Національний технічний університет України

«Київський політехнічний інститут імені Ігоря Сікорського»

Факультет інформатики та обчислювальної техніки

Кафедра обчислювальної техніки

Алгоритми та методи обчислення

Лабораторна робота №5

**«Розв’язання систем лінійних алгебраїчних рівнянь на комп’ютері»**

Виконала:

студентка групи ІО-64

Бровченко А. В.

Перевірив:

ст.вик. Порєв В. М.

Київ

2018 р.

**Мета:** Вивчити алгоритми методів розв'язання систем лінійних алгебраїчних

рівнянь на ЕОМ.

**Завдання:** Відповідно до варіанту завдання скласти схему алгоритму

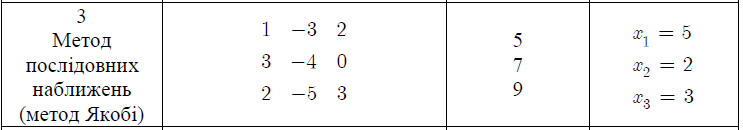
розв’язання систем лінійних алгебраїчних рівнянь зазначеним у варіанті

методом. Відповідно до блок-схеми скласти програму розв'язання систем

лінійних алгебраїчних рівнянь алгоритмічною мовою, узгодженою з

викладачем. Розв’язати СЛАР на ЕОМ відповідно до варіанту.

**За варіантом:**



**Роздруківка тексту програми:**

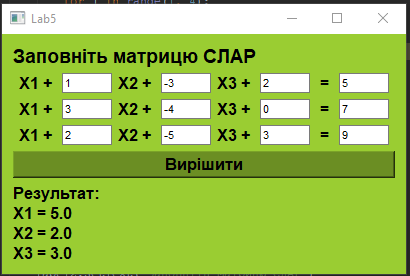
algorythm.py

import numpy as np  
  
X = np.array([[ 1., -3., 2.],  
 [ 3., -4., 0.],  
 [ 2., -5., 3.]])  
Y = np.array([[1.], [2.], [2.]])  
  
A = np.concatenate((X, Y), axis=1)  
  
def solve\_sistem(X,Y):  
 A = np.concatenate((X, Y), axis=1)  
 # Приведение матрицы к треугольному виду  
 for i in range(len(A)):  
 amax = A[i][i]  
 for j in range(i+1, len(A)):  
 if A[j][i]>amax:  
 amax = A[j][i]  
 l = j  
  
 a = np.array(A[i])  
 b = np.array(A[l])  
 A[i] = b  
 A[l] = a  
  
 for i in range(len(A)):  
 A[i] = A[i] / A[i][i]  
 for k in range(len(A)):  
 if k!=i:  
 A[k] = A[k] - A[k][i]\*A[i]  
 return A[0:3,3]  
  
if \_\_name\_\_ == '\_\_main\_\_':  
 X = np.array([[1., -3., 2.],  
 [3., -4., 0.],  
 [2., -5., 3.]])  
 Y = np.array([[5.], [7.], [9.]])  
 print(solve\_sistem(X,Y))  
# A = np.array([A[i] for i in range(len(A)-1, -1, -1)])  
# print(A)  
#  
# print()  
# for i in range(len(A)):  
# # print(A[i][i-1])  
# # A[i] = A[i] / A[i][i]  
# print(A)  
#  
# for k in range(len(A)):  
# A[k] = A[k] - A[k][i]\*A[i+1]  
#  
# print(A)

main.py

import sys  
  
import matplotlib.pyplot as plt  
import numpy as np  
from PyQt5.QtGui import \*  
from PyQt5.QtWidgets import \*  
from third\_sem.AMO.Lab5\_Nastusha.algorithm import solve\_sistem  
  
  
class Example(QWidget):  
  
 def \_\_init\_\_(self):  
 super().\_\_init\_\_()  
  
 self.initUI()  
  
 def initUI(self):  
 grid = QGridLayout()  
 self.setLayout(grid)  
 self.move(300, 150)  
 self.setWindowTitle('Lab5')  
 self.font = QFont("Arial", 8, QFont.Bold)  
 self.font.setPointSize(11)  
 self.setStyleSheet("background-color:yellowgreen")  
 self.TEXTSIZE = 12  
  
  
 x = [[1, -3, 2],  
 [3, -4, 0],  
 [2, -5, 3]]  
 y = [5, 7, 9]  
  
 self.x = [[QLineEdit() for \_ in range(3)] for \_ in range(3)]  
 for i in range(len(self.x)):  
 for j in range(len(self.x[0])):  
 self.x[i][j].setText(str(x[i][j]))  
 self.x[i][j].setMaximumSize(50, 50)  
 self.x[i][j].setStyleSheet("background-color:white")  
 self.y = [QLineEdit() for \_ in range(3)]  
 for i in range(len(self.x)):  
 self.y[i].setText(str(y[i]))  
 self.y[i].setMaximumSize(50, 50)  
 self.y[i].setStyleSheet("background-color:white")  
  
 self.x\_lable = [[QLabel('X' + str(i + 1) + ' + ') for i in range(3)] for \_ in range(3)]  
  
 for i in self.x\_lable:  
 for j in i:  
 j.setFont(QFont("Arial", self.TEXTSIZE, QFont.Bold))  
 self.equal = [QLabel(' = ') for i in range(3)]  
 for i in self.equal:  
 i.setFont(QFont("Arial", self.TEXTSIZE, QFont.Bold))  
  
  
  
 self.xlable = QLabel('X = ?')  
  
 button\_show\_ideal = QPushButton("Вирішити")  
 button\_show\_ideal.clicked.connect(self.solve)  
 button\_show\_ideal.setFont(QFont("Arial", self.TEXTSIZE, QFont.Bold))  
 button\_show\_ideal.setStyleSheet("background-color:olivedrab")  
  
  
 for i in range(3):  
 for j in range(1, 4):  
 grid.addWidget(self.x\_lable[i][j - 1], i + 2, (j) \* 2 - 1)  
  
 grid.addWidget(self.x[i][j - 1], i + 2, (j) \* 2)  
 print(self.equal[i])  
  
 grid.addWidget(self.equal[i], i + 2, (j) \* 2 + 1)  
 grid.addWidget(self.y[i], i + 2, (j) \* 2 + 2)  
  
 grid.addWidget(button\_show\_ideal, 7, 0, 1, 10)  
  
 self.rez = QLabel('')  
 grid.addWidget(self.rez, 9, 0, 1, 10)  
  
  
 label2 = QLabel(self)  
 label2.setText("Заповніть матрицю СЛАР")  
 label2.setFont(QFont("Arial", 14, QFont.Bold))  
 grid.addWidget(label2, 1, 0, 1, 10)  
  
  
 self.show()  
  
 def solve(self):  
 X = np.array([i[0] for i in np.linalg.solve(\*self.get\_matrixes())]).round(4)  
  
 self.rez.setText('Результат:\n'  
 'X1 = {}\n'  
 'X2 = {}\n'  
 'X3 = {}'.format(\*X))  
 self.rez.setFont(QFont("Arial", self.TEXTSIZE, QFont.Bold))  
  
  
 def get\_matrixes(self):  
 A = np.array([[float(j.text()) for j in i] for i in self.x])  
 Y = np.array([[float(j.text())] for j in self.y])  
 return A, Y  
  
  
if \_\_name\_\_ == '\_\_main\_\_':  
 app = QApplication(sys.argv)  
 ex = Example()  
 sys.exit(app.exec\_())

**Скріншоти виконання програми:**

****

**Висновки:** У ході виконання лабораторної роботи я ознайомилась з методиками різних алгоритмів розв’язання нелінійних рівнянь на комп’ютері, реалізувала метод половинного ділення на мові Python.