Университет ИТМО
Факультет программной инженерии и компьютерной техники

Лабораторная работа
по вычислительной математике №5

Вариант: 23
Преподаватель: Малышева Татьяна Алексеевна
Выполнил: Щербаков Александр Валерьевич
Группа: P3210

Санкт-Петербург
2022

Цель работы:

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

Порядок выполнения работы:

Вычислительная реализация задачи:

1) Вычислить значения функции при данных значениях аргумента
2) Построить таблицу конечных разностей.

Программная реализация задачи:

1) Исходные данные задаются в виде: а) набора данных (таблицы x,y), б) на основе выбранной функции (например, $sinx$).
2) Вычислить приближенное значение функции для заданного значения аргумента, введенного с клавиатуры, указанными методами (см. табл.5).
3) Построить графики заданной функции с отмеченными узлами интерполяции и интерполяционного многочлена Ньютона/Гаусса (разными цветами).

Рабочие формулы:

Интерполяционная формула Лагранжа:

$$L_n(x) = \sum_{i=0}^{n} y_i \prod_{\substack{j=0 \\ j \neq i}}^{n} \frac{(x - x_j)}{(x_i - x_j)}$$

, где n - количество узлов интерполяции

Формула Ньютона:
Чтобы вычислить интрополяционное значение функции в точке, если интервалы между точками не одинаковые (не равны константе)

$$N_n(x) = f(x_0) + \sum_{k=1}^{n} f(x_0, x_1, ...x_k) \prod_{j=0}^{k-1} (x - x_j)$$

Формула Ньютона: Чтобы вычислить интрополяционное значение функции в точке, если интервалы между точками одинаковые

Интрополяция вперёд:

$$N_n(x) = y_n + t\Delta y_{n-1} + \frac{t(t-1)}{2!}\Delta^2 y_{n-2} + \cdots + \frac{t(t+1)(t-n+1)}{n!}\Delta^n y_0, t = \frac{(x - x_0)}{h}$$

Интрополяция назад:

$$N_n(x) = y_n + t\Delta y_{n-1} + \frac{t(t+1)}{2!}\Delta^2 y_{n-2} + \cdots + \frac{t(t+1)(t+n-1)}{n!}\Delta^n y_0, t = \frac{(x - x_n)}{h}$$

Вычисление значений функции:

Исходные данные:

X	1.10	1.25	1.40	1.55	1.70	1.85	2.00
Y	0.2234	1.2438	2.2644	3.2984	4.3222	5.3516	6.3867

$x_1 = 1.891 \qquad x_2 = 1.671$

Вычислим конечные разности по рекуррентной формуле:

$$f(x_0, x_1) = \frac{f(x_1) - f(x_0)}{x_1 - x_0}, f(x_i, x_{i+1}) = \frac{f(x_{i+1}) - f(x_i)}{x_{i+1} - x_i}, f(x_i, x_{i+1}) = \frac{f(x_{i+1}) - f(x_i)}{x_{i+1} - x_i}$$

$$f(x_i, x_{i+1}, ..., x_{i+k}) = \frac{f(x_{i+1}, ..., x_{i+k}) - f(x_i, ..., x_{k-1})}{x_{i+1} - x_i}$$

Занесем конечные разности в таблицу:

	y_0	Δ	Δ^2	Δ^3	Δ^4	Δ^5	Δ^6
x_0	0.2234	1.0204	0.0002	0.0132	-2.0368	10.0762	-30.1313
x_1	1.2438	1.0206	0.0134	-2.0236	8.0394	-20.0551	
x_2	2.2644	1.034	-2.0102	6.0158	-12.0157		
x_3	3.2984	-0.9762	4.0056	-6			
x_4	2.3222	3.0294	-1.9943				
x_5	5.3516	1.0351					
x_6	6.3867						

Опираясь на таблицу “строим” многочлен ньютона по формуле:

$$N_n(x) = y_n + t\Delta y_{n-1} + \frac{t(t+1)}{2!}\Delta^2 y_{n-2} + \cdots + \frac{t(t+1)(t+n-1)}{n!}\Delta^n y_0, t = \frac{(x - x_n)}{h}$$

Упростив выражение получаем:

$$-16.0097x^6 + 150.0485x^5 - 580.8802x^4 + 1188.5865x^3 - 1355.3623x^2 + 823.2480x - 210.1952$$

подставляем туда значения x_1 и x_2 получаем 5.6367 и 4.1248 соответственно.

Листинг вычислительных методов программы:

Solver.scala

```
package main.Methods

import scala.collection.mutable

object Solver {

  private def crate_basic_polynomial(x_values: mutable.Buffer[Double], i: Int): Double =>
Double = {

    def basic_polynomial(x: Double): Double = {

      var divider: Double = 1.0

      var result: Double = 1.0

      for(j <- x_values.indices){

        if(j != i){

          result *= (x-x_values(j))

          divider *= (x_values(i) - x_values(j))

        }

      }

      result/divider

    }

    basic_polynomial

  }

  def create_Lagrange_polynomial(x_values: mutable.Buffer[Double], y_values:
mutable.Buffer[Double]): Double => Double = {

    val basic_polynomials = mutable.Buffer[Double => Double]()

    for(i <- 0 to x_values.size){

      basic_polynomials += crate_basic_polynomial(x_values, i)

    }

    def lagrange_polynomial(x: Double):Double ={

      var result: Double = 0.0

      for (i <- y_values.indices){

        result += y_values(i) * basic_polynomials(i).apply(x)
```

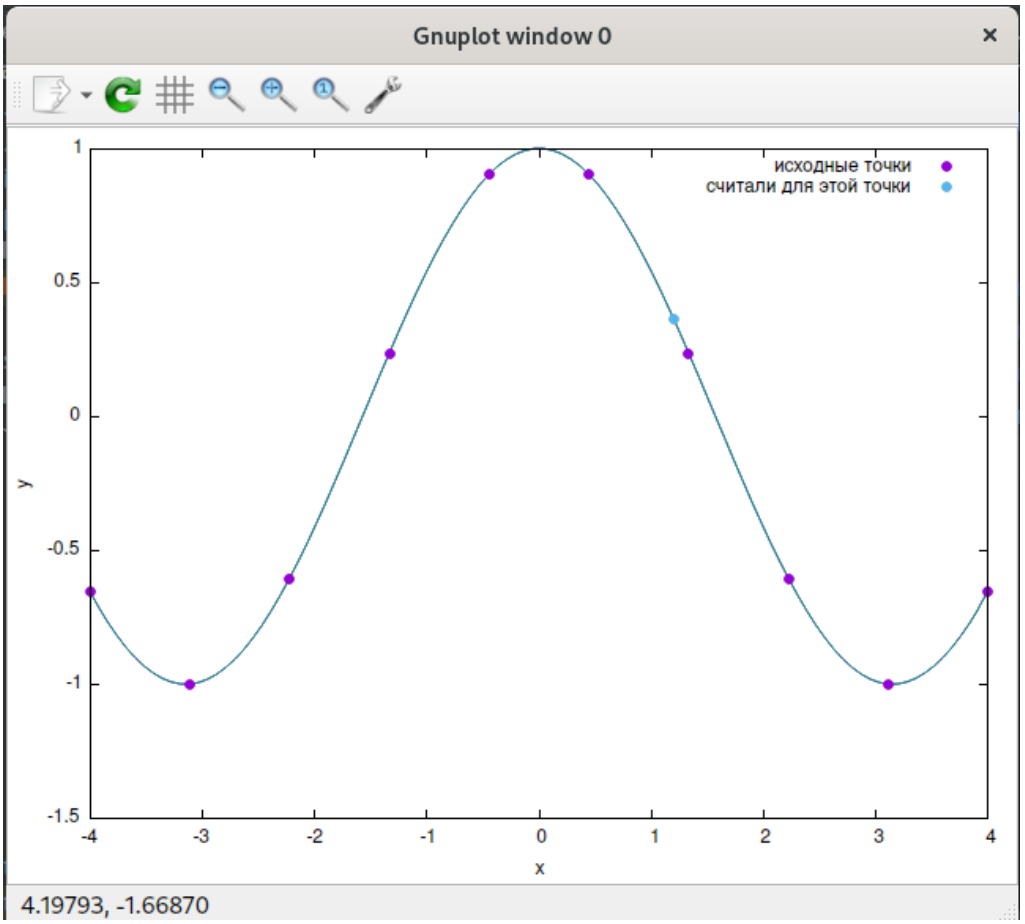
```
      }
      result
    }
    lagrange_polynomial
  }
  private def divided_differences(x_values: mutable.Buffer[Double], y_values:
mutable.Buffer[Double], k: Int): Double = {
    var result: Double = 0.0
    for(j <- 0 to k){
      var mul: Double = 1.0
      for (i <-0 to k) {
        if (i != j) {
          mul *= (x_values(j) - x_values(i))
        }
      }
      result += y_values(j) / mul
    }
    result
  }
  def create_Newton_polynomial(x_values: mutable.Buffer[Double], y_values:
mutable.Buffer[Double]): Double => Double = {
    val div_diff = mutable.Buffer[Double]()
    for(i <- 1 until x_values.size){
      div_diff += divided_differences(x_values, y_values, i)
    }
    def newton_polynomial(x: Double): Double = {
      var result: Double = y_values.head
      for(k <- 1 until y_values.size){
        var mul: Double = 1.0
        for(j <- 0 until k) {
```

```
        mul *= (x-x_values(j))
      }
      result += div_diff(k-1) * mul
    }
    result
  }
  newton_polynomial
 }
 def create_finite(y_values: mutable.Buffer[Double], k: Int): Double ={
  y_values(k)-y_values(k-1)
 }
 var defs = mutable.Buffer[mutable.Buffer[Double]]()
 def create_Newton_polynomial_finite(x_values: mutable.Buffer[Double], y_values:
mutable.Buffer[Double]): Double => Double = {
  defs += y_values
  for(i <- 1 until y_values.size){
   defs += mutable.Buffer[Double]()
   for(j <- 1 until defs(i-1).size - 1){
    defs.last += create_finite(defs(defs.size - 2), j)
   }
  }
  x => x
 }
}
```

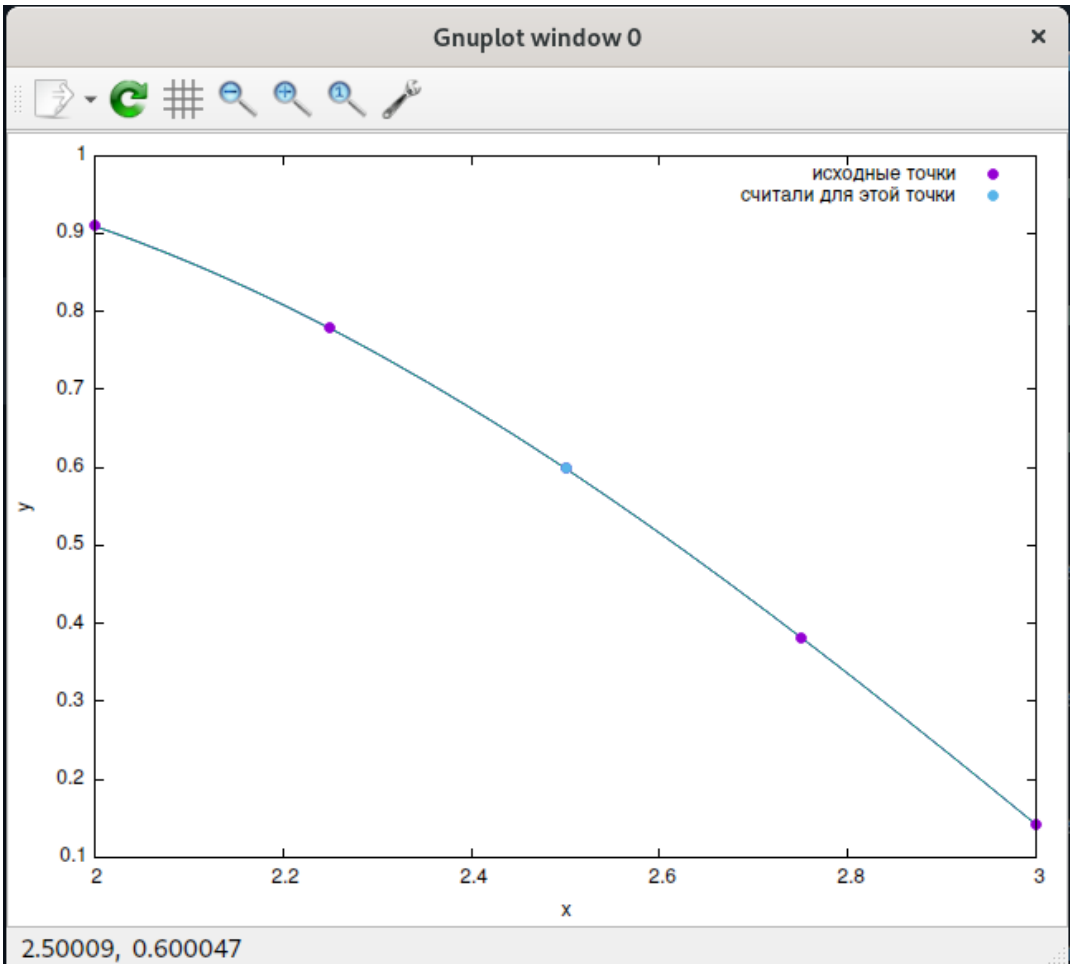
Примеры (результаты выполнения программы):

Пример 1:

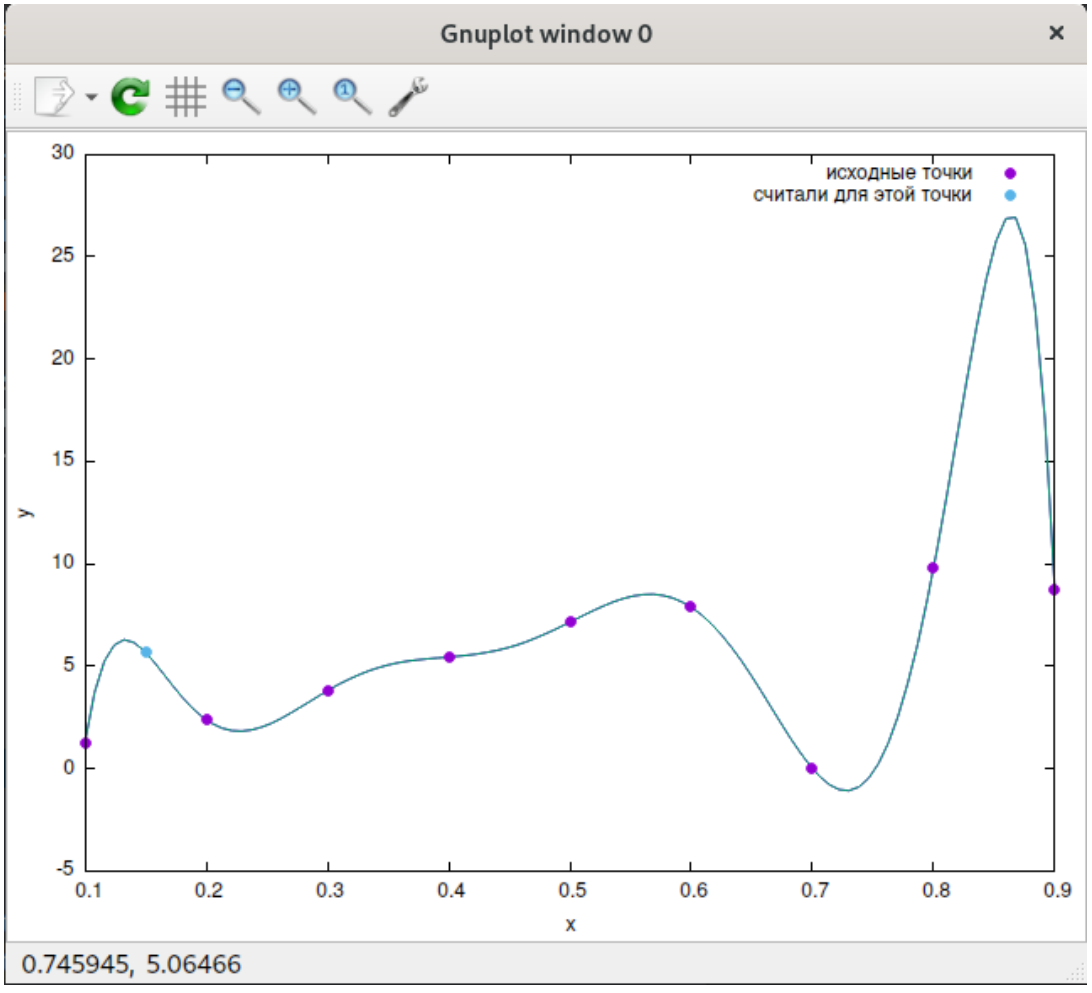
Привет, это пятая лаба по вычмату!
Сейчас будем интерполировать :)
Где будем брать точки?
1: В файле input/input1.csv
2: Зададим функцией
2
которой? (есть только заданные, парсер прикручивать лень)
1: sin(x)
2: cos(x)
3: x^10
2
Введите левый край промежутка в виде десятичной дроби
-4
Введите правый край промежутка в виде десятичной дроби
4
Сколько точек вы желаете видеть на промежутке?
9
Для какой точки будем считать значение?
1.2
Результат для выбранной точки: 0.362395211045334 - через полином Лагранджа.
0.36239521104533545 - через полином Ньютона



Пример 2:
Привет, это пятая лаба по вычмату!
Сейчас будем интерполировать :)
Где будем брать точки?
1: В файле input/input1.csv
2: Зададим функцией
2
которой? (есть только заданные, парсер прикручивать лень)
1: sin(x)
2: cos(x)
3: x^10
1
Введите левый край промежутка в виде десятичной дроби
2
Введите правый край промежутка в виде десятичной дроби
3
Сколько точек вы желаете видеть на промежутке?
4
Для какой точки будем считать значение?
2.5
Результат для выбранной точки: 0.5984721441039564 - через полином Лагранджа.
0.5984721441039564 - через полином Ньютона



Пример 3:
Привет, это пятая лаба по вычмату!
Сейчас будем интерполировать :)
Где будем брать точки?
1: В файле input/input1.csv
2: Зададим функцией
1
x: ArrayBuffer(0.1, 0.2, 0.3, 0.4, 0.5, 0.6, 0.7, 0.8, 0.9)
y: ArrayBuffer(1.25, 2.38, 3.79, 5.44, 7.14, 7.88, 0.0, 9.8, 8.7)
Для какой точки будем считать значение?
0.15
Результат для выбранной точки: 5.680613098144526 - через полином Лагранджа.
5.6806130981445255 - через полином Ньютона



Вывод:

В ходе выполнения лабораторной работы были получены знания о интерполяции по заданным точка, а также некоторые навыки программной реализации интерполяционных формул и построения графиков.