 1010 | 9,00 | 3030 | 1 | 11,48817 | 983,6753 | 3 | 1027,14 |
| 5 | 4 | 1020 | 16,00 | 4080 |  |  |  | 4 | 1045,628 |
| 6 | 5 | 1030 | 25,00 | 5150 |  | x0 | y(x0) | 5 | 1066,116 |
| 7 | 6 | 1050 | 36,00 | 6300 |  | 8 | 1139,581 | 6 | 1088,604 |
| 8 | 6,5 | 1060 | 42,25 | 6890 |  |  |  | 6,5 | 1100,598 |
| 9 | 7 | 1070 | 49,00 | 7490 |  |  |  | 7 | 1113,092 |
| 10 | 7,5 | 1080 | 56,25 | 8100 |  |  |  | 7,5 | 1126,087 |
| sum | 42,5 | 10325 | 238,75 | 44549 |  |  |  | 42,5 | 10563,75 |

**- Инженерная задача №1**

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Метод квадратичной аппроксимации | | | | | | | | | |
| Инженерная задача №1 | | | | | | | | | |
| i | x(i) | y(i) | x(i)^2 | x(i)\*y(i) | denom | b\_num | c\_num | x | y(x) |
| 1 | 10 | 1,5 | 100,00 | 15 | 236100 | 11691 | 179910 | 10 | 101,2572 |
| 2 | 20 | 1,8 | 400,00 | 36 |  |  |  | 20 | 401,7524 |
| 3 | 40 | 3 | 1600,00 | 120 | a | b | c | 40 | 1602,743 |
| 4 | 60 | 3,9 | 3600,00 | 234 | 1 | 0,049517154 | 0,762008 | 60 | 3603,733 |
| 5 | 90 | 4,8 | 8100,00 | 432 |  |  |  | 90 | 8105,219 |
| 6 | 110 | 5,5 | 12100,00 | 605 |  | x0 | y(x0) | 110 | 12106,21 |
| 7 | 120 | 5,7 | 14400,00 | 684 |  | 160 | 25608,68 | 120 | 14406,7 |
| 8 | 130 | 7 | 16900,00 | 910 |  |  |  | 130 | 16907,2 |
| 9 | 140 | 8,1 | 19600,00 | 1134 |  |  |  | 140 | 19607,69 |
| 10 | 150 | 9,4 | 22500,00 | 1410 |  |  |  | 150 | 22508,19 |
| sum | 870 | 50,7 | 99300,00 | 5580 |  |  |  | 870 | 99350,7 |

**- Инженерная задача №2**

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Метод квадратичной аппроксимации | | | | | | | | | |
| Инженерная задача №2 | | | | | | | | | |
| i | x(i) | y(i) | x(i)^2 | x(i)\*y(i) | denom | b\_num | c\_num | x | y(x) |
| 1 | 1,2 | 2,3 | 1,44 | 2,76 | 788 | -1,18 | 3627,35 | 1,2 | 6,041439 |
| 2 | 2 | 3,71 | 4,00 | 7,42 |  |  |  | 2 | 8,600241 |
| 3 | 3 | 4,81 | 9,00 | 14,43 | a | b | c | 3 | 13,59874 |
| 4 | 4 | 5,9 | 16,00 | 23,6 | 1 | -0,0015 | 4,603236 | 4 | 20,59725 |
| 5 | 5,1 | 6,3 | 26,01 | 32,13 |  |  |  | 5,1 | 30,6056 |
| 6 | 5,9 | 6,25 | 34,81 | 36,875 |  | x0 | y(x0) | 5,9 | 39,4044 |
| 7 | 7 | 5,87 | 49,00 | 41,09 |  | 10 | 104,5883 | 7 | 53,59275 |
| 8 | 8 | 4,82 | 64,00 | 38,56 |  |  |  | 8 | 68,59126 |
| 9 | 9 | 3,7 | 81,00 | 33,3 |  |  |  | 9 | 85,58976 |
| 10 | 9,8 | 2,29 | 96,04 | 22,442 |  |  |  | 9,8 | 100,6286 |
| sum | 55 | 45,95 | 381,30 | 252,607 |  |  |  | 55 | 427,25 |

**Графики**

**- Экономическая задача №1**

**- Экономическая задача №2**

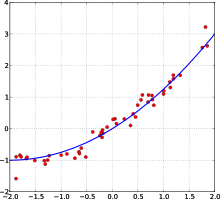
**- Инженерная задача №1**

**- Инженерная задача №2**

# **1.3 Метод наименьших квадратов**

# **Описание метода**

**Метод наименьших квадратов (МНК)** — математический метод, применяемый для решения различных задач, основанный на минимизации суммы квадратов отклонений некоторых функций от экспериментальных входных данных. Он может использоваться для «решения» переопределенных систем уравнений (когда количество уравнений превышает количество неизвестных), для поиска решения в случае обычных (не переопределенных) нелинейных систем уравнений, для аппроксимации точечных значений некоторой функции. МНК является одним из базовых методов [регрессионного анализа](https://ru.wikipedia.org/wiki/%D0%A0%D0%B5%D0%B3%D1%80%D0%B5%D1%81%D1%81%D0%B8%D0%BE%D0%BD%D0%BD%D1%8B%D0%B9_%D0%B0%D0%BD%D0%B0%D0%BB%D0%B8%D0%B7) для оценки неизвестных параметров регрессионных моделей по выборочным данным.

[](https://commons.wikimedia.org/wiki/File:Linear_least_squares(2).svg?uselang=ru)

Пример кривой, проведённой через точки, имеющие нормально распределённое отклонение от истинного значения.

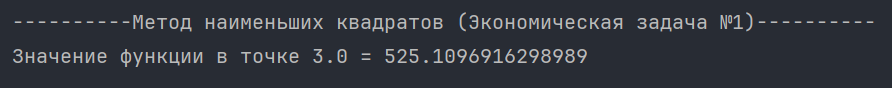
# **1.3.1 Исходный код**

*def* less\_squares(x\_values: list[float], y\_values: list[float], x: float) -> tuple[float] *or None*:  
 *if* len(y\_values) != len(x\_values) *or* len(x\_values) == 0:  
 *return* n: int = len(x\_values)  
 a: float  
 b: float  
  
 xy\_sum: float = 0  
  
 x\_sum: float = 0  
 y\_sum: float = 0  
 x\_sq\_sum: float = 0  
  
 *for* i *in* range(n):  
 xy\_sum += x\_values[i] \* y\_values[i]  
 x\_sum += x\_values[i]  
 y\_sum += y\_values[i]  
 x\_sq\_sum += x\_values[i] \*\* 2  
  
 sq\_sum\_x: float = x\_sum \*\* 2  
  
 a\_numerator: float = n \* xy\_sum - x\_sum \* y\_sum  
 b\_numerator: float = x\_sq\_sum \* y\_sum - x\_sum \* xy\_sum  
 denominator: float = n \* x\_sq\_sum - sq\_sum\_x  
  
 a = a\_numerator / denominator  
 b = b\_numerator / denominator  
  
 y\_approximate = a \* x + b  
  
 *return* a, b, y\_approximate

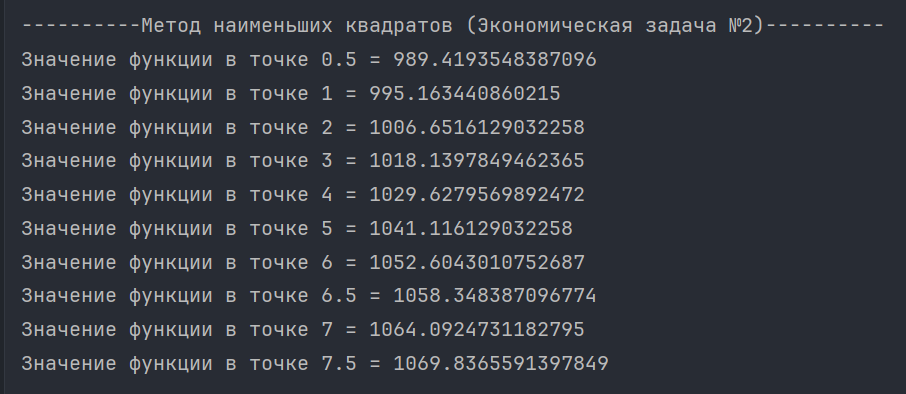
# **1.3.2 Результаты в программе**

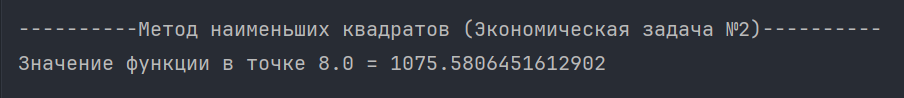
**- Экономическая задача №1**

# 

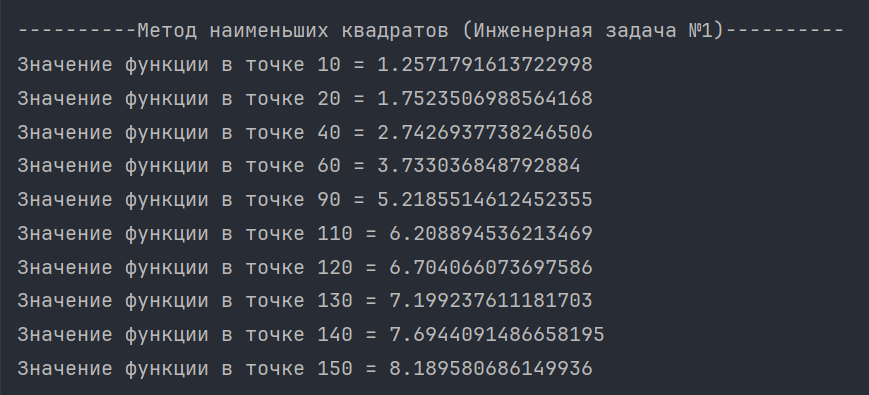


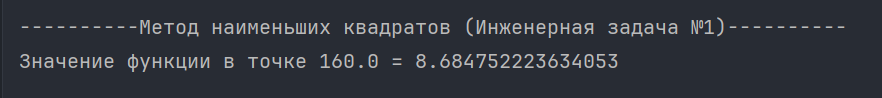
**- Экономическая задача №2**



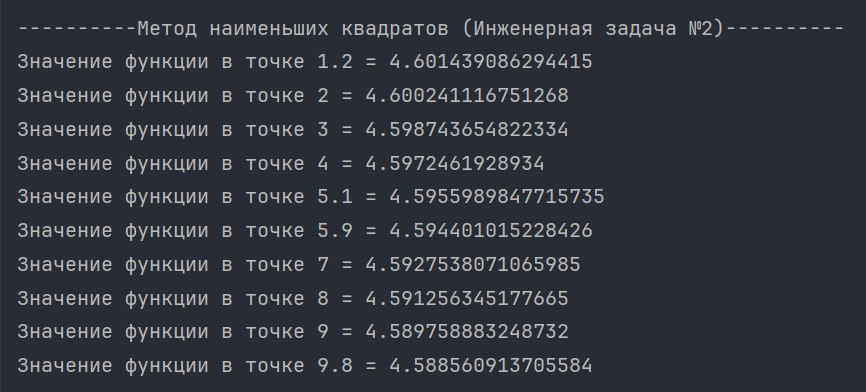


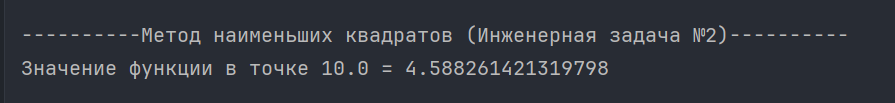
**- Инженерная задача №1**





**- Инженерная задача №2**





# **1.3.3 Результаты в таблице Excel**

**- Экономическая задача №1**

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Метод наименьших квадратов | | | | | | | | |
| Экономическая задача №1 | | | | | | | | |
| i | x(i) | y(i) | x(i)^2 | x(i)\*y(i) | a | b | x | y(x) |
| 1 | 2,32 | 427 | 5,38 | 990,64 | 139,6476 | 106,1669604 | 2,32 | 430,1493 |
| 2 | 2,33 | 430 | 5,43 | 1001,9 |  |  | 2,33 | 431,5458 |
| 3 | 2,38 | 440 | 5,66 | 1047,2 | x0 | y(x0) | 2,38 | 438,5282 |
| 4 | 2,41 | 444 | 5,81 | 1070,04 | 3 | 525,1096916 | 2,41 | 442,7176 |
| 5 | 2,44 | 448 | 5,95 | 1093,12 |  |  | 2,44 | 446,907 |
| 6 | 2,48 | 455 | 6,15 | 1128,4 |  |  | 2,48 | 452,493 |
| 7 | 2,51 | 460 | 6,30 | 1154,6 |  |  | 2,51 | 456,6824 |
| 8 | 2,55 | 462 | 6,50 | 1178,1 |  |  | 2,55 | 462,2683 |
| 9 | 2,58 | 465 | 6,66 | 1199,7 |  |  | 2,58 | 466,4577 |
| 10 | 2,6 | 466 | 6,76 | 1211,6 |  |  | 2,6 | 469,2507 |
| sum | 24,6 | 4497 | 60,61 | 11075,3 |  |  | 24,6 | 3541,497 |

**- Экономическая задача №2**

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Метод наименьших квадратов | | | | | | | | |
| Экономическая задача №2 | | | | | | | | |
| i | x(i) | y(i) | x(i)^2 | x(i)\*y(i) | a | b | x | y(x) |
| 1 | 0,5 | 1000 | 0,25 | 500 | 11,488172 | 983,6753 | 0,5 | 989,4194 |
| 2 | 1 | 1001 | 1,00 | 1001 |  |  | 1 | 995,1634 |
| 3 | 2 | 1004 | 4,00 | 2008 | x0 | y(x0) | 2 | 1006,652 |
| 4 | 3 | 1010 | 9,00 | 3030 | 8 | 1075,581 | 3 | 1018,14 |
| 5 | 4 | 1020 | 16,00 | 4080 |  |  | 4 | 1029,628 |
| 6 | 5 | 1030 | 25,00 | 5150 |  |  | 5 | 1041,116 |
| 7 | 6 | 1050 | 36,00 | 6300 |  |  | 6 | 1052,604 |
| 8 | 6,5 | 1060 | 42,25 | 6890 |  |  | 6,5 | 1058,348 |
| 9 | 7 | 1070 | 49,00 | 7490 |  |  | 7 | 1064,092 |
| 10 | 7,5 | 1080 | 56,25 | 8100 |  |  | 7,5 | 1069,837 |
| sum | 42,5 | 10325 | 238,75 | 44549 |  |  | 42,5 | 1471,923 |

**- Инженерная задача №1**

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Метод наименьших квадратов | | | | | | | | |
| Инженерная задача №1 | | | | | | | | |
| i | x(i) | y(i) | x(i)^2 | x(i)\*y(i) | a | b | x | y(x) |
| 1 | 10 | 1,5 | 100,00 | 15 | 0,049517 | 0,762007624 | 10 | 1,257179 |
| 2 | 20 | 1,8 | 400,00 | 36 |  |  | 20 | 1,752351 |
| 3 | 40 | 3 | 1600,00 | 120 | x0 | y(x0) | 40 | 2,742694 |
| 4 | 60 | 3,9 | 3600,00 | 234 | 160 | 8,684752224 | 60 | 3,733037 |
| 5 | 90 | 4,8 | 8100,00 | 432 |  |  | 90 | 5,218551 |
| 6 | 110 | 5,5 | 12100,00 | 605 |  |  | 110 | 6,208895 |
| 7 | 120 | 5,7 | 14400,00 | 684 |  |  | 120 | 6,704066 |
| 8 | 130 | 7 | 16900,00 | 910 |  |  | 130 | 7,199238 |
| 9 | 140 | 8,1 | 19600,00 | 1134 |  |  | 140 | 7,694409 |
| 10 | 150 | 9,4 | 22500,00 | 1410 |  |  | 150 | 8,189581 |
| sum | 870 | 50,7 | 99300,00 | 5580 |  |  | 870 | 43,84193 |

**- Инженерная задача №2**

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Метод наименьших квадратов | | | | | | | | |
| Инженерная задача №2 | | | | | | | | |
| i | x(i) | y(i) | x(i)^2 | x(i)\*y(i) | a | b | x | y(x) |
| 1 | 1,2 | 2,3 | 1,44 | 2,76 | -0,001497 | 4,603236 | 1,2 | 4,601439 |
| 2 | 2 | 3,71 | 4,00 | 7,42 |  |  | 2 | 4,600241 |
| 3 | 3 | 4,81 | 9,00 | 14,43 | x0 | y(x0) | 3 | 4,598744 |
| 4 | 4 | 5,9 | 16,00 | 23,6 | 10 | 4,588261 | 4 | 4,597246 |
| 5 | 5,1 | 6,3 | 26,01 | 32,13 |  |  | 5,1 | 4,595599 |
| 6 | 5,9 | 6,25 | 34,81 | 36,875 |  |  | 5,9 | 4,594401 |
| 7 | 7 | 5,87 | 49,00 | 41,09 |  |  | 7 | 4,592754 |
| 8 | 8 | 4,82 | 64,00 | 38,56 |  |  | 8 | 4,591256 |
| 9 | 9 | 3,7 | 81,00 | 33,3 |  |  | 9 | 4,589759 |
| 10 | 9,8 | 2,29 | 96,04 | 22,442 |  |  | 9,8 | 4,588561 |
| sum | 55 | 45,95 | 381,30 | 252,607 |  |  | 55 | 4,520876 |

**График**

**- Экономическая задача №1**

**- Экономическая задача №2**

**- Инженерная задача №1**

**- Инженерная задача №2**

# **Методы интерполяции**

**Дано условие задачи:**

Построить интерполяционные полиномы Лагранжа и Ньютона по заданным точкам:

|  |  |  |  |
| --- | --- | --- | --- |
| x | -1 | 4 | 5 |
| y | 2 | 1 | 3 |

# **2.1 Интерполяция полиномом Лагранжа**

**Интерполяционный многочлен Лагранжа** —

[многочлен](https://ru.wikipedia.org/wiki/%D0%9C%D0%BD%D0%BE%D0%B3%D0%BE%D1%87%D0%BB%D0%B5%D0%BD) минимальной степени, принимающий заданные значения в заданном наборе точек, то есть решающий задачу [интерполяции](https://ru.wikipedia.org/wiki/%D0%98%D0%BD%D1%82%D0%B5%D1%80%D0%BF%D0%BE%D0%BB%D1%8F%D1%86%D0%B8%D1%8F).

## [https://upload.wikimedia.org/wikipedia/commons/thumb/5/5a/Lagrange_polynomial.svg/220px-Lagrange_polynomial.svg.png](https://commons.wikimedia.org/wiki/File:Lagrange_polynomial.svg?uselang=ru)

Интерполяционный многочлен Лагранжа для четырёх точек

(-9,5), (4,2), (-1,-2) и (7,9), а также полиномы ����(�), каждый из которых проходит через одну из выделенных точек, и принимает нулевое значение в остальных.

**Определение**

Пусть задана �+1 пара чисел (�0,�0),(�1,�1),…,(��,��), , ,…,, где все �� различны. Требуется построить многочлен �(�) степени не более �, для которого �(��)=��.

**Общий случай**

[Ж. Л. Лагранж](https://ru.wikipedia.org/wiki/%D0%9B%D0%B0%D0%B3%D1%80%D0%B0%D0%BD%D0%B6,_%D0%96%D0%BE%D0%B7%D0%B5%D1%84_%D0%9B%D1%83%D0%B8) предложил следующий способ вычисления таких многочленов: , �(�)=∑�=0�����(�),где базисные полиномы �� определяются по формуле

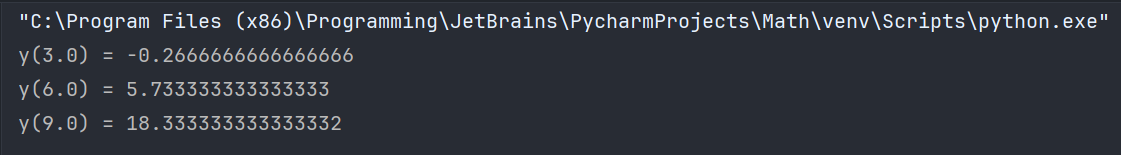
��(�)=∏�=0,�≠���−����−��=�−�0��−�0⋯�−��−1��−��−1⋅�−��+1��−��+1⋯�−����−��Для любого �=0,…,� многочлен �� имеет степень � и ��(��)={0,�≠�,1,�=�.

Отсюда следует, что �(�), являющийся [линейной комбинацией](https://ru.wikipedia.org/wiki/%D0%9B%D0%B8%D0%BD%D0%B5%D0%B9%D0%BD%D0%B0%D1%8F_%D0%BA%D0%BE%D0%BC%D0%B1%D0%B8%D0%BD%D0%B0%D1%86%D0%B8%D1%8F) многочленов ��(�), имеет степень не больше �и �(��)=��.

# **2.1.1 Исходный код**

*def* lagrange(x: list[float], y: list[float], x0: float) -> float | *None*:  
  
 *if* len(x) != len(y) *or* len(x) == 0:  
 *return* n: int = len(x)  
  
 y0: float = 0.0  
  
 *for* i *in* range(n):  
 l: float = y[i]  
 *for* j *in* range(n):  
 *if* i != j:  
 l \*= (x0 - x[j]) / (x[i] - x[j])  
  
 y0 += l  
  
 *return* y0

# **2.1.2 Результаты в программе:**



# **2.1.3 Результаты в таблице Excel:**

|  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Интерполяция полиномом Лагранжа | | | | | | | | | | | |
| i | x(i) | y(i) | l1(x(i)) | l2(x(i)) | l3(x(i)) | y(x(i)) | l1(x(i)) | l2(x(i)) | l3(x(i)) | x | y(x) |
| 1 | -1 | 2 | 1 | 0 | 0 | 2 | 0,066667 | 1,6 | -0,66667 | 3 | -0,26667 |
| 2 | 4 | 1 | 0 | 1 | 0 | 1 | 0,066667 | -1,4 | 2,333333 | 6 | 5,733333 |
| 3 | 5 | 3 | 0 | 0 | 1 | 3 | 0,666667 | -8 | 8,333333 | 9 | 18,33333 |

**График**

# **2.2 Интерполяция полиномом Ньютона**

**Интерполяционные формулы**[**Ньютона**](https://ru.wikipedia.org/wiki/%D0%9D%D1%8C%D1%8E%D1%82%D0%BE%D0%BD,_%D0%98%D1%81%D0%B0%D0%B0%D0%BA) — формулы [вычислительной математики](https://ru.wikipedia.org/wiki/%D0%92%D1%8B%D1%87%D0%B8%D1%81%D0%BB%D0%B8%D1%82%D0%B5%D0%BB%D1%8C%D0%BD%D0%B0%D1%8F_%D0%BC%D0%B0%D1%82%D0%B5%D0%BC%D0%B0%D1%82%D0%B8%D0%BA%D0%B0), применяющиеся для [полиномиального](https://ru.wikipedia.org/wiki/%D0%9C%D0%BD%D0%BE%D0%B3%D0%BE%D1%87%D0%BB%D0%B5%D0%BD) [интерполирования](https://ru.wikipedia.org/wiki/%D0%98%D0%BD%D1%82%D0%B5%D1%80%D0%BF%D0%BE%D0%BB%D1%8F%D1%86%D0%B8%D1%8F).

**Формулы**

Пусть заданы некоторые попарно различные точки �0,�1,…,�� называемые также узлами интерполяции, и известны значения �(�0),�(�1),…,�(��) некоторой функции � в этих точках.

**Случай неравноотстоящих узлов**

Если все расстояния между соседними узлами различны, то многочлен Ньютона строится по формуле , ��(�)=�(�0)+(�−�0)�(�0;�1)+(�−�0)(�−�1)�(�0;�1;�2)+…+(�−�0)…(�−��−1)�(�0;…;��),где �(�0;…;��) — [разделённая разность](https://ru.wikipedia.org/wiki/%D0%A0%D0%B0%D0%B7%D0%B4%D0%B5%D0%BB%D1%91%D0%BD%D0%BD%D0%B0%D1%8F_%D1%80%D0%B0%D0%B7%D0%BD%D0%BE%D1%81%D1%82%D1%8C) порядка �.

**Случай равноотстоящих узлов**

Если соседние узлы находятся друг от друга на некотором фиксированном расстоянии ℎ, то есть ��=�0+�ℎ, �=0,…,� то многочлен Ньютона можно строить либо начиная с �0 (в таком случае говорят об «интерполировании вперёд»), либо с �� («интерполирование назад»).

В первом случае формула для многочлена Ньютона принимает вид

��(�)=�0+�Δ�1+�(�−1)2!Δ2�2+…+�(�−1)…(�−�+1)�!Δ���, , где �=(�−�0)/ℎ,��=�(��), а выражения вида Δ��0— [конечные разности](https://ru.wikipedia.org/wiki/%D0%9A%D0%BE%D0%BD%D0%B5%D1%87%D0%BD%D1%8B%D0%B5_%D1%80%D0%B0%D0%B7%D0%BD%D0%BE%D1%81%D1%82%D0%B8).

Во втором случае формула принимает вид

��(�)=��+�Δ��−1+�(�+1)2!Δ2��−2+…+�(�+1)…(�+�−1)�!Δ��0, , где �=(�−��)/ℎ.

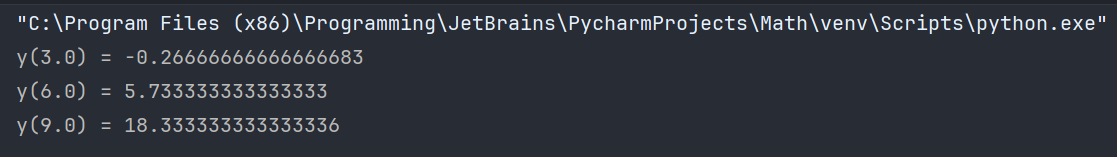
При ℎ=1 справедлива формула ��(�)=∑�=0���−�0�∑�=0�(−1)�−�����(�), где ��� — обобщённые на область действительных чисел [биномиальные коэффициенты](https://ru.wikipedia.org/wiki/%D0%91%D0%B8%D0%BD%D0%BE%D0%BC%D0%B8%D0%B0%D0%BB%D1%8C%D0%BD%D1%8B%D0%B5_%D0%BA%D0%BE%D1%8D%D1%84%D1%84%D0%B8%D1%86%D0%B8%D0%B5%D0%BD%D1%82%D1%8B).

# **2.1.1 Исходный код:**

*def* divided\_differences(x\_values: list[float], y\_values: list[float]) -> list[float]:  
 n = len(x\_values)  
 f = y\_values.copy()  
  
 *for* j *in* range(1, n):  
 *for* i *in* range(n - 1, j - 1, -1):  
 f[i] = (f[i] - f[i - 1]) / (x\_values[i] - x\_values[i - j])  
  
 *return* f

*def* newton\_interpolation(x\_values: list[float], y\_values: list[float], x: float) -> float:  
 n = len(x\_values)  
 f = divided\_differences(x\_values, y\_values)  
 result = f[0]  
  
 *for* i *in* range(1, n):  
 term = f[i]  
 *for* j *in* range(i):  
 term \*= (x - x\_values[j])  
 result += term

# **2.1.2 Результаты в программе:**



# **2.1.3 Результаты в таблице Excel:**

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Интерполяция полиномом Ньютона | | | | | | | | |
| i | x(i) | y(i) | d0(x(i)) | d1(x(i)) | d2(x(i)) | y(x(i)) | x | y(x) |
| 1 | -1 | 2 | 2 | -0,2 | 0,366667 | 2 | 3 | -0,26667 |
| 2 | 4 | 1 | 1 | 2 |  | 1 | 6 | 5,733333 |
| 3 | 5 | 3 | 3 |  |  | 3 | 9 | 18,33333 |

**График**

# **ОДУ и задачи Коши**

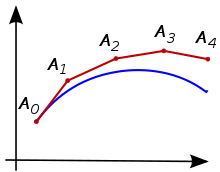
**Дано ОДУ:**

**Функция в питоне:**

*def* differential\_equation(x, y):  
 *return* 5 - y + x

# **3.1 Метод Эйлера:**

**Метод Эйлера** — простейший [численный метод](https://ru.wikipedia.org/wiki/%D0%A7%D0%B8%D1%81%D0%BB%D0%B5%D0%BD%D0%BD%D1%8B%D0%B5_%D0%BC%D0%B5%D1%82%D0%BE%D0%B4%D1%8B) решения систем [обыкновенных дифференциальных уравнений](https://ru.wikipedia.org/wiki/%D0%9E%D0%B1%D1%8B%D0%BA%D0%BD%D0%BE%D0%B2%D0%B5%D0%BD%D0%BD%D1%8B%D0%B5_%D0%B4%D0%B8%D1%84%D1%84%D0%B5%D1%80%D0%B5%D0%BD%D1%86%D0%B8%D0%B0%D0%BB%D1%8C%D0%BD%D1%8B%D0%B5_%D1%83%D1%80%D0%B0%D0%B2%D0%BD%D0%B5%D0%BD%D0%B8%D1%8F). Впервые описан [Леонардом Эйлером](https://ru.wikipedia.org/wiki/%D0%AD%D0%B9%D0%BB%D0%B5%D1%80,_%D0%9B%D0%B5%D0%BE%D0%BD%D0%B0%D1%80%D0%B4) в 1768 году в работе «Интегральное исчисление». Метод Эйлера является явным, одношаговым методом первого порядка точности. Он основан на аппроксимации [интегральной кривой](https://ru.wikipedia.org/wiki/%D0%98%D0%BD%D1%82%D0%B5%D0%B3%D1%80%D0%B0%D0%BB%D1%8C%D0%BD%D0%B0%D1%8F_%D0%BA%D1%80%D0%B8%D0%B2%D0%B0%D1%8F) кусочно-линейной функцией — так называемой ломаной Эйлера.

[](https://commons.wikimedia.org/wiki/File:Euler_method.svg?uselang=ru)

Ломаная Эйлера (красная линия) — приближённое решение в пяти узлах задачи Коши — и точное решение этой задачи (выделено синим цветом)

**Описание метода**

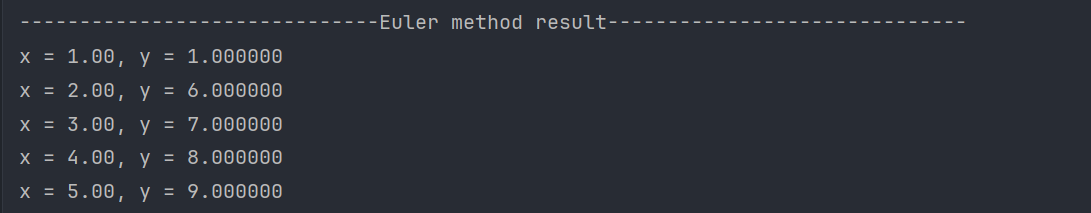
Пусть дана [задача Коши](https://ru.wikipedia.org/wiki/%D0%97%D0%B0%D0%B4%D0%B0%D1%87%D0%B0_%D0%9A%D0%BE%D1%88%D0%B8) для уравнения первого порядка: ,����=�(�,�),�|�=�0=�0,где функция � определена на некоторой области �⊂�2. Решение ищется на полуинтервале (�0,�]. На этом промежутке введём узлы �0<�1<⋯<��≤�.Приближенное решение в узлах ��, которое обозначим через ��, определяется по формуле

Эти формулы непосредственно обобщаются на случай систем обыкновенных дифференциальных уравнений.

# **3.1.1 Исходный код:**

*def* euler\_method(func, y0, x0, xn, h):  
 x\_values = [x0]  
 y\_values = [y0]  
  
 *while* x\_values[-1] < xn:  
 x\_n = x\_values[-1]  
 y\_n = y\_values[-1]  
 y\_n1 = y\_n + h \* func(x\_n, y\_n)  
  
 x\_values.append(x\_n + h)  
 y\_values.append(y\_n1)  
  
 *return* x\_values, y\_values

# **3.1.2 Результаты программы:**



# **3.1.3 Результаты в таблице Excel:**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Метод Эйлера | | | | |
| i | x | y | h | xn |
| 1 | 1 | 1 | 1 | 5 |
| 2 | 2 | 6 |  |  |
| 3 | 3 | 7 |  |  |
| 4 | 4 | 8 |  |  |
| 5 | 5 | 9 |  |  |

**График**

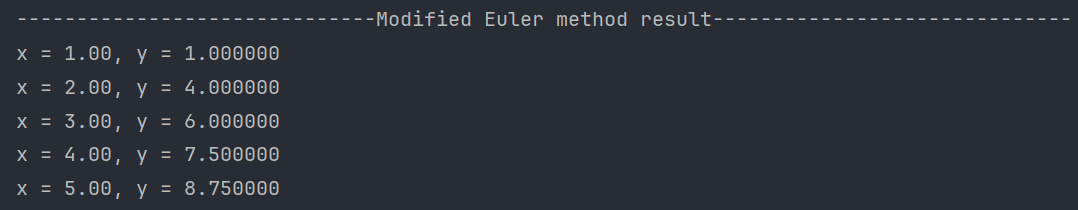
# **3.2 Модифицированный метод Эйлера:**

**Модифицированный метод Эйлера** имеет точность второго порядка. В методе Эйлера производная берется в начале шага и по ней прогнозируется движение системы на конец шага, считая, что во время шага производная неизменна. То есть в течение всего шага производная считается той, какой она была в самом начале шага.

# **3.2.1 Исходный код:**

*def* modified\_euler\_method(f, x0, y0, xn, h):  
 values = []  
  
 x = x0  
 y = y0  
  
 *while* x <= xn:  
 values.append((x, y))  
 y\_pred = y + h \* f(x, y)  
 y = y + h / 2 \* (f(x, y) + f(x + h, y\_pred))  
 x += h  
  
 *return* values

# **3.2.2 Результаты программы:**



# **3.2.3 Результаты в таблице Excel:**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Модифицированный метод Эйлера | | | | | |
| i | x | y | ~y | h | xn |
| 1 | 1 | 1 | 6 | 1 | 5 |
| 2 | 2 | 4 | 7 |  |  |
| 3 | 3 | 6 | 8 |  |  |
| 4 | 4 | 7,5 | 9 |  |  |
| 5 | 5 | 8,75 |  |  |  |

**График**

# **3.3 Метод Рунге-Кутта:**

**Методы Рунге — Кутты** (в литературе встречается название **методы Рунге — Кутта**) — большой класс [численных методов](https://ru.wikipedia.org/wiki/%D0%A7%D0%B8%D1%81%D0%BB%D0%B5%D0%BD%D0%BD%D1%8B%D0%B5_%D0%BC%D0%B5%D1%82%D0%BE%D0%B4%D1%8B) решения [задачи Коши](https://ru.wikipedia.org/wiki/%D0%97%D0%B0%D0%B4%D0%B0%D1%87%D0%B0_%D0%9A%D0%BE%D1%88%D0%B8) для [обыкновенных дифференциальных уравнений](https://ru.wikipedia.org/wiki/%D0%9E%D0%B1%D1%8B%D0%BA%D0%BD%D0%BE%D0%B2%D0%B5%D0%BD%D0%BD%D1%8B%D0%B5_%D0%B4%D0%B8%D1%84%D1%84%D0%B5%D1%80%D0%B5%D0%BD%D1%86%D0%B8%D0%B0%D0%BB%D1%8C%D0%BD%D1%8B%D0%B5_%D1%83%D1%80%D0%B0%D0%B2%D0%BD%D0%B5%D0%BD%D0%B8%D1%8F) и их систем. Первые методы данного класса были предложены около 1900 года немецкими математиками [К. Рунге](https://ru.wikipedia.org/wiki/%D0%A0%D1%83%D0%BD%D0%B3%D0%B5,_%D0%9A%D0%B0%D1%80%D0%BB) и [М. В. Куттой](https://ru.wikipedia.org/wiki/%D0%9C%D0%B0%D1%80%D1%82%D0%B8%D0%BD_%D0%92%D0%B8%D0%BB%D1%8C%D0%B3%D0%B5%D0%BB%D1%8C%D0%BC_%D0%9A%D1%83%D1%82%D1%82%D0%B0).

К классу методов Рунге — Кутты относятся [явный метод Эйлера](https://ru.wikipedia.org/wiki/%D0%9C%D0%B5%D1%82%D0%BE%D0%B4_%D0%AD%D0%B9%D0%BB%D0%B5%D1%80%D0%B0) и модифицированный метод Эйлера с пересчётом, которые представляют собой соответственно методы первого и второго порядка точности. Существуют стандартные явные методы третьего порядка точности, не получившие широкого распространения. Наиболее часто используется и реализован в различных математических пакетах ([Maple](https://ru.wikipedia.org/wiki/Maple" \o "Maple), [MathCAD](https://ru.wikipedia.org/wiki/MathCAD" \o "MathCAD), [Maxima](https://ru.wikipedia.org/wiki/Maxima" \o "Maxima)) *классический метод Рунге — Кутты*, имеющий четвёртый порядок точности.

## **Классический метод Рунге — Кутты четвёртого порядка**

**Метод Рунге — Кутты** четвёртого порядка при вычислениях с постоянным шагом интегрирования столь широко распространён, что его часто называют просто методом Рунге — Кутты.

Рассмотрим [задачу Коши](https://ru.wikipedia.org/wiki/%D0%97%D0%B0%D0%B4%D0%B0%D1%87%D0%B0_%D0%9A%D0%BE%D1%88%D0%B8) для системы обыкновенных дифференциальных уравнений первого порядка (далее **�,�,��∈��, а**�,ℎ∈�1).

**y′=f(�,y),y(�0)=y0.**

Тогда приближенное значение в последующих точках вычисляется по итерационной формуле:

**y�+1=y�+ℎ6(k1+2k2+2k3+k4)**

Вычисление нового значения проходит в четыре стадии:

**k1=f(��,y�),**

k2=f(��+ℎ2,y�+ℎ2k1),

**k1=f(��,y�),**

k3=f(��+ℎ2,y�+ℎ2k2),

**k1=f(��,y�),**

k4=f(��+ℎ,y�+ℎ k3).

**k1=f(��,y�),**

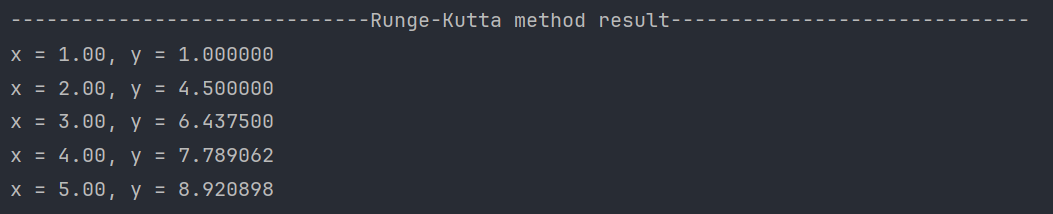
где ℎ — величина шага сетки по �.

Этот метод имеет четвёртый порядок точности. Это значит, что ошибка на одном шаге имеет порядок �(ℎ5), а суммарная ошибка на конечном интервале интегрирования имеет порядок �(ℎ4).

# **3.3.1 Исходный код:**

*def* runge\_kutta(f, x0, y0, xn, h):  
 result = []  
 x = x0  
 y = y0  
  
 *while* x <= xn:  
 result.append((x, y))  
 k1 = h \* f(x, y)  
 k2 = h \* f(x + h / 2, y + k1 / 2)  
 k3 = h \* f(x + h / 2, y + k2 / 2)  
 k4 = h \* f(x + h, y + k3)  
  
 y = y + (k1 + 2 \* k2 + 2 \* k3 + k4) / 6  
 x += h  
  
 *return* result

# **3.3.2 Результаты программы:**



# **3.3.3 Результаты в таблице Excel:**

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Метод Рунге-Кутта | | | | | | | | |
| i | x | y | k0 | k1 | k2 | k3 | h | xn |
| 1 | 1 | 1 | 5 | 3 | 4 | 2 | 1 | 5 |
| 2 | 2 | 4,5 | 2,5 | 1,75 | 2,125 | 1,375 |  |  |
| 3 | 3 | 6,4375 | 1,5625 | 1,28125 | 1,421875 | 1,140625 |  |  |
| 4 | 4 | 7,789063 | 1,210938 | 1,105469 | 1,158203 | 1,052734 |  |  |
| 5 | 5 | 8,920898 | 1,079102 | 1,039551 | 1,059326 | 1,019775 |  |  |

**График**

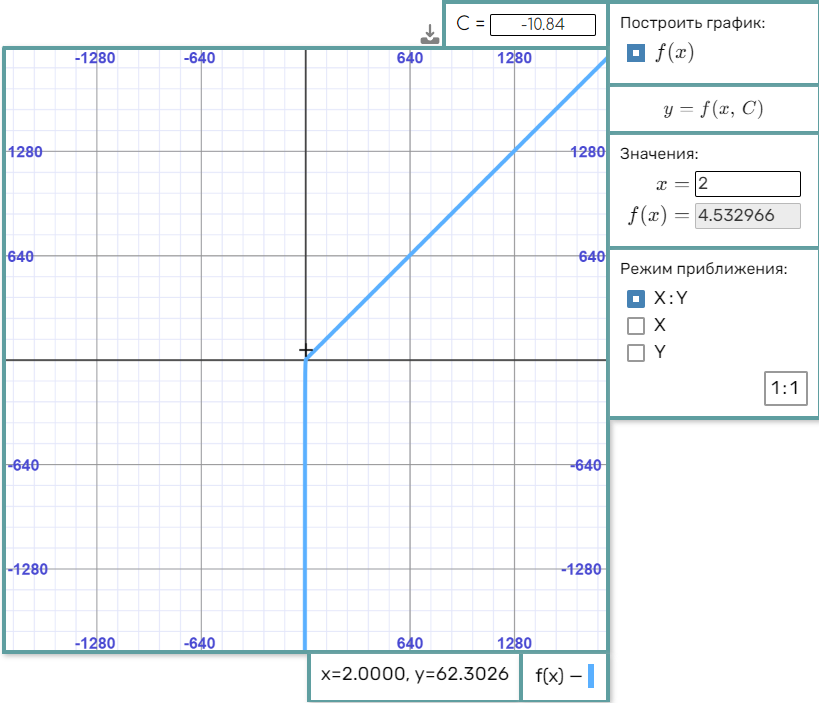
**Итог**

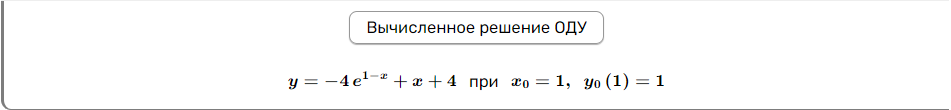
Метод Эйлера является простым и легко реализуемым, но при этом может давать ограниченную точность, особенно при больших шагах интегрирования. Модифицированный метод Эйлера улучшает точность за счет использования средней производной, что делает его более надежным в сравнении с обычным методом Эйлера.

Метод Рунге-Кутта, в свою очередь, представляет собой более сложный, но более точный и устойчивый метод. Его высокий порядок точности делает его предпочтительным для задач, где требуется высокая точность численного решения.

В зависимости от конкретных требований и особенностей задачи, исследователь может выбрать подходящий метод, учитывая баланс между вычислительной сложностью и точностью решения.

**3.4 Погрешности методов решения ОДУ**





1) Для метода Эйлера:

Абсолютная погрешность равна:

|13.5 – 15.9375| ≈ 2.4375 кв. ед.

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Погрешность при методе Эйлера | | | | | | | | |
|  |  |  |  | y(1) | y(2) | y(3) | y(4) | y(5) |
| Точное значение | | | | 1 | 4,528482 | 6,458659 | 7,800852 | 8,926737 |
| Приближенное значение | | | | 1 | 6 | 7 | 8 | 9 |
| Абсолютная погрешность | | | | 0 | 1,471518 | 0,541341 | 0,199148 | 0,073263 |
| Относительная погрешность | | | | 0 | 0,324947 | 0,083816 | 0,025529 | 0,008207 |

Средняя абсолютная погрешность: **0.45705**

Средняя относительная погрешность: **0.0885 или 8.85%**

2) Для модифицированного метода Эйлера:

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Погрешность при модифицированном методе Эйлера | | | | | | | | |
|  |  |  |  | y(1) | y(2) | y(3) | y(4) | y(5) |
| Точное значение | | | | 1 | 4,528482 | 6,458659 | 7,800852 | 8,926737 |
| Приближенное значение | | | | 1 | 4 | 6 | 7,5 | 8,75 |
| Абсолютная погрешность | | | | 0 | 0,528482 | 0,458659 | 0,300852 | 0,176737 |
| Относительная погрешность | | | | 0 | 0,116702 | 0,071015 | 0,038567 | 0,019799 |

Средняя абсолютная погрешность: **0.29295**

Средняя относительная погрешность: **0.04922 или 4.922%**

3) Для метода Рунге-Кутта:

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Погрешность при методе Рунге-Кутта | | | | | | | | |
|  |  |  |  | y(1) | y(2) | y(3) | y(4) | y(5) |
| Точное значение | | | | 1 | 4,528482 | 6,458659 | 7,800852 | 8,926737 |
| Приближенное значение | | | | 1 | 4,5 | 6,4375 | 7,789063 | 8,920898 |
| Абсолютная погрешность | | | | 0 | 0,028482 | 0,021159 | 0,011789 | 0,005839 |
| Относительная погрешность | | | | 0 | 0,00629 | 0,003276 | 0,001511 | 0,000654 |

Средняя абсолютная погрешность: **0.01345**

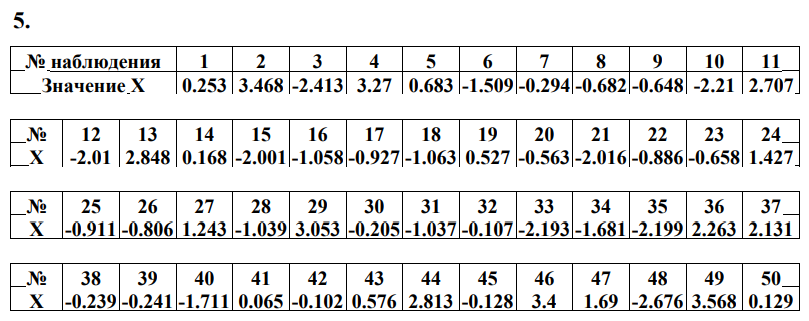
Средняя относительная погрешность: **0.00235 или 0.235%**

**Итог**

Методом с наименьшими абсолютной и относительной погрешностями является метод Рунге-Кутта. Модифицированный метод Эйлера менее точен. Самый неточный метод (алгоритм) это метод Эйлера.

# **Мат статистика**

**Дана задача:**



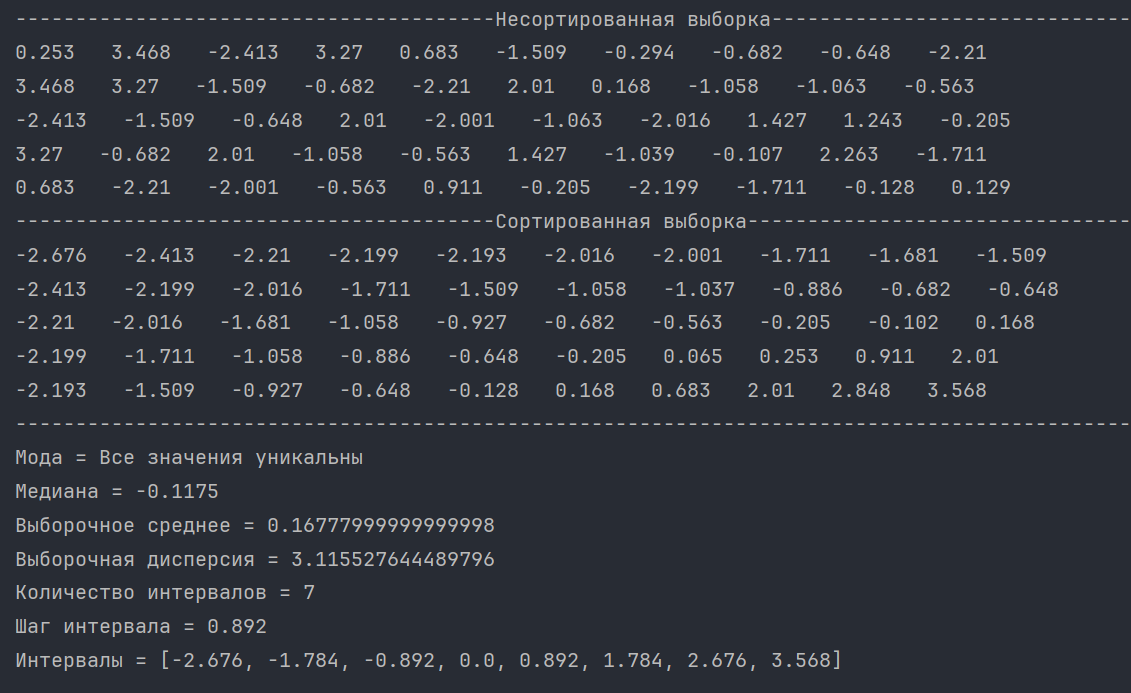
Выборка состоит из 50 значений некоторой случайной величины.

**Найти:** Построить гистограмму, вычислить выборочное среднее, выборочную дисперсию (исправленную), выборочные медиану и моду.

# **Исходный код:**

1. *from* statistics *import* median, variance  
     
   sample = [  
    0.253, 3.468, -2.413, 3.27, 0.683, -1.509, -0.294, -0.682, -0.648, -2.21, 2.707,  
    2.01, 2.848, 0.168, -2.001, -1.058, -0.927, -1.063, 0.527, -0.563, -2.016, -0.886, -0.658, 1.427,  
    0.911, -0.806, 1.243, -1.039, 3.053, -0.205, -1.037, -0.107, -2.193, -1.681, -2.199, 2.263, 2.131,  
    0.239, -0.241, -1.711, 0.065, -0.102, 0.576, 2.813, -0.128, 3.4, 1.69, -2.676, 3.568, 0.129  
   ]  
     
     
   *def* find\_mode(arr: list):  
    counts = [arr.count(arr[i]) *for* i *in* range(len(arr))]  
    *for* el *in* counts:  
    *if* el > 1:  
    *return* el  
    *else*:  
    *return* "Все значения уникальны"  
     
     
   max\_el = max(sample)  
   min\_el = min(sample)  
   sorted\_sample = sorted(sample)  
     
   num\_interval = 7  
   height = max\_el - min\_el  
   step\_interval = height / num\_interval  
     
   intervals = [round(min\_el + i \* step\_interval, 4) *for* i *in* range(0, num\_interval + 1)]  
     
     
   mode\_ = find\_mode(sample)  
   median\_ = median(sample)  
   variance\_ = variance(sample)  
   average\_ = sum(sample) / len(sample)  
     
     
   print("-" \* 40 + "Несортированная выборка" + "-" \* 40)  
   *for* i *in* range(5):  
    *for* j *in* range(10):  
    print(round(sample[i + (i + 1) \* j], 4), end=" ")  
    print()  
     
   print("-" \* 40 + "Сортированная выборка" + "-" \* 40)  
   *for* i *in* range(5):  
    *for* j *in* range(10):  
    print(round(sorted\_sample[i + (i + 1) \* j], 4), end=" ")  
    print()  
     
   print("-" \* 120)  
   print(f"Мода = {mode\_}")  
   print(f"Медиана = {median\_}")  
   print(f"Выборочное среднее = {average\_}")  
   print(f"Выборочная дисперсия = {variance\_}")  
   print(f"Количество интервалов = {num\_interval}")  
   print(f"Шаг интервала = {step\_interval}")  
   print(f"Интервалы = {intervals}")

# **4.2 Результаты программы**



# **4.3 Таблицы в Excel**

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| № | X | Сорт. X |  | Максимум | | |  | Макс-Мин | |
| 1 | 0,253 | -2.676 |  | 3,568 | | |  | 6,244 | |
| 2 | 3,468 | -2.413 |  |  |  |  |  |  |  |
| 3 | -2,413 | -2.21 |  | Минимум | | |  | Кол-во интервалов | |
| 4 | 3,27 | -2.199 |  | -2,676 | | |  | 7 | |
| 5 | 0,683 | -2.193 |  |  |  |  |  |  |  |
| 6 | -1,509 | -2.016 |  | Выборочное среднее | | |  | Шаг интервала | |
| 7 | -0,294 | -2.001 |  | 0,16778 | | |  | 0,892 | |
| 8 | -0,682 | -1.711 |  |  |  |  |  |  |  |
| 9 | -0,648 | -1.681 |  | Выборочная дисперсия | | |  | Интервалы | |
| 10 | -2,21 | -1.509 |  | 3,115527644 | | |  | Начало | Конец |
| 11 | 2,707 | -1.063 |  |  |  |  |  | -2,676 | -1,784 |
| 12 | 2,01 | -1.058 |  | Мода | | |  | -1,784 | -0,892 |
| 13 | 2,848 | -1.039 |  | #Н/Д | | |  | -0,892 | 0 |
| 14 | 0,168 | -1.037 |  |  |  |  |  | 0 | 0,892 |
| 15 | -2,001 | -0.927 |  | Медиана | | |  | 0,892 | 1,784 |
| 16 | -1,058 | -0.886 |  | -0,1175 | | |  | 1,784 | 2,676 |
| 17 | -0,927 | -0.806 |  |  |  |  |  | 2,676 | 3,568 |
| 18 | -1,063 | -0.682 |  |  |  |  |  |  |  |
| 19 | 0,527 | -0.658 |  |  |  |  |  |  |  |
| 20 | -0,563 | -0.648 |  |  |  |  |  |  |  |
| 21 | -2,016 | -0.563 |  |  |  |  |  |  |  |
| 22 | -0,886 | -0.294 |  |  |  |  |  |  |  |
| 23 | -0,658 | -0.241 |  |  |  |  |  |  |  |
| 24 | 1,427 | -0.205 |  |  |  |  |  |  |  |
| 25 | 0,911 | -0.128 |  |  |  |  |  |  |  |
| 26 | -0,806 | -0.107 |  |  |  |  |  |  |  |
| 27 | 1,243 | -0.102 |  |  |  |  |  |  |  |
| 28 | -1,039 | 0.065 |  |  |  |  |  |  |  |
| 29 | 3,053 | 0.129 |  |  |  |  |  |  |  |
| 30 | -0,205 | 0.168 |  |  |  |  |  |  |  |
| 31 | -1,037 | 0.239 |  |  |  |  |  |  |  |
| 32 | -0,107 | 0.253 |  |  |  |  |  |  |  |
| 33 | -2,193 | 0.527 |  |  |  |  |  |  |  |
| 34 | -1,681 | 0.576 |  |  |  |  |  |  |  |
| 35 | -2,199 | 0.683 |  |  |  |  |  |  |  |
| 36 | 2,263 | 0.911 |  |  |  |  |  |  |  |
| 37 | 2,131 | 1.243 |  |  |  |  |  |  |  |
| 38 | 0,239 | 1.427 |  |  |  |  |  |  |  |
| 39 | -0,241 | 1,69 |  |  |  |  |  |  |  |
| 40 | -1,711 | 2,01 |  |  |  |  |  |  |  |
| 41 | 0,065 | 2.131 |  |  |  |  |  |  |  |
| 42 | -0,102 | 2.263 |  |  |  |  |  |  |  |
| 43 | 0,576 | 2.707 |  |  |  |  |  |  |  |
| 44 | 2,813 | 2.813 |  |  |  |  |  |  |  |
| 45 | -0,128 | 2.848 |  |  |  |  |  |  |  |
| 46 | 3,4 | 3.053 |  |  |  |  |  |  |  |
| 47 | 1,69 | 3,27 |  |  |  |  |  |  |  |
| 48 | -2,676 | 3,4 |  |  |  |  |  |  |  |
| 49 | 3,568 | 3.468 |  |  |  |  |  |  |  |
| 50 | 0,129 | 3.568 |  |  |  |  |  |  |  |

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| Частоты |  |  | Отн. частоты | |  | Плотность | |
| 7 |  |  | 0,14 | |  | 0,14 | |
| 8 |  |  | 0,16 | |  | 0,16 | |
| 12 |  |  | 0,24 | |  | 0,24 | |
| 8 |  |  | 0,16 | |  | 0,16 | |
| 4 |  |  | 0,08 | |  | 0,08 | |
| 3 |  |  | 0,06 | |  | 0,06 | |
| 8 |  |  | 0,16 | |  | 0,16 | |
|  |  |  |  |  |  |  |  |
| Сумма частот | |  | Сумма отн. Частот | |  |  |  |
| 50 | |  | 1 | |  |  |  |

**Гистограмма**