Національний технічний університет України

«Київський політехнічний інститут імені Ігоря Сікорського»

Факультет інформатики та обчислювальної техніки

Кафедра обчислювальної техніки

Алгоритми та методи обчислень

Лабораторна робота №3

«**Розв’язання нелінійних рівнянь на комп’ютері**»

Виконав:

Студент групи ІВ-82

Троценко Д.А.

Номер залікової книжки 8227

Номер у списку 25

Перевірив:

ст.вик. Порєв В. М.

Київ

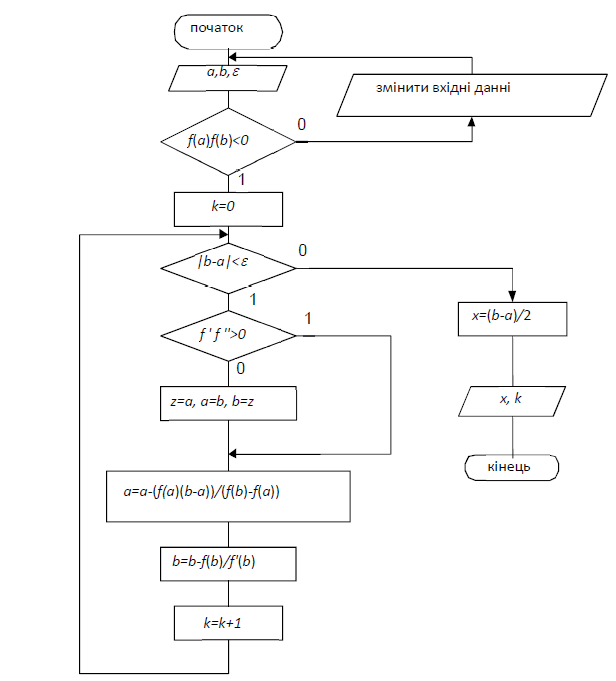
2020 р.

**Мета:** ознайомлення з методиками та вивчення різних алгоритмів розв’язання нелінійних рівнянь на комп’ютері.

**Завдання:** Закріплення знань студентів при вирішенні практичних завдань з розв’язування нелінійних рівнянь. Оволодіння методами і практичними навичками розв’язування нелінійних рівнянь на комп’ютері. Набуття умінь і навичок при програмуванні та налагодженні програм для розв’язування нелінійних рівнянь на комп'ютері.

|  |  |  |
| --- | --- | --- |
| № Варіанта |  | Метод |
| 25 |  | Комбінований метод |

Блок-схема:



Текст програми:

lagranz\_algo.py

def lagranz(x,y,t):

    z=0

    for j in range(len(y)):

        p1=1; p2=1

        for i in range(len(x)):

            if i==j:

                p1=p1\*1; p2=p2\*1

            else:

                p1=p1\*(t-x[i])

                p2=p2\*(x[j]-x[i])

        z=z+y[j]\*p1/p2

    return z

main.py

import numpy as np

import matplotlib.pyplot as plt

from lagranz\_algo import lagranz

import sys

from PyQt5.QtWidgets import (QInputDialog, QApplication, QLineEdit, QGridLayout, QLabel, QWidget, QToolTip, QTextEdit, QPushButton, QApplication, QMessageBox, QDesktopWidget, QMainWindow)

from algo import hord\_nuton

class MainWindow(QWidget):

    def \_\_init\_\_(self):

        super().\_\_init\_\_()

        self.initUI()

    def initUI(self):

        self.setGeometry(300, 300, 250, 150)

        lbl1 = QLabel('Лабораторна робота №4 - \nКомбінований метод\n2xlnx-1 - \n25 варіант')

        lbl2 = QLabel('Студент групи ІВ-82')

        lbl3 = QLabel('Троценко\nДанііл\nАнатолійович')

        btn1 = QPushButton('Показати графік')

        btn1.clicked.connect(self.Lag)

        label\_a = QLabel('a = ')

        label\_b = QLabel('b = ')

        label\_i = QLabel('i = ')

        self.edit\_a = QLineEdit()

        self.edit\_b = QLineEdit()

        self.edit\_i = QLineEdit()

        self.edit\_a.setText('1')

        self.edit\_b.setText('3')

        self.edit\_i.setText('10')

        grid = QGridLayout()

        grid.setSpacing(5)

        grid.addWidget(lbl1,0,0)

        grid.addWidget(lbl2,0,1)

        grid.addWidget(lbl3,0,2)

        grid.addWidget(label\_a,1,0)

        grid.addWidget(self.edit\_a,1,1)

        grid.addWidget(label\_b,2,0)

        grid.addWidget(self.edit\_b,2,1)

        grid.addWidget(label\_i,3,0)

        grid.addWidget(self.edit\_i,3,1)

        grid.addWidget(btn1, 4,1)

        self.setLayout(grid)

        self.setGeometry(300, 300, 500, 500)

        self.setWindowTitle('Lab-3')

        self.show()

    def Lag(self):

        a = int(self.edit\_a.text())

        b = int(self.edit\_b.text())

        k = int(self.edit\_i.text()) + 1

        f = lambda x: 2\*x\*np.log(x) - 1

        x=np.array([ a + i\*(b-a)/10 for i in range(k)], dtype=float)

        y = f(x)

        xnew=np.linspace(np.min(x),np.max(x), k)

        ynew=[lagranz(x,y,i) for i in xnew]

        fig, ax = plt.subplots()

        ax.plot(x,y, label='2x-lnx')

        ax.plot(xnew,ynew,'o', label='Інтерполяція Многочленом Лагранжа')

        ax.plot(hord\_nuton(), f(hord\_nuton()),'o', label='Корінь')

        ax.legend()

        fig.set\_figheight(5)

        fig.set\_figwidth(8)

        plt.grid(True)

        plt.show()

        plt.plot(x\_new\_end, errors)

        plt.show

if \_\_name\_\_ == "\_\_main\_\_":

    app = QApplication(sys.argv)

    ex = MainWindow()

    sys.exit(app.exec\_())

algo.py

import numpy as np

def hord\_nuton():

    a,b = 0.1, 3.

    e = 0.0001

    f = lambda x: 2\*x\*np.log(x) - 1

    f1 = lambda x: 2\*np.log(x) + 2

    f2 = lambda x: 2/x

    res = 0

    while  (np.abs(a-b)>2\*e):

        if (f(a) \* f2(a)<0):

            a += (b-a) / (f(a) - f(b)) \* f(a)

        else:

            a -= f(a)/f1(a)

        if (f(b) \* f2(b) < 0):

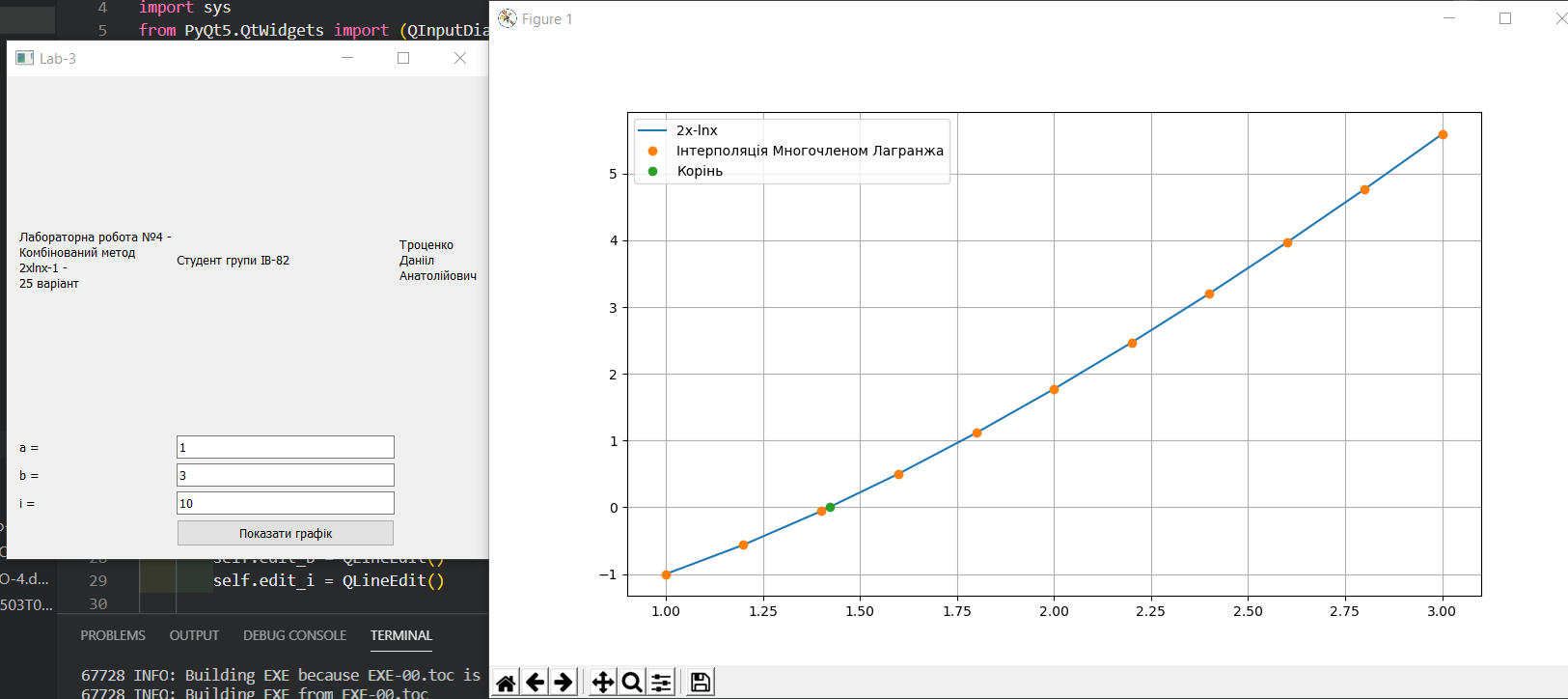
            b += (a-b) / (f(b) - f(a)) \* f(b)

        else:

            b -= f(b) / f1(b)

    return (a+b)/2.0

Результат роботи програми:



Висновок:

У ході виконання лабораторної роботи я ознайомився з методиками різних алгоритмів розв’язання нелінійних рівнянь на комп’ютері, реалізував комбінований метод на мові Python.