Міністерство освіти і науки України

Національний технічний університет України

«Київський політехнічний інститут ім. Ігоря Сікорського»

Факультет інформатики та обчислювальної техніки

Кафедра обчислювальної техніки

ЛАБОРАТОРНА РОБОТА № 5

з дисципліни «Алгоритми та методи обчислень»

на тему «Розв’язання систем лінійних алгебраїчних рівнянь»

ВИКОНАЛА:

студентка 2 курсу

групи ІВ-92

Бабенко В.В.

Залікова - 9201

ПЕРЕВІРИВ:

Доцент кафедри ОТ

Порєв В.М.

Київ - 2021

**Хід роботи**

**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

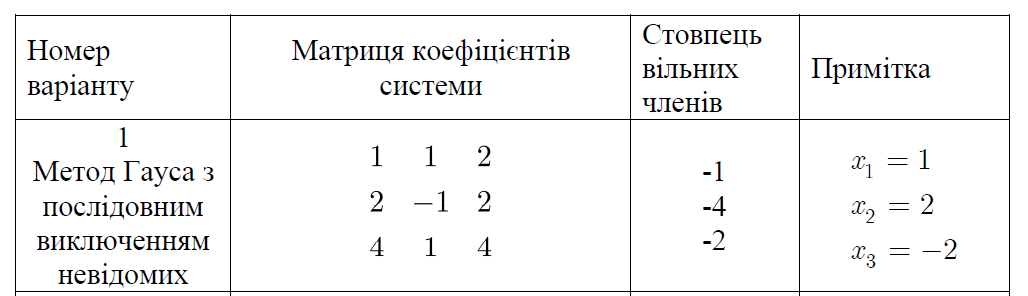
методом. Відповідно до блок-схеми скласти програму розв'язання систем

лінійних алгебраїчних рівнянь алгоритмічною мовою, узгодженою з

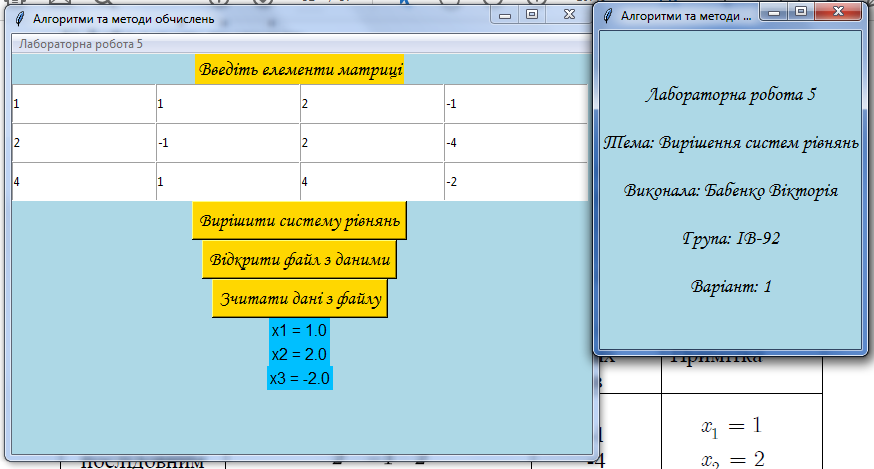
викладачем. Розв’язати СЛАР на комп’ютері відповідно до варіанту.

**Варіанти завдання**

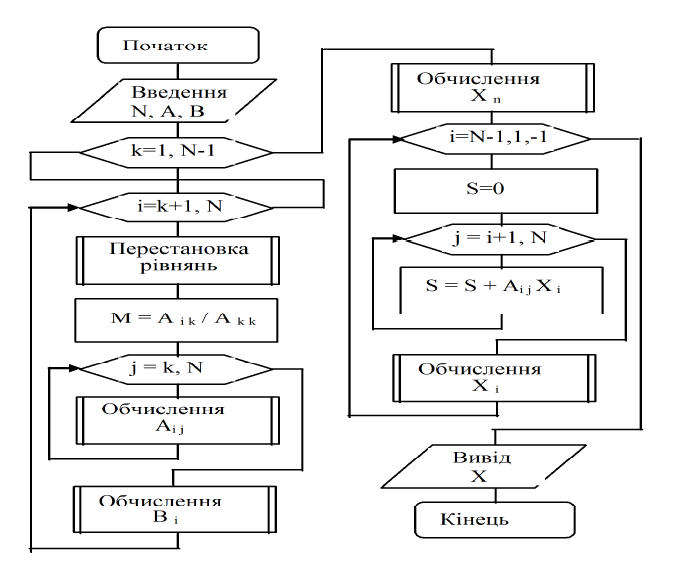
Варіант1



**Результати роботи програми**



**Блок-схеми алгоритмів**



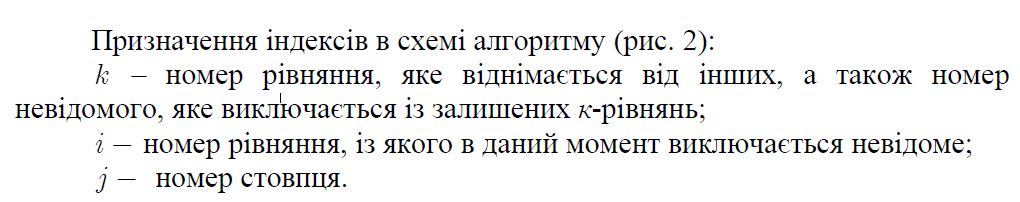
Блок-схема алгоритму методу Гаусса з послідовним виключенням невідомих

*Текстовий опис:*

У даному алгоритмі на вхідні значення подаються значення (A, B, N). Прямий хід методу Гауса. Перевіримо, щоб принаймні один із коефіцієнтів a11 , a21 , a31 не дорівнював нулю. Обчислюється множник: М2=а21/а11. Перше рівняння множимо на М2 і віднімаємо від другого рівняння, таким чином виключаємо х1. Далі обчислюємо Мв = ав1/а11, множимо цей множник на перше рівняння і віднімаємо від третього рівняння, тоді коефіцієнт при х1 стає нулем. Таким чином, ми позбулися від змінної х1, всі ті ж маніпуляції проводимо до зведення матриці до трикутної матриці коефіцієнтів. Починаємо зворотній хід методу Гауса Знайти розв’язок такої системи просто: із

3го рівняння знайти x3, підставити результат у друге і знайти x2 , підставити

x2 і x3 в 1-е рівняння системи і знайти 1 x .



**Роздруківка тексту програми**

from tkinter import \*  
from tkinter import messagebox  
import subprocess  
import file1  
root = Tk()  
root.geometry(**"580x400"**)  
root.title(**"Алгоритми та методи обчислень"**)  
root.config(bg=**"lightblue"**)  
  
label1 = Label(root, text=**"Введіть елементи матриці"**, font=(**'Monotype Corsiva'**, 15))  
label1.config(bg=**"gold"**)  
a11\_entry = Entry(root)  
a12\_entry = Entry(root)  
a13\_entry = Entry(root)  
a14\_entry = Entry(root)  
a21\_entry = Entry(root)  
a22\_entry = Entry(root)  
a23\_entry = Entry(root)  
a24\_entry = Entry(root)  
a31\_entry = Entry(root)  
a32\_entry = Entry(root)  
a33\_entry = Entry(root)  
a34\_entry = Entry(root)  
  
  
def solve\_task():  
 global a11\_entry  
 global a12\_entry  
 global a13\_entry  
 global a14\_entry  
 global a21\_entry  
 global a22\_entry  
 global a23\_entry  
 global a24\_entry  
 global a31\_entry  
 global a32\_entry  
 global a33\_entry  
 global a34\_entry  
  
  
 def bubble\_max\_row(m, col):  
 *# Replace matrix[col] row with the one of the underlying rows with the modulo greatest first element.  
 # :param matrix: matrix (list of lists)  
 # :param col: index of the column/row from which underlying search will be launched  
 # :return: None. Function changes the matrix structure.* max\_element = m[col][col]  
 max\_row = col  
 for i in range(col + 1, len(m)):  
 if abs(m[i][col]) > abs(max\_element):  
 max\_element = m[i][col]  
 max\_row = i  
 if max\_row != col:  
 m[col], m[max\_row] = m[max\_row], m[col]  
  
 def display\_results(x):  
 x1\_label[**"text"**] = **"x1 = "** + str(x[0])  
 x2\_label[**"text"**] = **"x2 = "** + str(x[1])  
 x3\_label[**"text"**] = **"x3 = "** + str(x[2])  
 for i in x:  
 print(i)  
  
 def solve\_gauss(m):  
 *"""Solve linear equations system with gaussian method.  
 :param m: matrix (list of lists)  
 :return: None  
 """* n = len(m)  
 *# forward trace* for k in range(n - 1):  
 bubble\_max\_row(m, k)  
 for i in range(k + 1, n):  
 div = m[i][k] / m[k][k]  
 m[i][-1] -= div \* m[k][-1]  
 for j in range(k, n):  
 m[i][j] -= div \* m[k][j]  
  
 *# check modified system for nonsingularity* if is\_singular(m):  
 print(**'The system has infinite number of answers...'**)  
 return  
  
 *# backward trace* x = [0 for i in range(n)]  
 for k in range(n - 1, -1, -1):  
 x[k] = (m[k][-1] - sum([m[k][j] \* x[j] for j in range(k + 1, n)])) / m[k][k]  
  
 *# Display results* display\_results(x)  
  
 def is\_singular(m):  
 *"""Check matrix for nonsingularity.  
 :param m: matrix (list of lists)  
 :return: True if system is nonsingular  
 """* for i in range(len(m)):  
 if not m[i][i]:  
 return True  
 return False  
  
 matrix = [[0, 0, 0, 0], [0, 0, 0, 0], [0, 0, 0, 0]]  
 try:  
 matrix[0][0] = float(a11\_entry.get())  
 matrix[0][1] = float(a12\_entry.get())  
 matrix[0][2] = float(a13\_entry.get())  
 matrix[0][3] = float(a14\_entry.get())  
  
 matrix[1][0] = float(a21\_entry.get())  
 matrix[1][1] = float(a22\_entry.get())  
 matrix[1][2] = float(a23\_entry.get())  
 matrix[1][3] = float(a24\_entry.get())  
  
 matrix[2][0] = float(a22\_entry.get())  
 matrix[2][1] = float(a22\_entry.get())  
 matrix[2][2] = float(a22\_entry.get())  
 matrix[2][3] = float(a22\_entry.get())  
 except:  
 messagebox.showinfo(**"Error"**, **"Введіть значення!!!"**)  
  
   
  
  
 solve\_gauss(matrix)  
  
def open\_file():  
 subprocess.call([**"notepad.exe"**, **"file1.py"**])  
  
def read\_file():  
 def bubble\_max\_row(m, col):  
 *# Replace matrix[col] row with the one of the underlying rows with the modulo greatest first element.  
 # :param matrix: matrix (list of lists)  
 # :param col: index of the column/row from which underlying search will be launched  
 # :return: None. Function changes the matrix structure.* max\_element = m[col][col]  
 max\_row = col  
 for i in range(col + 1, len(m)):  
 if abs(m[i][col]) > abs(max\_element):  
 max\_element = m[i][col]  
 max\_row = i  
 if max\_row != col:  
 m[col], m[max\_row] = m[max\_row], m[col]  
  
 def display\_results(x):  
 x1\_label[**"text"**] = **"x1 = "** + str(x[0])  
 x2\_label[**"text"**] = **"x2 = "** + str(x[1])  
 x3\_label[**"text"**] = **"x3 = "** + str(x[2])  
 for i in x:  
 print(i)  
  
 def solve\_gauss(m):  
 *"""Solve linear equations system with gaussian method.  
 :param m: matrix (list of lists)  
 :return: None  
 """* n = len(m)  
 *# forward trace* for k in range(n - 1):  
 bubble\_max\_row(m, k)  
 for i in range(k + 1, n):  
 div = m[i][k] / m[k][k]  
 m[i][-1] -= div \* m[k][-1]  
 for j in range(k, n):  
 m[i][j] -= div \* m[k][j]  
  
 *# check modified system for nonsingularity* if is\_singular(m):  
 x1\_label[**"text"**] = **"Система має нескінченну кількість розв'язків"** x2\_label[**"text"**] = **""** x3\_label[**"text"**] = **""** print(**'The system has infinite number of answers...'**)  
 return  
  
 *# backward trace* x = [0 for i in range(n)]  
 for k in range(n - 1, -1, -1):  
 x[k] = (m[k][-1] - sum([m[k][j] \* x[j] for j in range(k + 1, n)])) / m[k][k]  
  
 *# Display results* display\_results(x)  
  
 def is\_singular(m):  
 *"""Check matrix for nonsingularity.  
 :param m: matrix (list of lists)  
 :return: True if system is nonsingular  
 """* for i in range(len(m)):  
 if not m[i][i]:  
 return True  
 return False  
  
 matrix = file1.matrix  
  
 solve\_gauss(matrix)  
  
  
solve\_button = Button(root, text=**"Вирішити систему рівнянь"**, font=(**'Monotype Corsiva'**, 14), command=solve\_task)  
open\_button = Button(root, text=**"Відкрити файл з даними"**, font=(**'Monotype Corsiva'**, 14), command=open\_file)  
read\_button = Button(root, text=**"Зчитати дані з файлу"**, font=(**'Monotype Corsiva'**, 14), command=read\_file)  
  
solve\_button.config(bg=**"gold"**)  
open\_button.config(bg=**"gold"**)  
read\_button.config(bg=**"gold"**)  
  
x1\_label = Label(root, text=**"x1 ="**, font=10)  
x2\_label = Label(root, text=**"x2 ="**, font=10)  
x3\_label = Label(root, text=**"x3 ="**, font=10)  
x1\_label.config(bg=**"deep skyblue"**)  
x2\_label.config(bg=**"deep skyblue"**)  
x3\_label.config(bg=**"deep skyblue"**)  
label1.grid(row=0, column=1, columnspan=4)  
  
a11\_entry.grid(row=1, column=1, ipadx=10, ipady=10)  
a12\_entry.grid(row=1, column=2, ipadx=10, ipady=10)  
a13\_entry.grid(row=1, column=3, ipadx=10, ipady=10)  
a14\_entry.grid(row=1, column=4, ipadx=10, ipady=10)  
  
a21\_entry.grid(row=2, column=1, ipadx=10, ipady=10)  
a22\_entry.grid(row=2, column=2, ipadx=10, ipady=10)  
a23\_entry.grid(row=2, column=3, ipadx=10, ipady=10)  
a24\_entry.grid(row=2, column=4, ipadx=10, ipady=10)  
  
a31\_entry.grid(row=3, column=1, ipadx=10, ipady=10)  
a32\_entry.grid(row=3, column=2, ipadx=10, ipady=10)  
a33\_entry.grid(row=3, column=3, ipadx=10, ipady=10)  
a34\_entry.grid(row=3, column=4, ipadx=10, ipady=10)  
  
solve\_button.grid(row=4, column=1, columnspan=4)  
open\_button.grid(row=5, column=1, columnspan=4)  
read\_button.grid(row=6, column=1, columnspan=4)  
  
  
x1\_label.grid(row=7, column=1, columnspan=4)  
x2\_label.grid(row=8, column=1, columnspan=4)  
x3\_label.grid(row=9, column=1, columnspan=4)  
  
  
  
def hello():  
 global win1  
 if win1 == 0:  
 win1 = Toplevel(root)  
  
 label = Label(win1, text=**"**\n\n**Лабораторна робота 5**\n\n**Тема: Вирішення систем рівнянь**\n**"  
 "**\n**Виконала: Бабенко Вікторія**\n\n**Група: ІВ-92**\n\n**Варіант: 1**\n\n**"**,  
 font=(**'Monotype Corsiva'**, 15), bg=**"lightblue"**)  
 label.pack()  
 win1.config(bg=**"gold"**)  
 else:  
 win1.destroy()  
 win1 = 0  
  
mainmenu = Menu(root)  
root.config(menu=mainmenu)  
  
  
lab5 = Menu(mainmenu, tearoff=0)  
  
lab5.add\_command(label=**"Автор"**, command=hello)  
  
mainmenu.add\_cascade(label=**"Лабораторна робота 5"**, menu=lab5)  
  
win1 = 0  
  
  
root.mainloop()

**Аналіз результатів**

Реалізація алгоритму методу Гаусса з послідовним виключенням невідомих. Для матриць обмеженого розміру метод є менш трудомістким в порівнянні з іншими методами. Метод дозволяє однозначно встановити чи сумісна система, у випадку сумісності, знаходимо рішення. Метод Гаусса дозволяє знайти максимальне число лінійно-незалежних рівнянь(ранг матриці). Вище було перераховано переваги методу Гауса. Щодо недоліків методу: метод Гауса не буде оптимальним по швидкості для СЛАР з великою розмірністю, у 1969 році Штрассен довів, що матриці можна перемножити за час О(n^log2(7))=O(n^2.81). Саме з цього випливає, що перетворення матриць і рішення СЛАР можна здійснювати алгоритмами асимптотично більш швидкими по порядку ніж метод Гаусса. Рішено систему рівнянь методо Гаусса, це показано на скріншотах результату.

**Висновки**

В ході виконання цієї роботи було закрілено навички програмного розв’язання систем лінійних рівнянь. Програмно був реалізований метод Гаусса з послідовним виключенням невідомих. Окрім того були закріплені основні навички роботи з графічними бібліотекою tkinter. Всі методи в основному було реалізовано вручну. Отримані результати виконання програми є правильними, це перевірено на тестовому прикладі. Кінцевої мети досягнуто.