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 № 4

з дисципліни «Алгоритми та методи обчислень»

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

ВИКОНАЛА:

студентка 2 курсу

групи ІВ-92

Бабенко В.В.

Залікова - 9201

ПЕРЕВІРИВ:

Доцент кафедри ОТ

Порєв В.М.

Київ - 2021

**Хід роботи**

**Мета**: Метою даного заняття є 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

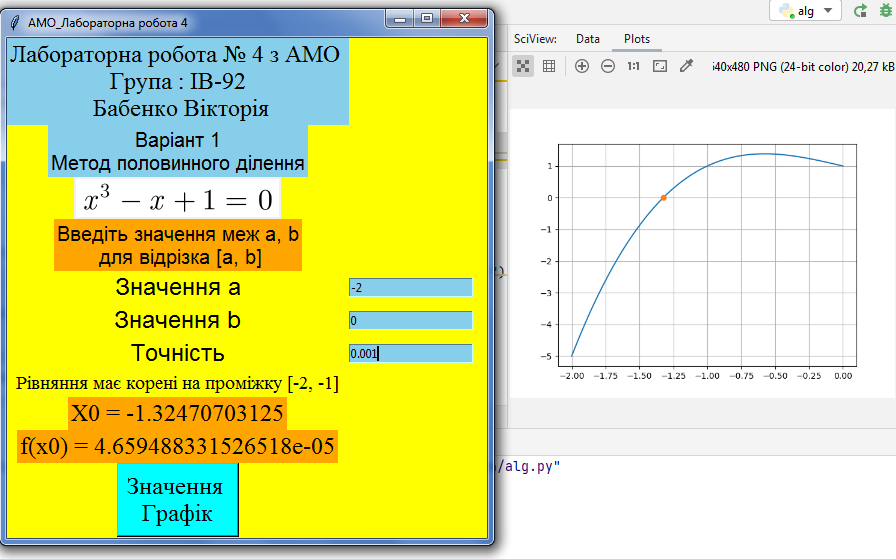
нелінійних рівнянь на комп'ютері.

**Варіанти завдання**

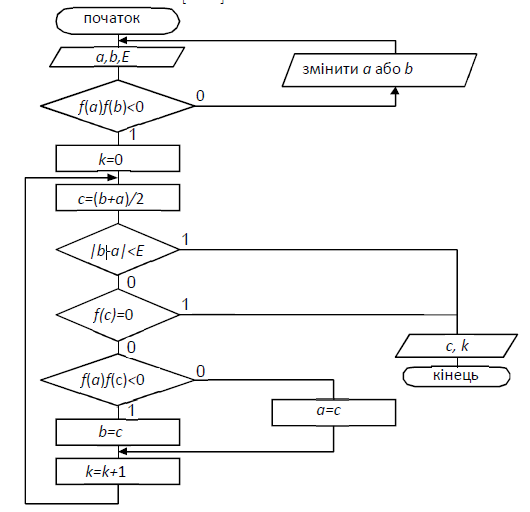
Варіант1(Метод половинного ділення)



**Результати роботи програми**



**Блок-схеми алгоритмів**



Блок-схема алгоритму методу половинного ділення

*Текстовий опис:*

У даному алгоритмі на вхідні значення подаються значення кінців відрізка для пошуку кореня та похибка (а, b, E). У розгалуженому циклі виконується перевірка на від’ємний знак добутку функції у кінцях відрізка аb, якщо умова не виконується, то потрібно змінити значення a, b. Оголошуємо лічильник k=0 та значення с(1), яке чисельно рівне півдобутку суми а та b. Виконуємо перевірку: якщо абсолютна величина різниці меж відрізка ab менша значення похибки, то виводимо значення лічильника та с, у протилежному випадку прирівнюємо значення функції в точці с до нуля, якщо справджується, то виводимо значення с та k, завершення програми, якщо ні, то виконуємо перевірку на знак добутку значень функції на кінцях вибраного діапазону. Якщо умова справджується, то присвоюємо значення b = c, в іншому випадку a = c. Додаємо до лічильника 1 і повертаємось до (1).

**Роздруківка тексту програми**

import math  
import matplotlib.pyplot as plt  
import numpy as np  
from tkinter import \*  
from tkinter import messagebox  
  
root = Tk()  
root.title(**"АМО\_Лабораторна робота 4"**)  
root.geometry(**'480x500'**)  
root[**"bg"**] = **"yellow"**a\_ent = Entry(root)  
b\_ent = Entry(root)  
ebs\_ent = Entry(root)  
  
*# похідна*def df(x):  
 return 3 \* x \*\* 2 - 1  
  
*# функція*def function(x):  
 return x\*\*3 - x + 1  
  
  
*# отримуємо а, б, похибку(значення)*def getDots():  
 try:  
 a = float(a\_ent.get())  
 b = float(b\_ent.get())  
 e = float(ebs\_ent.get())  
 x = np.linspace(a, b, 100)  
 y = list(map(function, x))  
 except:  
 messagebox.showinfo(**"Error"**, **"Введіть правильні значення!!!"**)  
  
 return a, b, e, x, y  
  
*# графік*def showGraph():  
 a, b, e, x, y = getDots()  
 if function(a) \* function(b) > 0:  
 messagebox.showinfo(**"Error"**, **"Помилкові а, b.**\n**Введіть інші значення!"**)  
 else:  
 x0, y0 = half\_divide\_method(a, b, e)  
 x\_0[**"text"**] = **"X0 = {}"**.format(x0)  
 f\_x0[**"text"**] = **"f(x0) = {}"**.format(y0)  
  
 plt.plot(x, y)  
 plt.plot(x0, y0, **'o'**)  
  
 plt.grid(True)  
 plt.show()  
  
  
*# пошук діапазонів коренів рівняння*def find\_ranges():  
 a = -100  
 b = 100  
 step = 1  
 n = (b-a)/step  
 b = a+step  
 kranges = []  
 for i in range(int(n)):  
 fa = function(a)  
 fb = function(b)  
 if fa\*fb < 0:  
 kranges.append([a, b])  
 a = b  
 b = b+step  
 return kranges  
  
*# метод половинного ділення(алгоритм)*def half\_divide\_method(a, b, ebs):  
 x = (a + b) / 2  
 while math.fabs(function(x)) >= ebs:  
 x = (a + b) / 2  
 a, b = (a, x) if function(a) \* function(x) < 0 else (x, b)  
 x = (a + b)/ 2  
 return (a + b) / 2, function(x)  
  
*# промжок кореня*ranges = find\_ranges()  
  
*# GUI*label\_1 = Label(root, text=**"Лабораторна робота № 4 з АМО** \n **Група : ІВ-92** \n **Бабенко Вікторія"**, font=(**'Times New Roman'**, 18))  
label\_1.grid(row=0, column=0)  
label\_1.config(bg=**"skyblue"**)  
  
label\_2 = Label(root, text=**"Варіант 1**\n**Метод половинного ділення"**, font=(**'Helvetica'**, 15))  
label\_2.config(bg=**"skyblue"**)  
label\_2.grid(row=1, column=0)  
  
task\_path = PhotoImage(file=**"var.png"**)  
task\_image = Label(root, image=task\_path)  
task\_image.grid(column=0, row=2)  
  
label\_3 = Label(root, text=**"Введіть значення меж а, b**\n **для відрізка [a, b]"**, font=(**'Helvetica'**, 15))  
label\_3.grid(row=3, column=0)  
label\_3.config(bg=**"orange"**)  
  
label\_4 = Label(root, text=**"Значення a"**, font=(**'Helvetica'**, 18))  
label\_5 = Label(root, text=**"Значення b"**, font=(**'Helvetica'**, 18))  
label\_6 = Label(root, text=**"Точність"**, font=(**'Helvetica'**, 18))  
label\_4.config(bg=**"yellow"**)  
label\_5.config(bg=**"yellow"**)  
label\_6.config(bg=**"yellow"**)  
  
label\_4.grid(column=0,row=4)  
label\_5.grid(column=0,row=5)  
label\_6.grid(column=0,row=6)  
  
a\_ent.grid(column=1, row=4)  
b\_ent.grid(column=1, row=5)  
ebs\_ent.grid(column=1, row=6)  
  
a\_ent.config(bg=**"skyblue"**)  
b\_ent.config(bg=**"skyblue"**)  
ebs\_ent.config(bg=**"skyblue"**)  
  
x\_0 = Label(root,text=**"X0 = "**, font=(**'Times New Roman'**, 18))  
f\_x0 = Label(root,text=**"f(x0) = "**, font=(**'Times New Roman'**, 18))  
  
x\_0.grid(column=0, row=9)  
f\_x0.grid(column=0, row=10)  
x\_0.config(bg=**"orange"**)  
f\_x0.config(bg=**"orange"**)  
  
solving\_range = Label(root, text=**"Рівняння має корені на проміжку {0}"**.format(ranges[0]), font=(**'Times New Roman'**, 14))  
solving\_range.config(bg=**"yellow"**)  
solving\_range.grid(column=0, row=8)  
  
graphic = Button(root, text=**"Значення** \n**Графік"**, font=(**'Times New Roman'**, 18), command=showGraph)  
graphic.config(bg=**"aqua"**)  
graphic.grid(column=0, row=15)  
root.mainloop()  
sys.exit()

**Аналіз результатів**

Реалізація алгоритму методу половинного ділення є описаною аналітично у наведеній блок-схемі, а також текстово. Програмна реалізація методу половинного ділення не є важкою, проте, цей алгоритм є потужним для рішення нелінійних рівнянь. Програмно реалізовано пошук діапазону, на якому існує корінь рівняння, сам корінь та значення функції в точці кореня. Побудовано графіки функції з дійсним коренем, це показано на скріншоті.

**Висновки**

В ході виконання цієї роботи було закрілено навички програмного розв’язання нелінійних рівнянь. Програмно був реалізований метод половинного ділення. Окрім того були закріплені основні навички роботи з графічними бібліотеками matplotlib, tkinter, numpy. Отримані результати виконання програми є правильними, це перевірено на тестовому прикладі. Кінцевої мети досягнуто.