Министерство науки и высшего образования РФ

ФГБОУ ВО «Тамбовский государственный технический университет»

Кафедра САПР

Отчет по лабораторной работе №3

по дисциплине «Численные методы решения инженерных задач»

Выполнил:

студент гр. БВТ-212

Саликов З.Ю.

Проверила:

Майстренко Н.В.

# Тамбов 2022

**Вариант:**

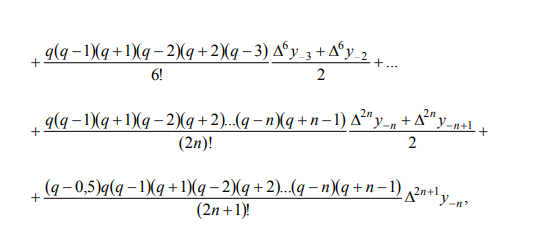
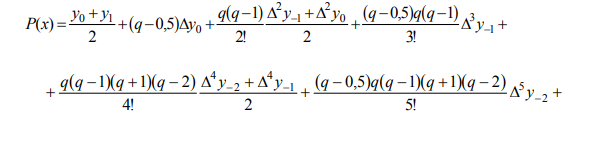
Первая цифра варианта – 6, вторая цифра варианта – 14.

**Задача**:

Для *n+1* точки *x0, x1, …, xn,* называемых узлами интерполяции, и значений некоторой функции *yi = f(xi), i = 0, 1, 2, …, n,* требуется построить интерполяционный полином *Pn(x)* степени не выше *n*, проходящий через узлы интерполяции, т.е. *Pn(x0) = y0; Pn(x1) = y1, …, Pn(xn) = yn,* и вычислить с помощью *Pn(x)* значения функций в заданных точках *xj, j = 1, 2, …, m.*

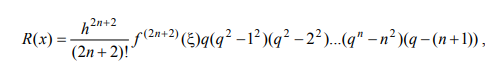
**Решение:**

Интерполяционная формула Бесселя также, как и формула Стирлинга, получается путём некоторых преобразований из формул Гаусса. Она имеет следующий вид:





Остаточный член интерполяционной формулы Бесселя, определяющий погрешность интерполяции, вычисляется по формуле

**Листинг программы:**

import sympy as sym

from sympy import \*

def ucal(u, n):

if (n == 0):

return 1

temp = u

for i in range(1, int(n / 2 + 1)):

temp = temp \* (u - i)

for i in range(1, int(n / 2)):

temp = temp \* (u + i)

return temp

def fact(n):

f = 1

for i in range(2, n + 1):

f \*= i

return f

n = 6;

x = [0.180, 0.185, 0.190, 0.195, 0.200, 0.205]

y = [[0 for i in range(n)]

for j in range(n)]

y[0][0] = 5.61543

y[1][0] = 5.46693

y[2][0] = 5.32634

y[3][0] = 5.19304

y[4][0] = 5.06649

y[5][0] = 4.94619

for i in range(1, n):

for j in range(n - i):

y[j][i] = y[j + 1][i - 1] - y[j][i - 1]

def func(value):

sum = (y[2][0] + y[3][0]) / 2

k = 0

if ((n % 2) > 0):

k = int(n / 2)

else:

k = int(n / 2 - 1)

u = (value - x[k]) / (x[1] - x[0])

for i in range(1, n):

if (i % 2):

sum = sum + ((u - 0.5) \*

ucal(u, i - 1) \* y[k][i]) / fact(i)

else:

sum = sum + (ucal(u, i) \* (y[k][i] + y[k - 1][i]) / (fact(i) \* 2))

k -= 1

return sum

value = [0.1838, 0.1875, 0.1944, 0.1976, 0.2038]

for i in value:

print("X:", i, "Y:", round(func(i),5))

x\_sym = sym.Symbol('x')

polinom = func(x\_sym)

print("ПОЛИНОМЧИК:")

print(simplify(polinom))

h = 0.005

derivative\_f = polinom

q = (x\_sym - x[0]) / h

q2 = q

factorial = 1

for j in range(1, n+1):

q2 \*= (q\*\*2 - j\*\*2)

for i in range(1, n):

derivative\_f = derivative\_f.diff(x\_sym)

for i in range(2, 2\*n+3):

factorial \*= i

res\_mem = ((h \*\* (2\*n + 2)) \* q2 \* derivative\_f \* value[0]) \* (q-(n+1)) / factorial

print("Остаточный член: ", '\n', res\_mem)

**Результат работы программы:**

К сожалению, результат слишком длинный относительно консоли.

Вывод программы:

**X: 0.1838 Y: 5.50182**

**X: 0.1875 Y: 5.39569**

**X: 0.1944 Y: 5.20867**

**X: 0.1976 Y: 5.12642**

**X: 0.2038 Y: 4.97452**

**ПОЛИНОМЧИК:**

**-106666.666676839\*x\*\*5 + 106666.666676484\*x\*\*4 - 43320.0000037878\*x\*\*3 + 9050.33333406371\*x\*\*2 - 1006.40193340372\*x + 54.3602100027115**

**Остаточный член:**

**-1.64712737583108e-37\*(200.0\*x - 43.0)\*(200.0\*x - 36.0)\*(40000.0\*(x - 0.18)\*\*2 - 36)\*(40000.0\*(x - 0.18)\*\*2 - 25)\*(40000.0\*(x - 0.18)\*\*2 - 16)\*(40000.0\*(x - 0.18)\*\*2 - 9)\*(40000.0\*(x - 0.18)\*\*2 - 4)\*(40000.0\*(x - 0.18)\*\*2 - 1)**

**Графики:**

|  |
| --- |
|  |
|  |

**Вывод**:

С помощью первой интерполяционной формулы Бесселя был построен интерполяционный полином, вычислены значения функций в заданных точках и построен график, иллюстрирующий узлы интерполяции и результат интерполяции.