

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.

## Хід роботи

**Мета:** Метою даного заняття є ознайомлення з методиками та вивчення різних алгоритмів розв'язання нелінійних рівнянь на комп'ютері.

### Завдання:

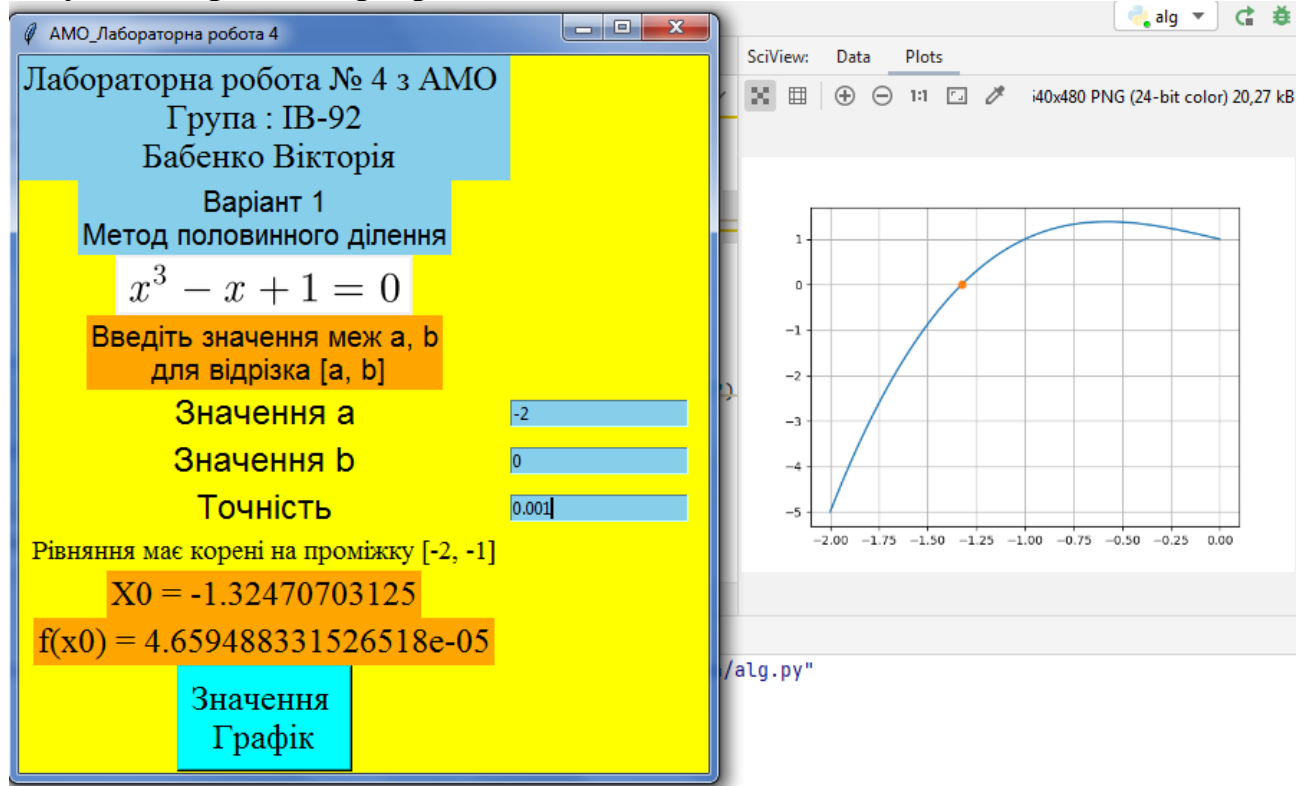
Закріплення знань студентів при вирішенні практичних завдань з розв'язування нелінійних рівнянь. Оволодіння методами і практичними навичками розв'язування нелінійних рівнянь на комп'ютері. Набуття умінь і навичок при програмуванні та налагодженні програм для розв'язування нелінійних рівнянь на комп'ютері.

### Варіанти завдання

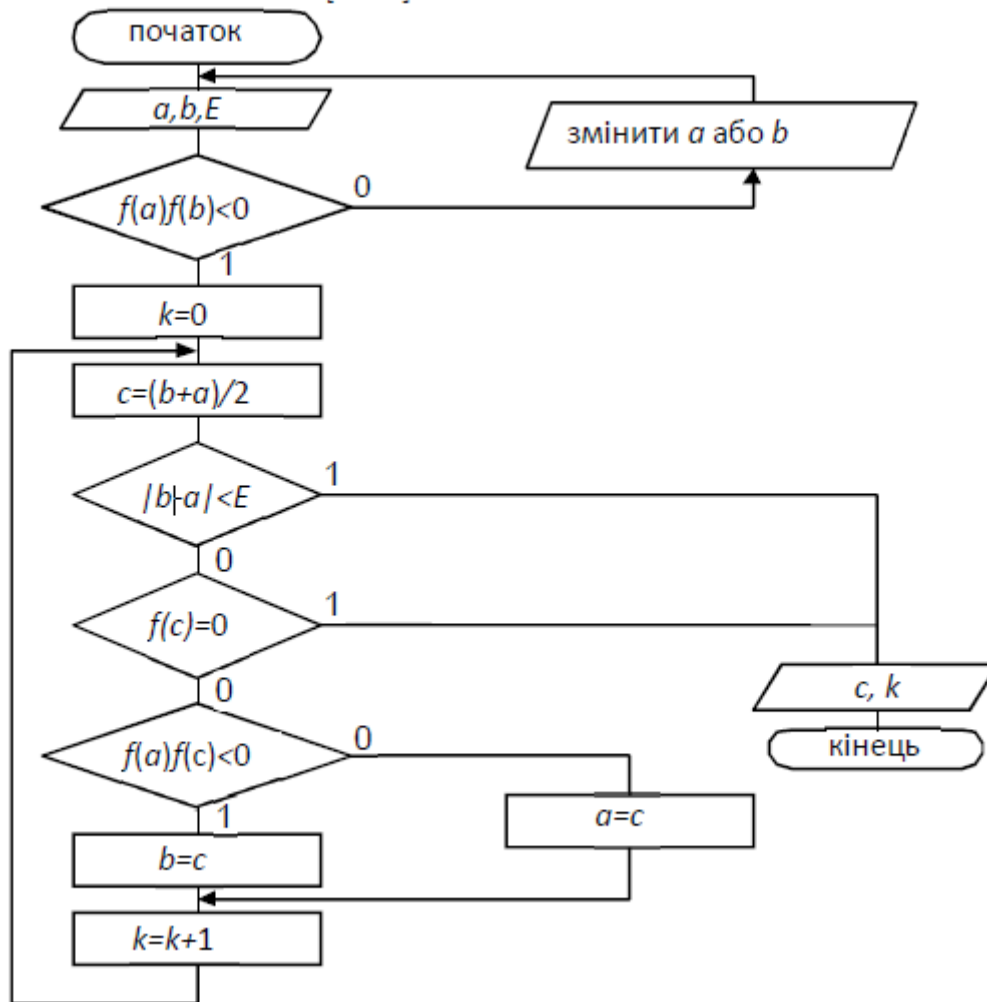
Варіант 1 (Метод половинного ділення)

1	$x^3 - x + 1 = 0$	-1.325
---	-------------------	--------

### 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 ( $a, b, E$ ). У розгалуженому циклі виконується перевірка на від'ємний знак добутку функції у кінцях відрізка  $ab$ , якщо умова не виконується, то потрібно змінити значення  $a, b$ . Оголошуємо лічильник  $k=0$  та значення  $c(1)$ , яке чисельно рівне півдобутку суми  $a$  та  $b$ . Виконуємо перевірку: якщо абсолютна величина різниці меж відрізка  $ab$  менша значення похибки, то виводимо значення лічильника та  $c$ , у протилежному випадку порівнюємо значення функції в точці  $c$  до нуля, якщо справджується, то виводимо значення  $c$  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 a, b, 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 a, b.\nВведіть 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\nМетод 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 a, 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 \nГрафік", 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 наведений 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.