**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** Пронин А.С.

**Группа** ИУ7-42Б

**Оценка (баллы) \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**

**Преподаватель** Градов В.М.

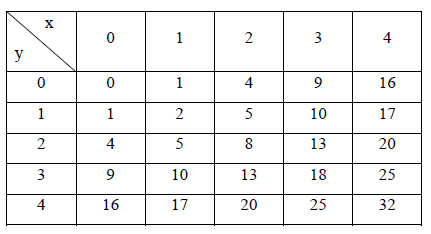
Москва.

2021 г

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

# ****1. Исходные данные****

1. Таблица функции с количеством узлов 5x5



2. Степень аппроксимирующих полиномов - nx и ny. (От 1 до 3)

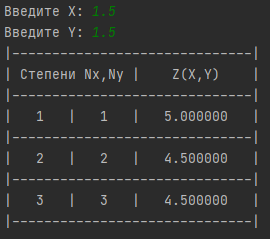
3. Значение аргументов x, y, для которого выполняется интерполяция. (Вводятся пользователем)

# 2. Код программы

z\_table = [ (0, 1, 4, 9, 16),  
 (1, 2, 5, 10, 17),  
 (4, 5, 8, 13, 20),  
 (9, 10, 13, 18, 25),  
 (16, 17, 20, 25, 32) ]  
  
def get\_xy\_array(x, stepen):  
 min\_el = 0  
 min\_id = 0  
 for i in range (1, 4):  
 if abs(x - i) < x - min\_el:  
 min\_el = i  
 min\_id = i  
 rez = [min\_el]  
  
 #x\_array = [0, 1, 2, 3, 4]  
 cur\_id = min\_id  
 for i in range(stepen):  
 if (i % 2 == 0):  
 cur\_id = max(rez) + 1  
 else:  
 cur\_id = min(rez) - 1  
  
 if (cur\_id > 4):  
 cur\_id = min(rez) - 1  
  
 if (cur\_id < 0):  
 cur\_id = max(rez) + 1  
  
 if (cur\_id <= 4 and cur\_id >= 0):  
 rez.append(cur\_id)  
 else:  
 print("get\_x\_array ERROR!")  
  
 rez.sort()  
 return rez  
  
def get\_diff(y, \*args):  
 if len(args) == 0:  
 return None  
 elif len(args) == 1:  
 return y[args[0]]  
 else:  
 return (get\_diff(y, \*args[:-1]) - get\_diff(y, \*args[1:]))/(args[0] - args[-1])  
  
  
def get\_polinom(xi, y):  
 polinom = []  
 for i in range(len(xi)):  
 k = xi[:i]  
 polinom.append(k)  
 diff = get\_diff(y, \*xi[:(i + 1)])  
 polinom.append(diff)  
 return polinom  
  
  
def take\_x(brackets, x):  
 if not brackets:  
 return 1  
 result = 1  
 for bracket in brackets:  
 result \*= (x - bracket)  
 return result  
  
  
def calc\_value(polinom, x):  
 result = 0  
 for i in range(0, len(polinom), 2):  
 result += take\_x(polinom[i], x) \* polinom[i + 1]  
 return result  
  
  
def mult\_interpol(x, y, stepen\_x, stepen\_y):  
 x\_array = get\_xy\_array(x, stepen\_x)  
 y\_array = get\_xy\_array(y, stepen\_y)  
 xi\_values = dict()  
 for i in y\_array:  
 polinom = get\_polinom(x\_array, z\_table[i])  
 xi\_values[i] = calc\_value(polinom, x)  
  
 polinom = get\_polinom(y\_array, xi\_values)  
 z = calc\_value(polinom, y)  
 print("|{:^7d}|{:^7d}".format(stepen\_x, stepen\_y), end = "")  
 print("| {:^12.6f} |".format(z), end = "")  
 print("\n|------------------------------|")  
  
  
def main():  
 x = float(input('Введите X: '))  
 y = float(input('Введите Y: '))  
  
 print("|------------------------------|")  
 print("| Степени Nx,Ny | Z(X,Y) |")  
 print("|------------------------------|")  
 for i in range(1, 4):  
 #for j in range(1, 4):  
 mult\_interpol(x, y, i, i)  
  
if \_\_name\_\_ == '\_\_main\_\_':  
 main()

# 3. Результаты работы

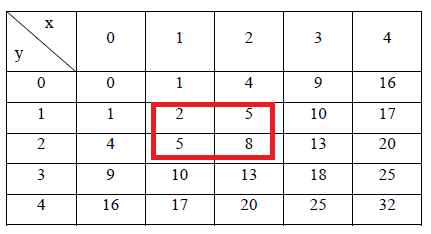
1. Результат интерполяции z(x,y) при степенях полиномов 1,2,3 для x=1.5, y=1.5



# 4. Вопросы при защите лабораторной работы

***1. Пусть производящая функция таблицы суть z(x,y)=x2+y2. Область определения по x и y 0-5 и 0-5. Шаги по переменным равны 1. Степени nx = ny =1, x=y=1.5. Приведите по шагам те. значения функции, которые получаются в ходе последовательных интерполяций. по строкам и столбцу.***

При степени полиномов 1 и x0 = 1.5 на первом шаге будет выбрано два столбца y1 = 1 и y2 = 2. При интерполяции внутри каждого этих столбцов для x0 = 1.5 будет выбрано по две точки — x1 = 1 и x2 = 2. В столбцах функция будет аппроксимироваться полиномом первой степени — прямой, проходящей через две точки конфигурации, а значит будут получены значений — 3.5 для первого столбца и 6.5 для второго. Дальше два полученных значения снова аппроксимируются прямой внутри строки. Аналогично получаем — z(1.5, 1.5) = 5.



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

Лемма: Пусть функция f(x, y) аппроксимируется в точке (х0, у0) по конфигурации 2\*n точек. Тогда степень итогового полинома будет n.

Доказательство:

При интерполяции по столбцам полученные полиномы будут степени n-1 (def). Далее происходит интерполяция внутри строки по двум точкам. То есть справедлива итоговое выражение: f(x, y) = y \* (f(x, y1) – f(x, y2)) / (y1 – y2) + b – уравнение прямой проходящей через f(x, y1) и f(x, y2), значения, в которых определяются полиномом n-1 степени. (Коэффициент b не расписан, так как его степень меньше степени первого слагаемого). Таким образом, при второй интерполяции (внутри строки) степень полинома будет на 1 больше, чем при первой, ч.т.д.

1) Возможные конфигурации: 2\*2, 3\*1, 1\*3. Исходя из леммы, в первом случае степень будет 2. Второй и третий случай являются одномерной интерполяцией, степень также будет 2.

2) Конфигурации: 1\*5, 5\*1, 3\*2, 2\*3. В первом и втором случае — 4. Во втором и третьем (они эквивалентны) — 3.

***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?***

При нерегулярной сетке можно ограничиться интерполяционным полиномом первой степени:

z = a + bx + cy. Его коэффициенты можно найти по трем узлам, выбираемым в окрестности точки интерполяции: zi = a + bxi + cyi, где 0 ≤ i ≤ 2. Узлы не могут располагаться на одной прямой. Так же можно использовать полином второй степени: 𝑧𝑖=𝑎+𝑏𝑥𝑖+𝑐𝑦𝑖+𝑑𝑥𝑖2+𝑔𝑦𝑖2+ℎ𝑥𝑖𝑦𝑖, где 0 ≤ i ≤ 5. Узлы не могут располагаться на одной плоскости.

Заметим, что всегда выбираются узлы, ближайшие к точке интерполяции.

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

Если в двумерной интерполяции используется двумерная таблица значений, то для трёхмерной необходима трёхмерная. Это прямоугольный параллелепипед в ортогональной декартовой системе координат со значениями функций, записанных в его узлах. Пусть исследуемая точка — (x0, y0, z0). Аппроксимируем функцию на необходимом числе «слоёв» как функцию от двух переменных (для фиксированного z = z1, z2 … zn) — получим значения: f(x0, y0, z1), f(x0, y0, z2)...f(x0, y0, zn). По данному набору интерполируем функцию в точке f(x0, y0, z0) как функцию одной переменной

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

Можно. Аналогично предыдущему пункту интерполяцию функции любого количества переменных n можно реализовать, опираясь на алгоритм интерполяции функции n-1 переменной и одномерной интерполяции. То есть, всё сводится к одномерным интерполяциям. При этом реализуемая функция будет только использовать полученные в подфункциях значения, никак не завися от методов интерполяции в каждой из подфункций. Но погрешность в общем случае ухудшится.

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

Необходимо проверять возможность выбора в каждом столбце введённого количества узлов. Если невозможно — интерполировать по максимально возможному количеству узлов (по всем узлам столбца). Об этом и идёт речь в предыдущем пункте