



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

Тема: Построение и программная реализация алгоритма многомерной интерполяции табличных функций.

Студент: Чаушев Александр

Группа: ИУ7-46Б

Оценка (баллы) _____

Преподаватель : Градов В. М.

Москва.
2020 г.

Цель работы: Получение навыков построения алгоритма интерполяции таблично заданных функций двух переменных.

Входные данные: таблица значений функции двух переменных, аргументы, степень полинома Ньютона для каждой переменной.

Результат работы программы: Значение функции от двух переменных $f(x,y)$.

Алгоритм:

С помощью функции `func_matrix` создаем матрицу за значениями функции от двух аргументов. производим выборку узлов по x и y (функция `find_nodes`). Для каждого фиксированного значения первого аргумента из таблицы выполнить одномерную интерполяцию по второму аргументу. Из полученных значений функции и значений первого аргумента сформировать таблицу. Выполнить двумерную интерполяцию по полученной таблице.

Пример:

$X \backslash Y$... 2	5 ...		2 5		2 3		3 z1		
					= z1		= z2		= z(x,y)	
.				5 6		5 10		4 z2		
.										
3		5 6								
4		3 10								
.										
.										

Контрольные вопросы:

Пусть производящая функция таблицы суть $z(x,y)=x^2+y^2$. Область определения по x и y 0-5 и 0-5. Шаги по переменным равны 1. Степени $p_x = p_y = 1$, $x=y=1.5$. Приведите по шагам те. значения функции, которые получаются в ходе последовательных интерполяций. по строкам и столбцу.

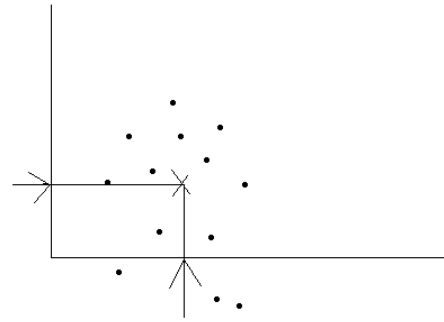
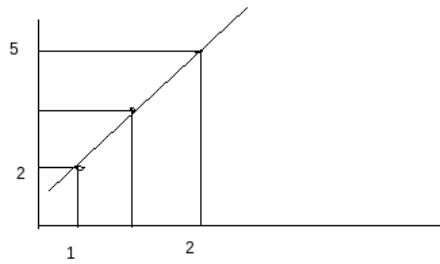
- При степен полиномов 1 и $x_0 = 1.5$ выбирает две строки $x = 1$, $x = 2$. При интерполяция в этих строках в $y_0 = 1.5$, тоже будет выбрано по две точки $y = 1$, $y = 2$. В строках функция аппроксимируется прямой, а 1 и 2 равноудалены от x_0 , следовательно полученное значение будет равно среднему арифметическое значение в узлах - 3.5 для первой строки и 6.5 для второй. Полученное значение снова аппроксимируются по прямой внутри столбца. Аналогично получаем среднее арифметическое для $z(1.5,1.5) = 5$

Какова степень двумерного полинома, построенного на четырех узлах? На шести узлах?

- Второй степени на четырех узлах. Четвертая степень на шести узлах

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

- 1. Выбираем третью переменную. 2. Строим выборку (нужные третьи переменные) 3. Вызываем интерполяцию для двух переменных в цикле с каждым выбранным значением. 4. Потом по полученным точкам нужно провести одномерную интерполяцию



$U=f(x,y)$
 $U=f(x,y,z)$
 $U_0=f(x,y,z_0)$
 $U_1=f(x,y_1,z_1)$
 \cdot
 \cdot
 \cdot
 $U_{nz+1}=f(x,y,z_{n+1})$

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

- Да можно.

Опишите алгоритм двумерной интерполяции на треугольной конфигурации узлов.

- Алгоритм идентичен, за исключение того что нам надо проверить нужное количество узлов в строке при одномерная интерполяция и использование всех в случае, если интерполяция не одномерная

Код:

```

from math import *

def print_xyz(x,y,z):
    print("    ", end = " ")
    for i in range(12):
        print("{:6}".format(i), end = ' ')
    for i in range(12):
        print("\n{:6}".format(i), end = ' ')
        for j in z[i]:
            print("{:6}".format(j), end = ' ')
def razd_razn(X, Y, n):
    RR = []
    RR.append(X)

```

```

RR.append(Y)

for i in range(n-1):

    T = []

    for j in range(n-i-1):

        T.append((RR[i+1][j]-RR[i+1][j+1])/(RR[0][j] - RR[0][i+j+1]))

    RR.append(T)

return RR

def poly(RR, n, x):

    p = RR[1][0];

    for i in range(1, n):

        tek = 1

        for j in range(i):

            tek = tek*(x-RR[0][j])

        p = p +tek*RR[i+1][0];

    return p;

```

#Функция для набора конфигураций(без учета экстраполяции)

```

def find_nodes(xf, x, count, nodes):

    xn = []

    d = left = 0

    for k in range(count):

        if xf < x[k]:

            xn.append(k)

            d += 1 #кол-во узлов уже записанных

            break

        left += 1

    c = 1

    right = count - left

    while d < nodes:

        if left > 0 :

            xn.append(k - c)

            left -= 1

            d += 1

        if d == nodes:

            break

        if right > 0:

            xn.append(k + c)

            d += 1

            right -= 1

        c += 1

    return min(xn), max(xn)

```

```

def f(x,y):
    return x*x + y*y

def func_matrix():
    x = [0 + i*1 for i in range(12)]
    y = [0 + i*1 for i in range(12)]
    z = [[f(i, j) for i in x] for j in y]

    return x,y,z

def interpolation2(x, y, z, xf, yf, nodesx, nodesy):
    bx, ex = find_nodes(xf, x, len(x), nodesx)
    by, ey = find_nodes(yf, y, len(y), nodesy)

    x = x[bx : ex + 1]
    y = y[by : ey + 1]
    z = z[by : ey + 1]

    for i in range(nodesy):
        z[i] = z[i][bx : (ex + 1)]

    #print(x,y,z)

    res = []

    for i in range(nodesy):
        RR = razd_razn(x, z[i], nodesx)
        res.append(poly(RR, nodesx, xf))

    RR1 = razd_razn(y, res, nodesy)

    return poly(RR1, nodesy, yf)

#Ввод данных
x,y,z = func_matrix()

print_xyz(x,y,z)

xf = float(input("\nInput x: "))
yf = float(input("Input y: "))

nodesx = int(input("Input polynomial degree for x: ")) + 1
nodesy = int(input("Input polynomial degree for y: ")) + 1

#print('индексы x',find_nodes(xf, x, len(x), nodesx))
#print('индексы y',find_nodes(yf, y, len(y), nodesy))

print("Real result: ", f(xf,yf))

found = interpolation2(x, y, z, xf, yf, nodesx, nodesy)

print("Result: ", found)

```
