

НИУ ИТМО
Факультет программной инженерии и компьютерных технологий

Отчет по лабораторной работе №5
по дисциплине Вычислительная математика

Студент группы № P32151
Преподаватель

Шипулин Павел Андреевич
Машина Екатерина Алексеевна

Санкт-Петербург

2023

Цель работы

Решить задачу интерполяции, найти значения функции при заданных значениях аргумента, отличных от узловых точек.

Вычислительная часть лабораторной работы

Исходные данные:

x	y
1,1	0,2234
1,25	1,2438
1,4	2,2644
1,55	3,2984
1,7	4,3222
1,85	5,3516
2	6,3867

$$X_1 = 1,168; X_2 = 1,463; h = 0,15.$$

Таблица конечных разностей:

n	x	$\Delta^0 y_i$	$\Delta^1 y_i$	$\Delta^2 y_i$	$\Delta^3 y_i$	$\Delta^4 y_i$	$\Delta^5 y_i$	$\Delta^6 y_i$
1	1,1	0,2234	1,0204	0,0002	0,0132	-0,0368	0,0762	-0,1313
2	1,25	1,2438	1,0206	0,0134	-0,0236	0,0394	-0,0551	
3	1,4	2,2644	1,0340	-0,0102	0,0158	-0,0157		
4	1,55	3,2984	1,0238	0,0056	0,0001			
5	1,7	4,3222	1,0294	0,0057				
6	1,85	5,3516	1,0351					
7	2	6,3867						

Для X_1 – интерполирование многочленом Ньютона (вперед).

$$x_1 \leq X_1 \leq x_2 \Rightarrow t = \frac{(X_1 - x_1)}{h} = 0,4533$$

$$t_1 = 1; t_2 = t; t_3 = t(t - 1)$$

$$t_{2m} = t(t - 1)(t + 1) \dots (t + m - 1)$$

$$t_{2m+1} = t(t - 1)(t + 1) \dots (t - m)$$

$$c_k = \Delta^{k-1} y_1 \frac{t_k}{(k-1)!}$$

$$N(X_1) = \sum c_k$$

Расчет t_k, c_k :

k	t_k	c_k
1	1	0,2234
2	0,4533	0,4626
3	-0,2478	0,0000
4	-0,3602	-0,0008
5	0,1969	-0,0003
6	0,2861	0,0002
7	-0,1564	0,0000

$$N(X_1) = 0,6851$$

Для X_2 – интерполирование многочленом Гаусса, вторая формула

$$x_3 \leq X_2 \leq x_4 \Rightarrow t = \frac{(X_2 - x_3)}{h} = -0,58$$

$$t_1 = 1; t_2 = t; t_3 = t(t + 1)$$

$$t_{2m} = t(t - 1)(t + 1) \dots (t - m)$$

$$t_{2m+1} = t(t - 1)(t + 1) \dots (t + m - 1)$$

$$c_k = \Delta^{k-1} y_{4-\lfloor \frac{k}{2} \rfloor} \frac{t_k}{(k-1)!}$$

$$G(X_2) = \sum c_k$$

Расчет t_k, c_k :

k	t_k	c_k
1	1	3,2984
2	-0,5800	-0,5938
3	-0,2436	-0,0007
4	0,3849	0,0000
5	0,1617	0,0000
6	-0,2554	0,0000
7	-0,1073	0,0000

$$G(X_2) = 2,7039$$

Код методов

```
import math

def mul(l=[]):
    result = 1.0
    for i in l:
        result = result * i
    return result

class AnyMethod:
    name = "some name"

    def __init__(self, x_train=[], y_train=[]):
        self.x_train = x_train
        self.y_train = y_train
```

```

        self.interpolating_func = lambda x: 0

    def interpolate(self):
        pass

    def calc(self, x):
        pass

    def __str__(self):
        return self.name

class Lagrange(AnyMethod):
    name = "Метод Лагранжа"

    def __init__(self, x_train=[], y_train=[]):
        super().__init__(x_train, y_train)
        self.lagrange_elements = []
        self.interpolate()

    def interpolate(self):
        if len(self.x_train) != len(self.y_train):
            raise Exception("Размеры данных не совпадают")

        n = len(self.x_train)
        self.lagrange_elements = []

        for j in range(n):
            self.lagrange_elements.append(lambda x, i: mul([(x - x_j) /
(self.x_train[i] - x_j)
                                                                    for x_j in
(set(self.x_train) - {self.x_train[i]})]))

            self.interpolating_func = lambda x:
sum([(self.lagrange_elements[j](x, j) * self.y_train[j]) for j in range(n)])

    def calc(self, x):
        return self.interpolating_func(x)

class Gauss(AnyMethod):
    name = "Метод Гаусса"

    def __init__(self, x_train=[], y_train=[]):
        super().__init__(x_train, y_train)

        self.index_center = 0
        self.x_center = 0
        self.h = 1
        self.delta_y = []
        self.first_interpolating_func = lambda t: 0
        self.second_interpolating_func = lambda t: 0

        self.interpolate()

    def interpolate(self):
        if len(self.x_train) != len(self.y_train):
            raise Exception("Размеры данных не совпадают")

        n = len(self.x_train)

        delta_x = [self.x_train[i + 1] - self.x_train[i] for i in range(n -
1)]

        delta_x_mean = sum(delta_x) / (n - 1)

```

```

        for i in range(n - 1):
            if not (0.5 <= delta_x[i] / delta_x_mean <= 2):
                raise Exception("Промежутки между точками должны быть
одинаковыми")

        self.index_center = n // 2
        self.x_center = self.x_train[self.index_center]
        self.h = delta_x_mean

        self.delta_y = [self.y_train.copy()]
        for i in range(1, n):
            self.delta_y.append([])
            for j in range(n - i):
                self.delta_y[i].append(self.delta_y[i - 1][j + 1] -
self.delta_y[i - 1][j])

        self.first_interpolating_func = lambda t:
self.delta_y[0][self.index_center] + sum([
            self.delta_y[i][self.index_center - i // 2] * mul([
                t + ((-1) ** j) * (j + 1) // 2
                for j in range(i)
            ]) / math.factorial(i)
            for i in range(1, len(self.x_train) - 1)
        ])

        self.second_interpolating_func = lambda t:
self.delta_y[0][self.index_center] + sum([
            self.delta_y[i][self.index_center - (i + 1) // 2] * mul([
                t - ((-1) ** j) * (j + 1) // 2
                for j in range(i)
            ]) / math.factorial(i)
            for i in range(1, len(self.x_train) - 1)
        ])

    def calc_first(self, x):
        return self.first_interpolating_func((x - self.x_center) / self.h)

    def calc_second(self, x):
        return self.second_interpolating_func((x - self.x_center) / self.h)

    def calc(self, x):
        if x > self.x_center:
            return self.calc_first(x)
        return self.calc_second(x)

```

Примеры выполнения

Пример 1.

[Info]: Введите команду:

lab5

[Input]: lab5

[Info]: Выберите способ ввода:

- 1) из файла / с клавиатуры
- 2) функция

1

[Input]: 1

[Info]: Введите путь файла или пустую строку (ввод с клавиатуры)

input.txt

[Input]: input.txt

[Info]: Ввод из файла

[Info]: Введите количество точек

[Input]: 8

[Info]: Введите x_0

[Input]: -3

[Info]: Введите y_0

[Input]: -3

[Info]: Введите x_1

[Input]: -2

[Info]: Введите y_1

[Input]: -2

[Info]: Введите x_2

[Input]: -1

[Info]: Введите y_2

[Input]: -1

[Info]: Введите x_3

[Input]: 1

[Info]: Введите y_3

[Input]: 0

[Info]: Введите x_4

[Input]: 1

[Info]: Введите y_4

[Input]: 1

[Info]: Введите x_5

[Input]: 2

[Info]: Введите y_5

[Input]: 2

[Info]: Введите x_6

[Input]: 3

[Info]: Введите y_6

[Input]: 3

[Info]: Введите x_7

[Input]: 4

[Info]: Введите y_7

[Input]: 4

[Info]: Введите x

[Info]: Ввод из файла `F:\Programming\python\CalcMath\Labs\Lab5\input.txt` закончен
1.5

[Input]: 1.5

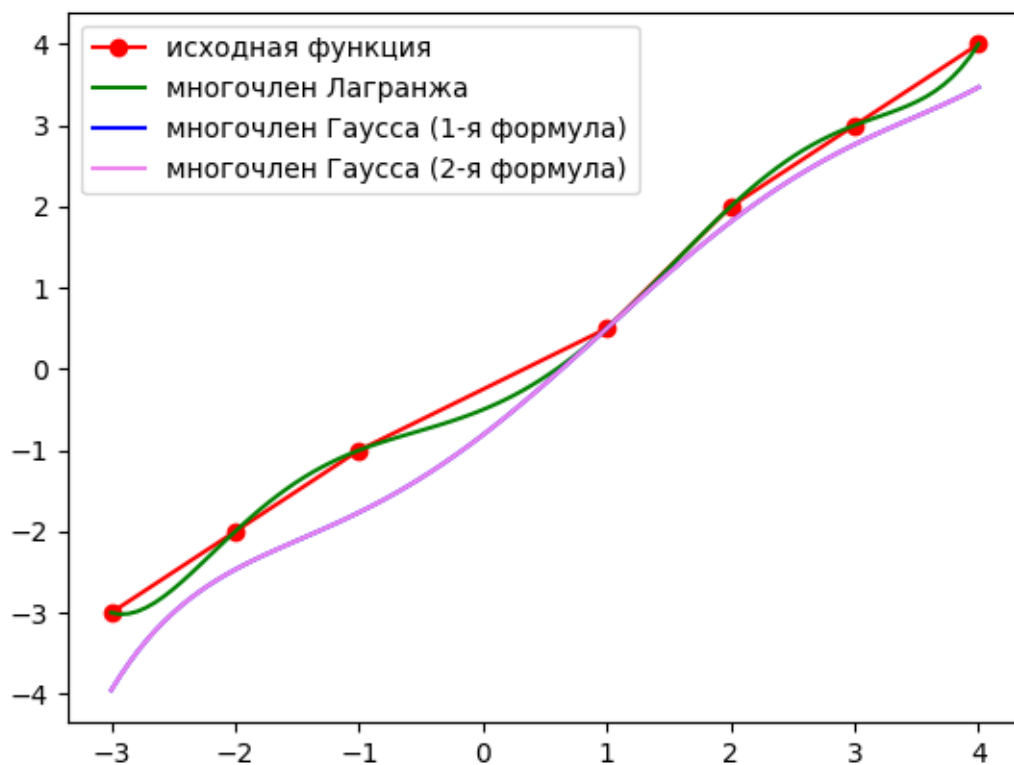
[Info]: Многочлен Лагранжа: $y(1.5) = 1.24365234375$

[Info]: Многочлен Гаусса (1-я формула): $y(1.5) = 1.183137978223359$

[Info]: Многочлен Гаусса (2-я формула): $y(1.5) = 1.183137978223359$

[Info]: Таблица конечных разностей:

i	y _i	d1 y _i	d2 y _i	d3 y _i	d4 y _i	d5 y _i	d6 y _i
-3	-3.00000	1.00000	0.00000	0.50000	-1.00000	1.00000	0.00000
-2	-2.00000	1.00000	0.50000	-0.50000	0.00000	1.00000	-
-1	-1.00000	1.50000	0.00000	-0.50000	1.00000	-	-
0	0.50000	1.50000	-0.50000	0.50000	-	-	-
1	2.00000	1.00000	0.00000	-	-	-	-
2	3.00000	1.00000	-	-	-	-	-
3	4.00000	-	-	-	-	-	-



[Info]: Лабораторная работа 5 (интерполяция) завершилась

Пример 2.

[Info]: Введите команду:

lab5

[Input]: lab5

[Info]: Выберите способ ввода:

1) из файла / с клавиатуры

2) функция

2

[Input]: 2

[Info]:

```

+-----+-----+
| Номер |          Уравнение |
+-----+-----+
|      1 |      x * sin(x ** 2) |
+-----+-----+
|      2 | e ** (-x ** 2) * sin(x) |

```

```

+-----+-----+
|      3 |           x * cos(x) |
+-----+-----+

```

[Info]: Введите номер уравнения

3

[Input]: 3

[Info]: Введите количество точек

5

[Input]: 5

[Info]: первая граница интервала:

-1

[Input]: -1

[Info]: вторая граница интервала

3

[Input]: 3

[Info]: Введите x

2

[Input]: 2

[Info]: Многочлен Лагранжа: $y(2.0) = -0.8322936730942848$

[Info]: Многочлен Гаусса (1-я формула): $y(2.0) = -0.8322936730942847$

[Info]: Многочлен Гаусса (2-я формула): $y(2.0) = -0.8322936730942847$

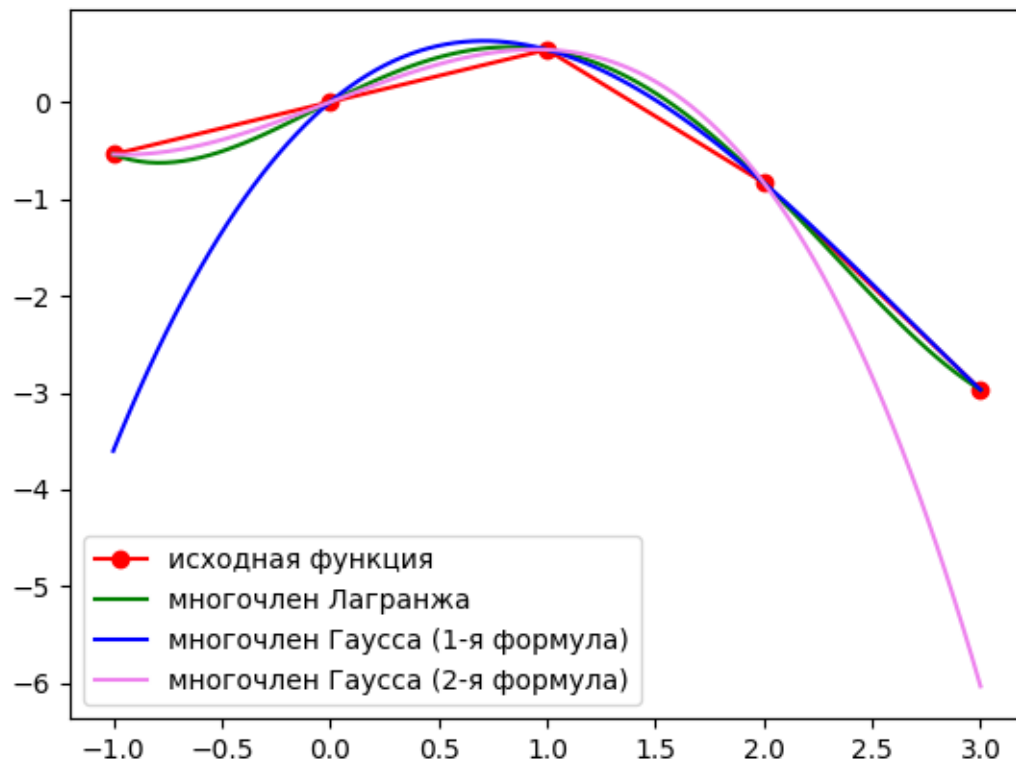
[Info]: Таблица конечных разностей:

```

+----+-----+-----+-----+-----+-----+
|  i  |      y_i |   d1 y_i |   d2 y_i |   d3 y_i |   d4 y_i |
+----+-----+-----+-----+-----+-----+
| -2  | -0.54030 |  0.54030 |  0.00000 | -1.91290 |  3.06071 |
+----+-----+-----+-----+-----+-----+
| -1  |  0.00000 |  0.54030 | -1.91290 |  1.14781 |         - |
+----+-----+-----+-----+-----+-----+
|  0  |  0.54030 | -1.37260 | -0.76509 |         - |         - |
+----+-----+-----+-----+-----+-----+
|  1  | -0.83229 | -2.13768 |         - |         - |         - |
+----+-----+-----+-----+-----+-----+

```

| 2 | -2.96998 | - | - | - | - |
+---+-----+-----+-----+-----+-----+-----+



[Info]: Лабораторная работа 5 (интерполяция) завершилась

Выводы

Научился интерполировать функции, в частности, многочленами: Лагранжа, Ньютона (вперед), Гаусса (1-я и 2-я формулы).