|  |  |
| --- | --- |
| Gerb-BMSTU_01 | **Министерство науки и высшего образования Российской Федерации**  **Федеральное государственное бюджетное образовательное учреждение**  **высшего образования**  **«Московский государственный технический университет**  **имени Н.Э. Баумана**  **(национальный исследовательский университет)»**  **(МГТУ им. Н.Э. Баумана)** |

ФАКУЛЬТЕТ **Информатика и системы управления**

КАФЕДРА **Программное обеспечение ЭВМ и информационные технологии**

**Лабораторная работа №2.**

**«Построение и программная реализация алгоритма многомерной интерполяции табличных функций»**

Студент **Леонов Владислав Вячеславович**

Группа **ИУ7-46Б**

Студент **\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_** Леонов В.В.

*подпись, дата фамилия, и.о.*

Преподаватель \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ Градов В.М.

*подпись, дата фамилия, и.о.*

Оценка \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

*Москва, 2021 г.*

**Оглавление**

[Цель работы 3](#_Toc64927853)

[Исходные данные 3](#_Toc64927854)

[Описание алгоритма 3](#_Toc64927855)

[Код программы 4](#_Toc64927856)

[Результаты работы 6](#_Toc64927857)

[Ответы на контрольные вопросы 7](#_Toc64927858)

# Цель работы

Получение навыков построения алгоритма интерполяции таблично заданных функций двух переменных.

# Исходные данные

1. Таблица функции с количеством узлов 5x5.

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
|  | 0 | 1 | 2 | 3 | 4 |
| 0 | 0 | 1 | 4 | 9 | 16 |
| 1 | 1 | 2 | 5 | 10 | 17 |
| 2 | 4 | 5 | 8 | 13 | 20 |
| 3 | 9 | 10 | 13 | 18 | 24 |
| 4 | 16 | 17 | 20 | 25 | 32 |

2. Степень аппроксимирующих полиномов – и .

3. Значение аргументов для которых выполняется интерполяция.

# Описание алгоритма

Для решения поставленной задачи используется построение **полинома Ньютона**. Удобно ввести понятие разделенной разности:

*…*

Правило образования таких конструкций понятно из приведенной выше записи. Тогда в результате получаем так называемый полином Ньютона, который имеет следующий вид:

Для заданной табличной функции выполняется сначала интерполяция по *x.* При этом выполняется одномерных интерполяций для выбранных значений , и вычисляются значения функции А затем по полученным значениям функции, привязанным теперь к , совершается одна интерполяция по *.*

# Код программы

Код программы представлен ниже.

|  |
| --- |
| Файл ***main.py*** |
| **def** data\_read(filename):  dots = []  f = open(filename, "r")  values = []  x\_dots = f.readline()  x\_dots = x\_dots.replace("**\n**", "").split(" ")  x\_dots = [float(dot) **for** dot **in** x\_dots]  dots.append(x\_dots)  y\_dots = f.readline()  y\_dots = y\_dots.replace("**\n**", "").split(" ")  y\_dots = [float(dot) **for** dot **in** y\_dots]  dots.append(x\_dots)  **for** i **in** range(len(dots[1])):  line = f.readline()  line = line.replace("**\n**", "").split(" ")  line = [float(dot) **for** dot **in** line]  values.append(line)  **return** dots, values      **def** data\_print(dots, values):  **for** i **in** range(len(dots)):  **if** (i == 0):  **print**("X:")  **else**:  **print**("**\n**Y:")  **for** dot **in** dots[i]:  **print**(dot, end=" ")  **print**("**\n**Values:")  **for** row **in** values:  **for** dot **in** row:  **print**(dot, end=" ")  **print**()      **def** find\_dots(data, polinom\_degree, func\_arg):  **for** i **in** range(len(data)-1):  **if** (data[i] < func\_arg **and** data[i+1] > func\_arg):  index = i  **break**  curr\_dots = 0  step = 0  **while** curr\_dots <= polinom\_degree:  **if** (index - step >= 0):  left = index - step  curr\_dots += 1  **if** (index + 1 + step < len(data) **and** curr\_dots <= polinom\_degree):  right = index + 1 + step  curr\_dots += 1  step += 1  **return** left, right      **def** reduce\_data(dots, values, polinom\_degree\_x, polinom\_degree\_y, x\_arg, y\_arg):  x\_left, x\_right = find\_dots(dots[0], polinom\_degree\_x, x\_arg)  y\_left, y\_right = find\_dots(dots[1], polinom\_degree\_y, y\_arg)  dots[0] = dots[0][x\_left:x\_right+1:1]  dots[1] = dots[1][y\_left:y\_right+1:1]  tmp = []  **for** i **in** range(y\_left, y\_right+1, 1):  tmp.append(values[i][x\_left:x\_right+1:1])  values = tmp  **return** dots, values      **def** newtone\_func\_value(data, func\_arg):  coefficients = []  curr\_col = []  curr\_col\_i = 1  **for** row **in** data:  curr\_col.append(row[1])  coefficients.append(curr\_col[0])  new\_col = []  **while** (len(curr\_col) > 1):  **for** i **in** range(len(curr\_col)-1):  new\_col.append((curr\_col[i]-curr\_col[i+1]) /  (data[i][0]-data[i+curr\_col\_i][0]))  **if** (i == 0):  coefficients.append(new\_col[0])  curr\_col\_i += 1  curr\_col = new\_col.copy()  new\_col.clear()  func\_value = coefficients[0]  curr\_x = 1  **for** i **in** range(len(data)-1):  curr\_x \*= (func\_arg - data[i][0])  func\_value += coefficients[i+1]\*curr\_x  **return** func\_value      **def** func\_value(dots, values, polinom\_degree\_x, polinom\_degree\_y, x\_arg, y\_arg):  dots, values = reduce\_data(  dots, values, polinom\_degree\_x, polinom\_degree\_y, x\_arg, y\_arg)  interpolated\_x = []  **for** i **in** range(polinom\_degree\_y+1):  tmp\_data = [(dots[0][j], values[i][j])  **for** j **in** range(polinom\_degree\_x+1)]  interpolated\_x.append(newtone\_func\_value(tmp\_data, x\_arg))  tmp\_data = [(dots[1][j], interpolated\_x[j])  **for** j **in** range(polinom\_degree\_y+1)]  **return** newtone\_func\_value(tmp\_data, y\_arg)      **def** main():  filename = input("Data file:")  dots, values = data\_read(filename)  data\_print(dots, values)  polinom\_degree\_x = int(input("Polinom degree X:"))  polinom\_degree\_y = int(input("Polinom degree Y:"))  x\_arg = float(input("X value:"))  y\_arg = float(input("Y value:"))  func = func\_value(dots, values, polinom\_degree\_x,  polinom\_degree\_y, x\_arg, y\_arg)  **print**("Function value = ", func)      **if** (\_\_name\_\_ == "\_\_main\_\_"):  main() |

# Результаты работы

1. Результат интерполяции при степенях полиномов для .

|  |  |  |  |
| --- | --- | --- | --- |
|  | 1 | 2 | 3 |
| 1 | 5.0 | 4.75 | 4.75 |
| 2 | 4.75 | 4.5 | 4.5 |
| 3 | 4.75 | 4.5 | 4.5 |

# Ответы на контрольные вопросы

**1. Пусть производящая функция таблицы суть . Область определения по x и y 0-5 и 0-5. Шаги по переменным равны 1. Степени , . Приведите по шагам те. значения функции, которые получаются в ходе последовательных интерполяций. по строкам и столбцу.**

*Выбор узлов:*

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
|  | 0 | 1 | 2 | 3 | 4 | 5 |
| 0 | 0 | 1 | 4 | 9 | 16 | 25 |
| 1 | 1 | 2 | 5 | 10 | 17 | 26 |
| 2 | 4 | 5 | 8 | 13 | 20 | 29 |
| 3 | 9 | 10 | 13 | 18 | 24 | 34 |
| 4 | 16 | 17 | 20 | 25 | 32 | 41 |
| 5 | 25 | 26 | 29 | 34 | 41 | 50 |

*Интерполяция по :*

1 строка:

|  |  |  |
| --- | --- | --- |
|  |  |  |
|  |  |  |
|  |  |

2 строка:

|  |  |  |
| --- | --- | --- |
|  |  |  |
|  |  |  |
|  |  |

*Интерполяция по :*

|  |  |  |
| --- | --- | --- |
|  |  |  |
|  |  |  |
|  |  |

**2. Какова минимальная степень двумерного полинома, построенного на 4 узлах? На шести узлах?**

В случае, если у нас есть 4 узла, можно построить двумерный полином 2 степени, если же узлов 6, то можно построить также полином 2 степени, однозначно определив соответствующие коэффициенты, решая систему уравнений.

**3. Предложите алгоритм двумерной интерполяции при хаотичном расположении узлов, т.е. когда таблицы функции на регулярной сетке нет, и метод последовательной интерполяции не работает. Какие имеются ограничения на расположение узлов при разных степенях полинома?**

В случае неравномерной сетки возможно воспользоваться следующим методом: пусть нужно построить полином первой степени, тогда имеем , и его коэффициенты находят по трем узлам, выбираемым в окрестности точки интерполяции:

номер узла.

При использовании полинома второй степени:

Далее аналогично. Необходимо помнить о следующих ограничениях – при интерполяции полиномом узлы интерполяции не должны лежать на одной кривой, задаваемой полином степени .

**4. Пусть на каком-либо языке программирования написана функция, выполняющая интерполяцию по двум переменным. Опишите алгоритм использования этой функции для интерполяции по трем переменным.**

Пусть задана функция для интерполяции и функция, которая соответственно производит интерполяцию. Изначально необходимо произвести интерполяцию по любым двум переменным, например, и – . После чего необходимо произвести интерполяцию по полученным в предыдущем шаге значениям и по оставшейся переменной, в данном конкретном случае . Таким образом, можно про интерполировать функцию 3 переменных, производя последовательную интерполяцию по 2 переменным.

**5. Можно ли при последовательной интерполяции по разным направлениям использовать полиномы несовпадающих степеней или даже разные методы одномерной интерполяции, например, полином Ньютона или сплайн?**

Да, можно, однако использование тех или иных методов, степеней и порядка их применения будет существенно влиять на точность расчетов, также необходимо помнить об определённых ограничениях при интерполяции.

**6. Опишите алгоритм двумерной интерполяции на треугольной конфигурации узлов.**

На треугольной конфигурации узлов для каждого узла, находящегося на границе множества узлов, вычисляем углы наклона плоскостей всех треугольников, в которые входит узел, и берем среднее значение. Тогда для треугольников не на краю получаем 6 параметров и можно построить полином 2 степени.