Министерство науки и высшего образования РФ

Федеральное государственное бюджетное образовательное учреждение

высшего образования

«Курский государственный университет»

Кафедра программного обеспечения и администрирования информационных систем

Направление подготовки математическое обеспечение и администрирование информационных систем

Форма обучения очная

**Отчет**

**по лабораторной работе №5**

«Решение систем линейных уравнений методом Халецкого»

дисциплина «Методы вычислений»

Выполнил:

студент группы 313.1 Козявин М.С.

Проверил:

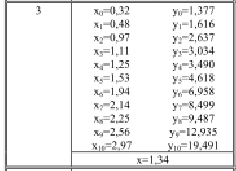
доцент кафедры АГиТОМ Селиванова И. В.

Курск, 2023

Цель: Изучение алгоритма численного решения систем методом Халецкого.

**Задание:**

1. Разработать алгоритм и программу вычисления значения функции в промежуточной точке по экспериментальным данным для не равноотстоящих узлов.



**Код программы:**

import math

import matplotlib.pyplot as plt

data = [

    [0.32, 1.377],

    [0.73, 2.075],

    [0.97, 2.637],

    [1.13, 3.095],

    [1.52, 4.572],

    [1.57, 4.806],

    [2.02, 7.538],

    [2.52, 12.428],

    [2.96, 19.297],

    [3.40, 39.964],

    [3.79, 44.256]

]

coef = [[0 for \_\_ in range(len(data))] for \_ in range(len(data))]

*def* calc\_l(*n*):

    d = 1

    c = [[0 for \_\_ in range(2)] for \_ in range(len(data)-1)]

    correct = 0

    for i in range(len(data)):

        if i == n:

            correct = -1

            continue

        d \*= data[n][0] - data[i][0]  *# знаменатель*

        c[i+correct][0] = -data[i][0]  *# числ*

        c[i+correct][1] = 1

    l = 3

    while len(c) > 1:

        tmp = [0 for \_ in range(l)]

        for i in range(len(c[0])):

            for j in range(2):

                tmp[i+j] += c[0][i] \* c[1][j]

        c = [tmp, \*c[2:]]

        l += 1

    for i in range(len(c[0])):

        c[0][i] /= d

    return c[0]

*def* f(*x*, *nums*):

    res = 0

    for i in range(len(nums)):

        res += nums[i]\*math.pow(x, i)

    return res

*def* draw\_graph(*nums*, *size*=100, *dots*=False, *style*="solid"):

    xs = [x / size for x in range(-size\*10, size\*10)]

    ys = [f(x, nums) for x in xs]

    plt.plot(xs, ys, *linestyle*=style)

    if dots:

        for i in range(len(data)):

            print(data[i][0], data[i][1])

            plt.scatter(data[i][0], data[i][1], *color*="black")

r = [0 for \_ in range(len(data))]

for x in range(len(data)):

    calc\_r = calc\_l(x)

    draw\_graph(calc\_r, 100, False, "dotted")

    for i in range(len(data)):

        r[i] += data[x][1]\*calc\_r[i]

for i in range(len(r)):

    if r[i] != 0:

        if i == len(r)-1:

            if r[i] > 0:

                print(" + ", *end*="")

            else:

                print(" - ", *end*="")

            print(abs(r[i]), *end*="")

        else:

            if r[i] > 0:

                print(" + ", *end*="")

            else:

                print(" - ", *end*="")

            print(*str*(abs(r[i])) + " X" + *str*(len(r)-i-1), *end*="")

print()

print(f(1.96, r))

draw\_graph(r, 100, True)

plt.grid()

plt.show()

**Тестирование:**

