

Федеральное государственное автономное образовательное учреждение высшего образования «**Национальный исследовательский университет ИТМО**»

Факультет Программной Инженерии и Компьютерной Техники

Лабораторная работа №5
«Интерполяция функции»

по дисциплине «Вычислительная математика»

Вариант: 13

Преподаватель:
Наумова Надежда Александровна

Выполнил:
Саранча Павел Александрович
Группа: P3209

Санкт-Петербург, 2025 г.

Цель работы: решить задачу интерполяции, найти значения функции при заданных значениях аргумента, отличных от узловых точек.

1. Вычислительная реализация задачи

1. Выбрать таблицу $y = f(x)$:

	x	y	N варианта	X ₁	X ₂
Таблица 1.3	1.10	0.2234	13	1.168	1.463
	1.25	1.2438			
	1.40	2.2644			
	1.55	3.2984			
	1.70	4.3222			
	1.85	5.3516			
	2.00	6.3867			

2. Построить таблицу конечных разностей:

№	x _i	y _i	Δy _i	Δ ² y _i	Δ ³ y _i	Δ ⁴ y _i	Δ ⁵ y _i	Δ ⁶ y _i
0.	1,1	0,2234	1,0204	0,0002	0,0132	-0,0368	0,0762	-0,1313
1.	1,25	1,2438	1,0206	0,0134	-0,0236	0,0394	-0,0551	
2.	1,4	2,2644	1,034	-0,0102	0,0158	-0,0157		
3.	1,55	3,2984	1,0238	0,0056	0,0001			
4.	1,7	4,3222	1,0294	0,0057				
5.	1,85	5,3516	1,0351					
6.	2	6,3867						

3. Вычислить значения функции для аргумента X₁, используя первую или вторую интерполяционную формулу **Ньютона**:

Воспользуемся формулой Ньютона для интерполирования **вперед**, так как X₁ = 1.168 лежит в левой половине отрезка.

$$\text{Для } X_1 = 1.168: t = \frac{(x-x_n)}{h} = \frac{(1.168-1.1)}{0.15} = 0.453$$

$$N_6(x) = y_0 + t\Delta y_0 + \frac{t(t-1)}{2!}\Delta^2 y_0 + \frac{t(t-1)(t-2)}{3!}\Delta^3 y_0 + \frac{t(t-1)(t-2)(t-3)}{4!}\Delta^4 y_0 + \frac{t(t-1)(t-2)(t-3)(t-4)}{5!}\Delta^5 y_0 + \frac{t(t-1)(t-2)(t-3)(t-4)(t-5)}{6!}\Delta^6 y_0$$

$$\begin{aligned}
y(1.168) \approx & 0.2234 + 0.453 * 1.0204 + \frac{0.453(0.453 - 1)}{2} * 0.0002 \\
& + \frac{0.453(0.453 - 1)(0.453 - 2)}{6} * 0.0132 \\
& + \frac{0.453(0.453 - 1)(0.453 - 2)(0.453 - 3)}{24} * (-0.0368) \\
& + \frac{0.453(0.453 - 1)(0.453 - 2)(0.453 - 3)(0.453 - 4)}{120} * 0.0762 \\
& + \frac{0.453(0.453 - 1)(0.453 - 2)(0.453 - 3)(0.453 - 4)(0.453 - 5)}{720} \\
& * (-0.1313)
\end{aligned}$$

$$y(1.168) \approx 0.69337$$

4. Вычислить значения функции для аргумента X_2 , используя первую или вторую интерполяционную формулу Гаусса:

Центральная точка $a = 0.65$, $X_2 = 0.645 < 0.65$, то есть $x < a \rightarrow$ используем **вторую** интерполяционную формулу Гаусса.

$$t = \frac{(x - x_0)}{h} = \frac{(1.463 - 1.55)}{0.15} = -0.58$$

$$\begin{aligned}
P_6(x) = & y_0 + t\Delta y_{-1} + \frac{t(t+1)}{2!}\Delta^2 y_{-1} + \frac{t(t+1)(t-1)}{3!}\Delta^3 y_{-2} \\
& + \frac{t(t+1)(t-1)(t+2)}{4!}\Delta^4 y_{-2} + \frac{t(t+1)(t-1)(t+2)(t-2)}{5!}\Delta^5 y_{-3} \\
& + \frac{t(t+1)(t-1)(t+2)(t-2)(t+3)}{6!}\Delta^6 y_{-3}
\end{aligned}$$

$$\begin{aligned}
y(1.463) \approx & 3.2984 + (-0.58) * 1.034 + \frac{-0.58(0.42)}{2} * (-0.0102) + \frac{(-0.58)(0.42)(-1.58)}{6} \\
& * (-0.0236) + \frac{(-0.58)(0.42)(-1.58)(1.42)}{24} * (0.0394) \\
& + \frac{(-0.58)(0.42)(1.42)(-2.58)}{120} * (0.0762) \\
& + \frac{(-0.58)(0.42)(-1.58)(1.42)(-2.58)(2.42)}{720} * (-0.1313)
\end{aligned}$$

$$y(1.463) \approx 2.69903$$

2. Программная реализация задачи

https://github.com/PaulLocust/comp_math_lab5

Листинг методов:

```
1 | def lagrange_polynomial(xs, ys, n, x):
2 |     """
3 |     Интерполяция по методу Лагранжа.
4 |
5 |     Входные параметры:
6 |     xs : массив координат узлов интерполяции
7 |     ys : массив значений функции в узлах
8 |     n  : количество узлов
9 |     x  : точка, в которой вычисляем значение интерполированного многочлена
10|
11|     Возвращает:
12|     значение интерполированного многочлена в точке x
13|     """
14|     total = 0
15|     for i in range(n):
16|         product = 1
17|         for j in range(n):
18|             if i != j:
19|                 product *= (x - xs[j]) / (xs[i] - xs[j])
20|         total += ys[i] * product
21|     return total
```

```
1 | def newton_divided_difference_polynomial(xs, ys, n, x):
2 |     """
3 |     Интерполяция многочленом Ньютона с использованием разделённых разностей.
4 |
5 |     Входные параметры:
6 |     xs : массив координат узлов интерполяции
7 |     ys : массив значений функции в узлах
8 |     n  : количество узлов
9 |     x  : точка, в которой вычисляем значение интерполированного многочлена
10|
11|     Возвращает:
12|     значение интерполированного многочлена в точке x
13|     """
14|     coef = divided_differences(xs, ys)
15|     total = ys[0]
16|     for k in range(1, n):
17|         product = 1
18|         for j in range(k):
19|             product *= x - xs[j]
20|         total += coef[k] * product
21|     return total
22|
```

```
1 | def newton_finite_difference_polynomial(xs, ys, n, x):
2 |     """
3 |     Интерполяция многочленом Ньютона с использованием конечных разностей.
4 |
5 |     Входные параметры:
6 |     xs : массив координат узлов интерполяции (равномерно расположенных)
7 |     ys : массив значений функции в узлах
8 |     n  : количество узлов
9 |     x  : точка, в которой вычисляем значение интерполированного многочлена
10|
11|     Возвращает:
12|     значение интерполированного многочлена в точке x
13|     """
14|
15|     h = xs[1] - xs[0] # шаг
16|     delta_y = finite_differences(ys)
17|     total = ys[0]
18|     for k in range(1, n):
19|         product = 1
20|         for j in range(k):
21|             product *= (x - xs[0]) / h - j
22|         total += delta_y[0][k] * product / factorial(k)
23|     return total
```

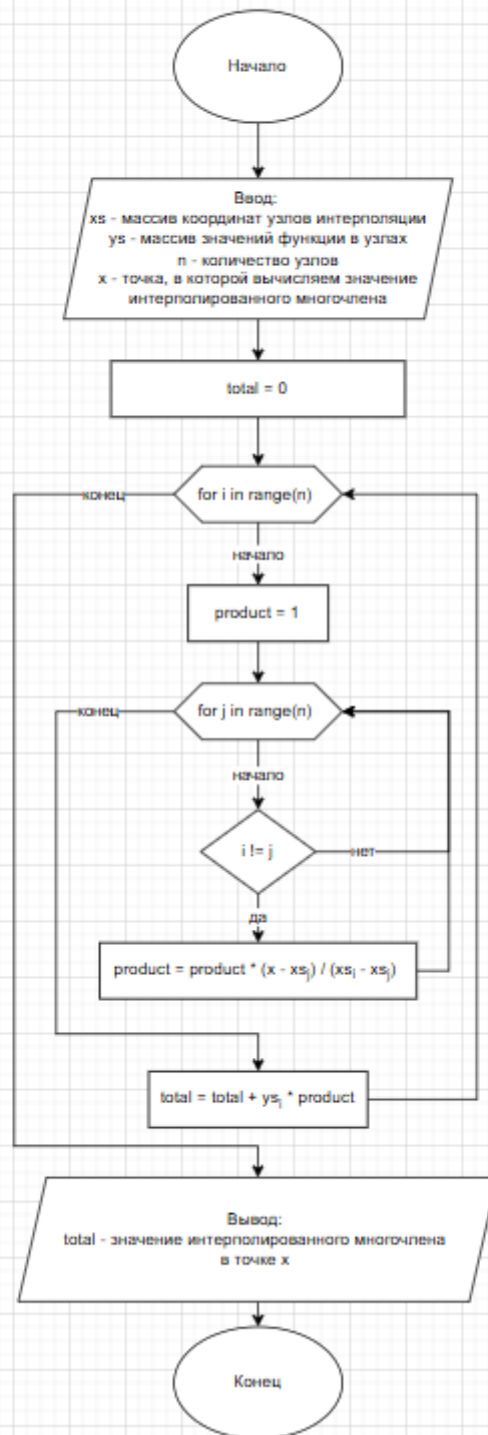
```

1 | def gauss_polynomial(xs, ys, n, x):
2 |     """
3 |     Интерполяция многочленом Гаусса с использованием центральных конечных разностей.
4 |
5 |     Входные параметры:
6 |     xs : массив координат узлов интерполяции (равномерно расположенных)
7 |     ys : массив значений функции в узлах
8 |     n : количество узлов
9 |     x : точка, в которой вычисляем значение интерполированного многочлена
10 |
11 |     Возвращает:
12 |     значение интерполированного многочлена в точке x
13 |     """
14 |
15 |     m = len(xs)
16 |     alpha_ind = (m - 1) // 2 # центральный узел
17 |
18 |     # Построение таблицы конечных разностей
19 |     fin_difs = []
20 |     fin_difs.append(ys[:])
21 |     for k in range(1, m):
22 |         prev = fin_difs[-1]
23 |         current = []
24 |         for i in range(len(prev) - 1):
25 |             current.append(prev[i + 1] - prev[i])
26 |         fin_difs.append(current)
27 |
28 |     h = xs[1] - xs[0]
29 |     t = (x - xs[alpha_ind]) / h
30 |     total = ys[alpha_ind]
31 |
32 |     dts = [0, 1, -1, 2, -2, 3, -3, 4, -4] # порядок множителей для произведения
33 |     for k in range(1, n):
34 |         product = 1
35 |         for j in range(k):
36 |             product *= t + dts[j]
37 |         mid_index = len(fin_difs[k]) // 2
38 |         if len(fin_difs[k]) % 2 == 0:
39 |             index = mid_index - 1
40 |         else:
41 |             index = mid_index
42 |         total += fin_difs[k][index] * product / factorial(k)
43 |
44 |     return total
45 |

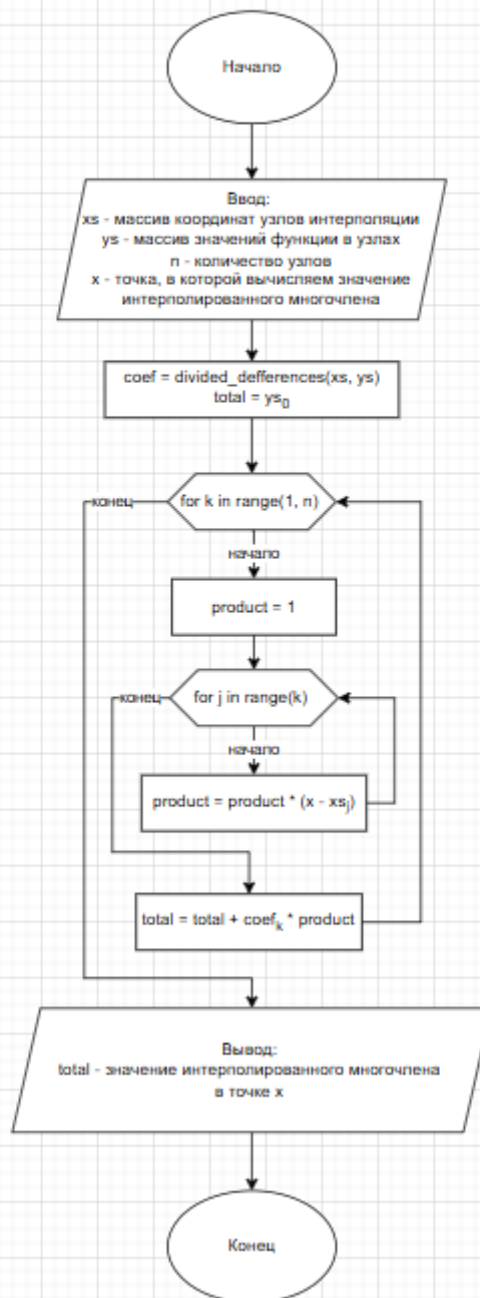
```

Логические схемы:

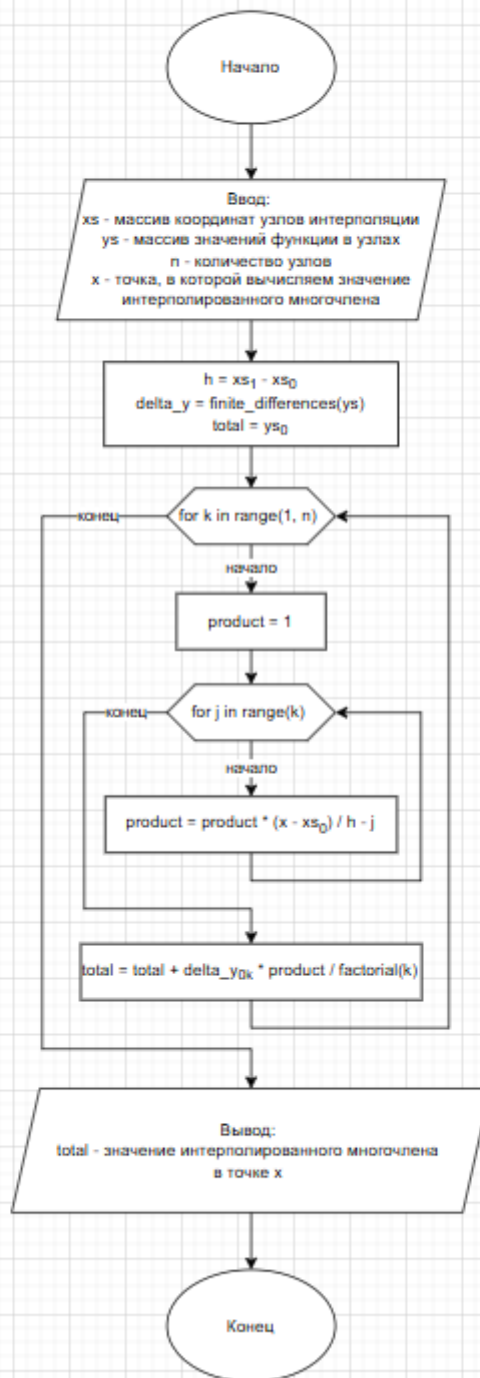
Многочлен Лагранжа



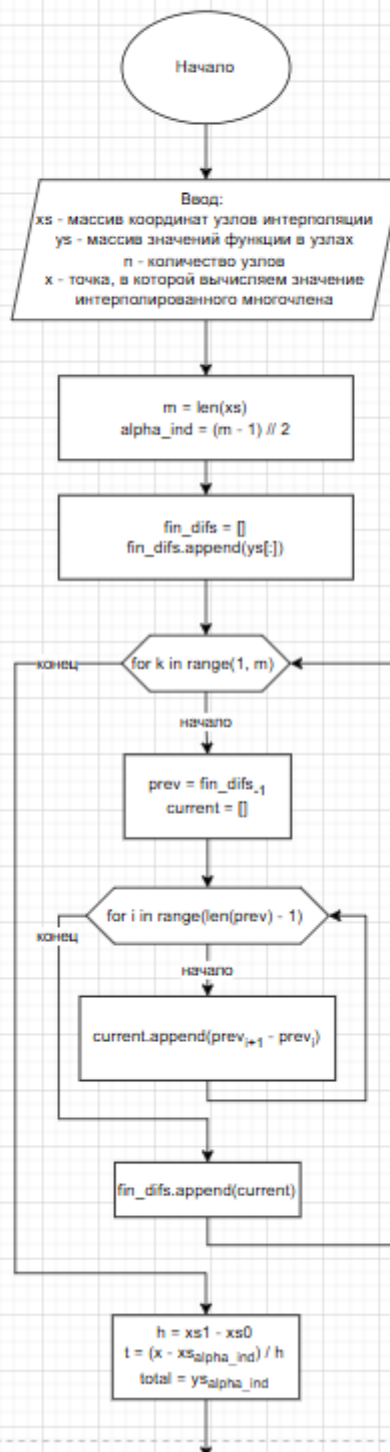
Многочлен Ньютона (разделённые разности)

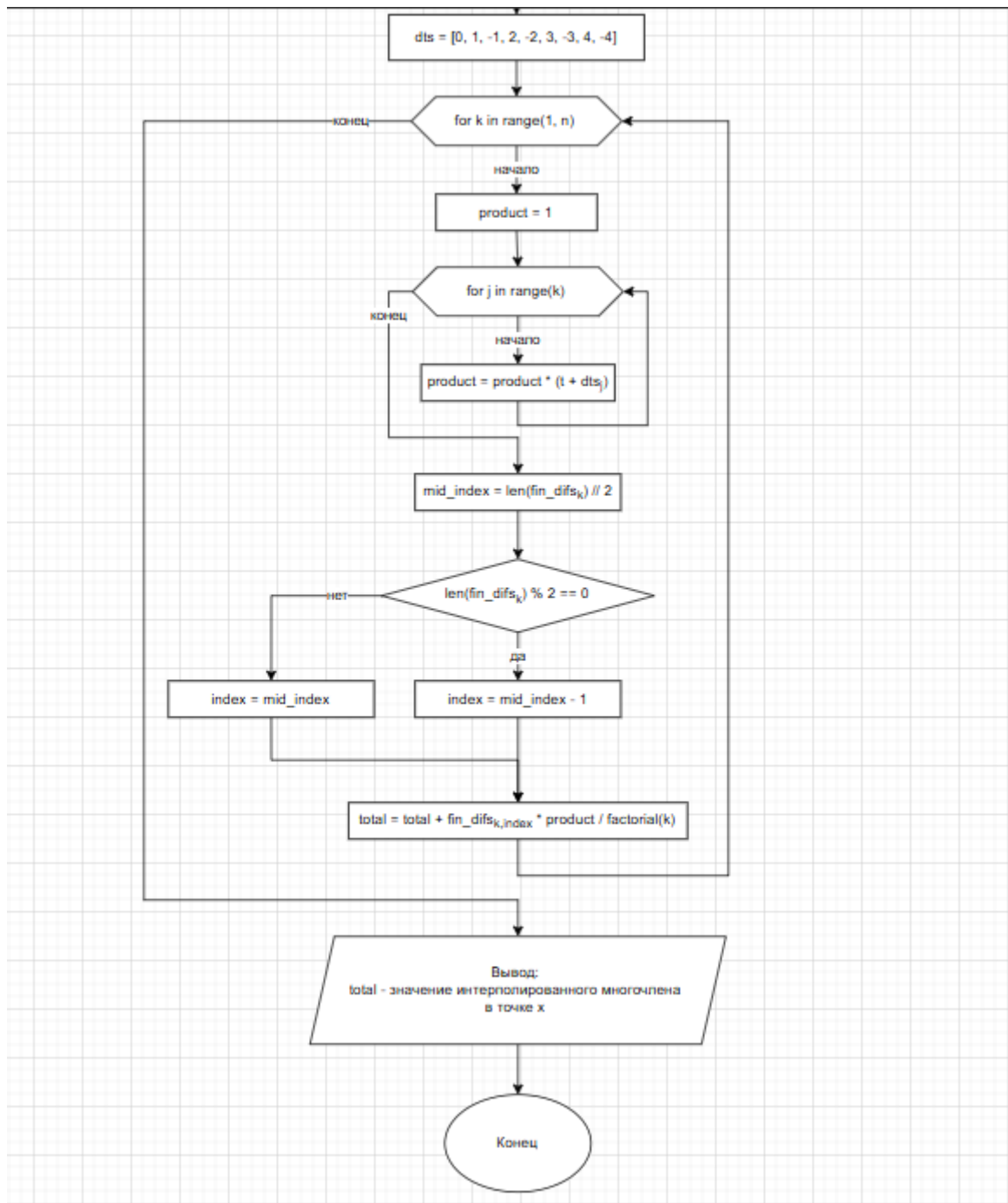


Многочлен Ньютона (конечные разности)



Многочлен Гаусса





Результаты выполнения программы при различных исходных данных:

[Загрузить из файла](#) [Ввод вручную](#) [Использовать пример](#) [Сгенерировать по функции](#)[Решить и Построить графики](#)

0.0001

Многочлен Лагранжа:

 $P(1.168) = 0.693365$

Многочлен Ньютона (конеч. разности):

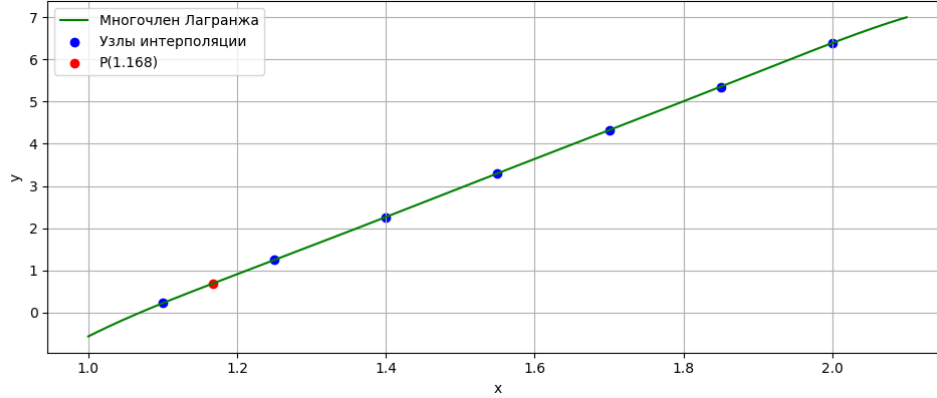
 $P(1.168) = 0.693365$

Многочлен Гаусса:

 $P(1.168) = 0.693365$ [← Предыдущий](#)[Следующий →](#)

График 1 из 3

Интерполяция методом: Многочлен Лагранжа

[Загрузить из файла](#) [Ввод вручную](#) [Использовать пример](#) [Сгенерировать по функции](#)[Решить и Построить графики](#)

Многочлен Лагранжа:

 $P(150.0) = 12.244495$

Многочлен Ньютона (раздел. разности):

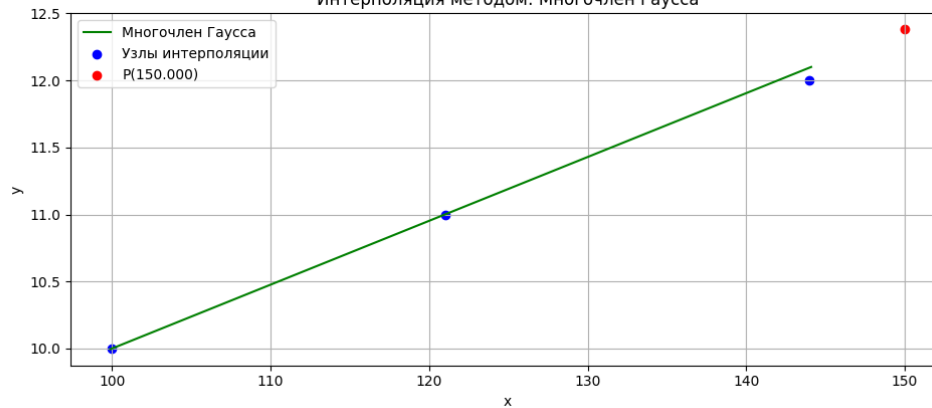
 $P(150.0) = 12.244495$

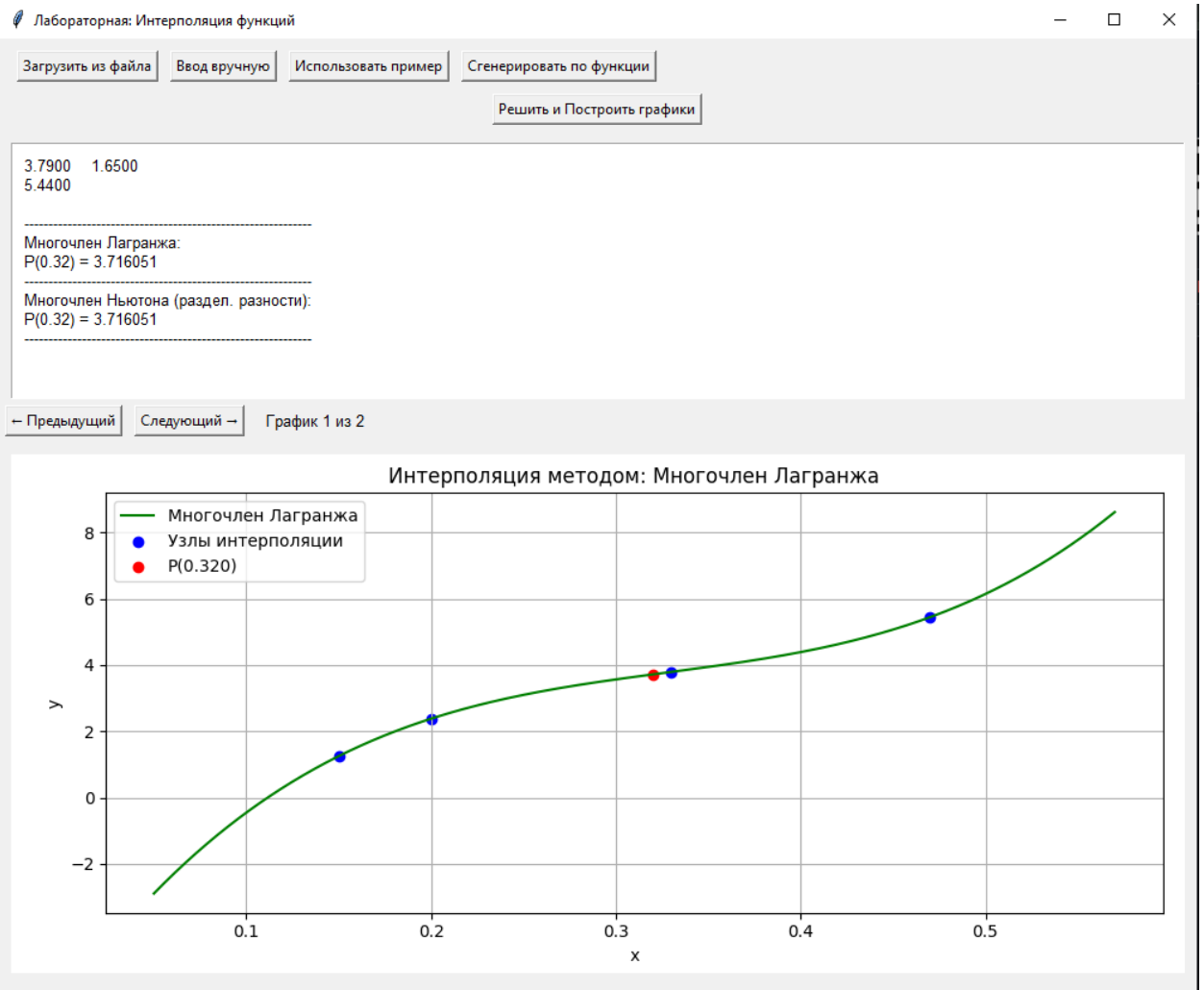
Многочлен Гаусса:

 $P(150.0) = 12.380952$ [← Предыдущий](#)[Следующий →](#)

График 3 из 3

Интерполяция методом: Многочлен Гаусса





Вывод

В ходе выполнения данной лабораторной работы я рассмотрел и реализовал методы интерполяции Ньютона и Гаусса для заданной таблицы данных. Интерполяция позволяет нам предсказывать значения функции в промежуточных точках на основе имеющихся данных.

С помощью разработанной программы были вычислены приближенные значения функции для заданных аргументов с использованием методов Ньютона и Гаусса. Было проведено сравнение результатов, полученных разными методами.

Результаты показали, что оба метода могут быть эффективно использованы для интерполяции, но их точность может зависеть от конкретной функции и распределения данных. Эта работа подчеркивает важность выбора подходящего метода интерполяции в соответствии с требованиями конкретной задачи.