НАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ УНІВЕРСИТЕТ УКРАЇНИ

"КИЇВСЬКИЙ ПОЛІТЕХНІЧНИЙ ІНСТИТУТ імені ІГОРЯ СІКОРСЬКОГО”

ПОЯСНЮВАЛЬНА ЗАПИСКА

ДО КУРСОВОГО ПРОЕКТУ

з дисципліни:

«Алгоритми і системи комп’ютерної математики.

Математичні алгоритми»

На тему:

«Розв’язування інтегралів за допомогою методу лівих прямокутників»

Виконав Артеменко Я. К.

Група КМ-63 ФПМ

N залікової книжки КМ-6301

Керівник\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ (Бай Ю. П.)

"\_\_\_" \_\_\_\_\_\_\_\_\_ 2020р.

Захищено з оцінкою\_\_\_\_\_\_\_\_\_\_\_

КИЇВ

2020

**ЗМІСТ**

[**ВСТУП** 2](#_Toc36631893)

[**1 ПОСТАНОВКА ЗАДАЧІ** 3](#_Toc36631894)

[**2 ПРОЕКТУВАННЯ МАТЕМАТИЧНОГО ЗАБЕЗПЕЧЕННЯ** 4](#_Toc36631895)

[**2.1 Описання тематики** 4](#_Toc36631896)

[**2.2 Математичний метод розв’язання задачі** 5](#_Toc36631897)

[**2.3 Опис користувацького інтерфейсу** 6](#_Toc36631898)

[**3 ПРОГРАМНА РЕАЛІЗАЦІЯ МАТЕМАТИЧНОЇ ЗАДАЧІ** 7](#_Toc36631899)

[**3.1 Опис розроблених алгоритмів** 7](#_Toc36631900)

[**3.2 Мова керування програмами** 8](#_Toc36631901)

[**3.3 Керівництво користувача** 9](#_Toc36631902)

[**3.4 Опис експериментальних даних** 10](#_Toc36631903)

[**3.5 Опис контрольних прикладів** 14](#_Toc36631904)

[**ВИСНОВКИ** 18](#_Toc36631905)

[**ЛІТЕРАТУРА** 19](#_Toc36631906)

[**ДОДАТКИ** 20](#_Toc36631907)

[**Додаток А (код програми)** 20](#_Toc36631908)

[**Додаток Б (скріншоти виконання)** 24](#_Toc36631909)

[**Додаток В (скрішоти меню)** 26](#_Toc36631910)

# **ВСТУП**

Даний курсовий проект присвячений для того, щоб допомагати студентам перевіряти свої рішення за допомогою програмних засобів. Програмні засоби будуть створюватись для розв’язування інтегралів за допомогою методів лівих прямокутників.

Для виконання цієї роботи буде використовуватись мова Python, оскільки дану мову зручно використовувати для рішення математичних задач.

Необхідність даної задачі полягає в тому, що в багатьох задачах, що пов’язані з аналізом, ідентифікацією, оцінкою якості різних засобів та систем автоматики та управління, виникає необхідність обчислення певних інтегралів. Даний метод є актуальним і часто використовується для розв’язання визначених інтегралів. Актуальність тестування даного методу полягає в тому, що цей метод потребує послідовних обчислень, в яких може виникати помилка. Тому при розв’язанні задач даним метод, потрібно завжди перевіряти його на правильність виконання. Задача чисельного інтегрування полягає в знаходженні приблизного значення інтегралу.

# **1 ПОСТАНОВКА ЗАДАЧІ**

Даний курсовий проект присвячений розробці програмного забезпечення для розв’язування інтегралів за допомогою методу лівих прямокутників.

Потрібно обчислити наближено інтеграл

із заданою точністю Е.

Перелік основних функцій:

F(x) – задається підінтегральна функція, для якої буде рахуватись інтеграл.

Work(f, a, b, n) – функція, в якій прописаний алгоритм, по якому буде рахуватись інтеграл.

Main(f, a, b) – головна функція, яка відповідає за підключення інших функцій і яка виводить результат обчислень.

submitForm() – функція, яка викликається після натискання на клавішу.

Дана програма повинна рахувати інтеграл від функції *f(x)* з вказаними межами інтегрування, що задаються користувачем та виводити результат обчислення з точністю, яка також вказується користувачем.

# **2 ПРОЕКТУВАННЯ МАТЕМАТИЧНОГО ЗАБЕЗПЕЧЕННЯ**

# **2.1 Описання тематики**

Чисельне інтегрування – обчислення значення певного інтеграла (як правило, наближене). Під чисельним інтеграцією розуміють набір чисельних методів для знаходження значення певного інтеграла.

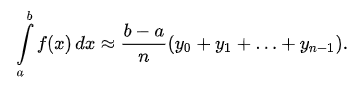
Чисельне інтегрування застосовується, коли:

* Сама підінтегральна функція не задана аналітично. Наприклад, вона представлена ​​у вигляді таблиці (масиву) значень у вузлах деякої розрахункової сітки.
* Аналітичне подання підінтегральної функції відомо, але її первісна не виражає через аналітичні функції.

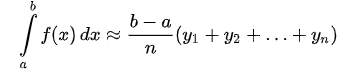
У цих двох випадках неможливо обчислення інтеграла за формулою Ньютона - Лейбніца. Також можлива ситуація, коли вид первісної настільки складний, що швидше обчислити значення інтеграла чисельним методом.

# **2.2 Математичний метод розв’язання задачі**

Нехай потрібно визначити значення інтегралу функції на відрізку [a, b]. Цей відрізок ділиться точками x0, x1, … , xn-1, xn на n рівних відрізках довжиною . Позначимо через y0, y1, … , yn-1 . Кожна з сум – інтегральна сума для f(x) на [a, b] і тому наближено виражає інтеграл.

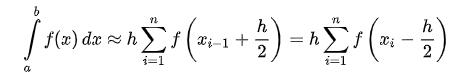


Якщо задана функція - позитивна і зростаюча, то ця формула виражає площу ступінчастою фігури, складеної з «вхідних» прямокутників, також звана формулою лівих прямокутників, а формула



висловлює площа ступінчастою фігури, що складається з «виходять» прямокутників, також звана формулою правих прямокутників. Чим менше довжина відрізків, на які ділиться відрізок [a, b], тим точніше значення, що обчислюється за цією формулою, шуканого інтеграла.

Очевидно, варто розраховувати на велику точність якщо брати, в якості опорної точки для знаходження висоти, точку посередині проміжку. В результаті отримуємо формулу середніх прямокутників:

,

де .

З огляду на апріорно більшу точність останньої формули при тому ж обсязі і характері обчислень її називають формулою прямокутників.

# **2.3 Опис користувацького інтерфейсу**

Після запуску програми користувач матиме можливість вводити функцію, яка буде інтегруватись, межі інтегрування та точність, з якою буде обчислюватись інтеграл. Для цього буде виділено спеціальні поля, де користувача буде запрошено ввести потрібні дані. Дані маніпуляції користувач виконує в екранній формі, яка з’являється після запуску програми.

# **3 ПРОГРАМНА РЕАЛІЗАЦІЯ МАТЕМАТИЧНОЇ ЗАДАЧІ**

# **3.1 Опис розроблених алгоритмів**

Для розв’язання був розроблений алгоритм на мові Python, який дозволяє рахувати значення інтегралу по заданим межам інтегрування та точності і після підрахунку виводити результат, який був отриманий в ході рішення.

Безпосередньо були створені наступні функції:

Work(f, a, b, n) – функція, в якій прописаний алгоритм методу лівих прямокутників, по якому буде рахуватись інтеграл.

Main(f, a, b) – головна функція, яка відповідає за підключення інших функцій і яка виводить результат обчислень.

submitForm() – функція, яка викликається після натискання на клавішу на екранній формі користувача.

# **3.2 Мова керування програмами**

Блок схема роботи даної програми виглядає наступним чином:

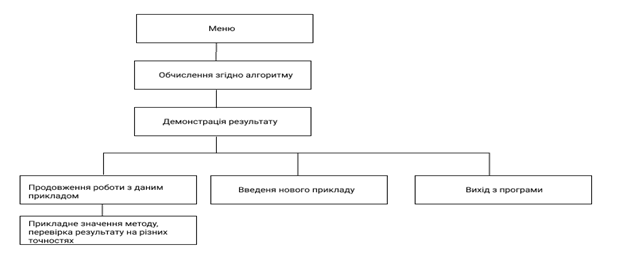


Рисунок 3.1 – Блок-схема

При запуску програми з’явиться вікно:

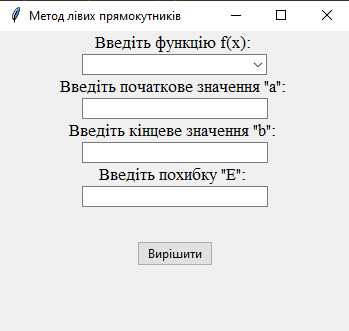


Рисунок 3.2 – Екранна форма

За допомогою даної форми у користувача буде можливість ввести функцію, яка буде підлягати інтегруванню, точність обчислень та межі інтегрування. Після натискання на кнопку «Вирішити» програма почне підрахунок і виведе остаточний результат.

# **3.3 Керівництво користувача**

Для введення певної функції потрібно використовувати формати, які використовуються в мові Python. Наприклад:

f(x) = exp(x), f(x) = x\*\*2, f(x) = tan(x)

Функція на відрізку [a; b] повинна бути неперервною.

Для вказання верхньої та нижньої межі інтегрування використовується тип float.

Формат вхідних даних для вказання точності:

1. Точність має бути менше 1.0 та більше ніж 0.000001;
2. Е = 0,0001, Е = 1е-5, Е = 0,000001.

Перелік повідомлень:

1. ValueError: could not convert string to float вказує на те, що при введені даних було допущено помилку, а саме було введено букву замість числа.
2. ValueError: invalid literal for int() with base 10 вказує на те, що замість цілого числа було введено дійсне, що викликало помилку.

# **3.4 Опис експериментальних даних**

1. Коректні вхідні дані:

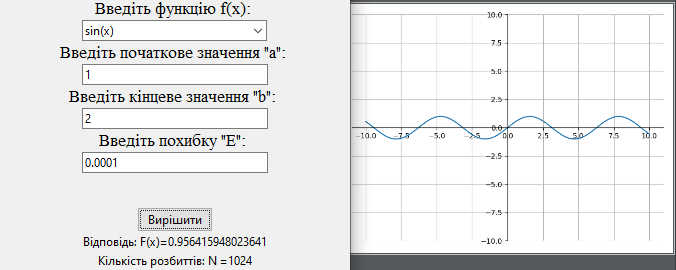


Рисунок 3.3 – Результат виконання для перших експериментальних даних

Рисунок 3.4 - Результат виконання для других експериментальних даних

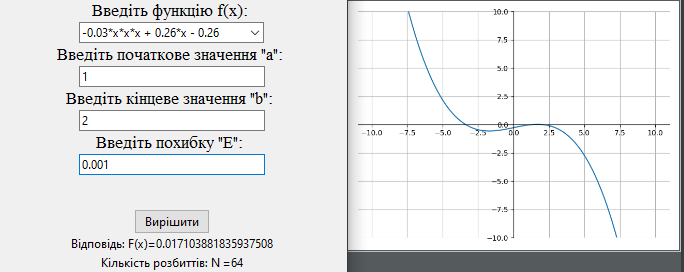


Рисунок 3.5 - Результат виконання для третіх експериментальних даних

1. Некоректні вхідні дані:

* Некоректно введена функція:

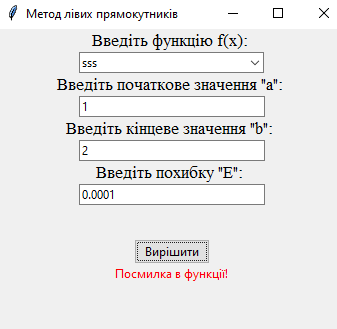


Рисунок 3.6 - Результат виконання для перших некоректних експериментальних даних

* Некоректно введено верхню межу b:

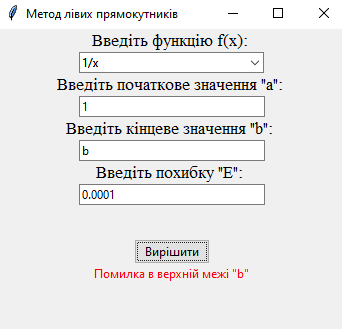


Рисунок 3.7 - Результат виконання для других некоректних експериментальних даних

* Некоректно введено нижню межу a:

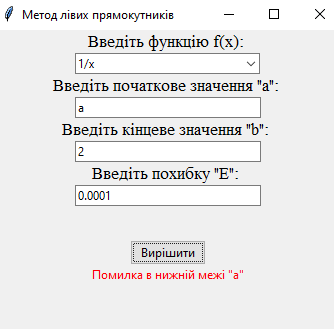


Рисунок 3.8 - Результат виконання для третіх некоректних експериментальних даних

* Некоректно введено значення похибки Е:

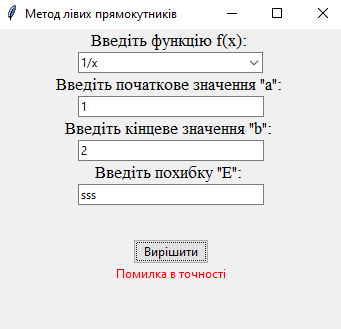


Рисунок 3.9 - Результат виконання для четвертих некоректних експериментальних даних

* Некоректно введено значення похибки Е:

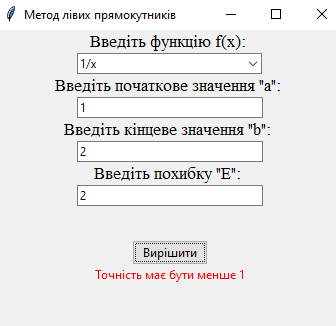


Рисунок 3.9 - Результат виконання для п’ятих некоректних експериментальних даних

# **3.5 Опис контрольних прикладів**

Приклад 1.

Функція , а = 1, b = 9, Е = 0.0001

*.*

|  |  |  |
| --- | --- | --- |
| i | xi | yi |
| 0 | 1 | 1 |
| 1 | 1.8 | 0.5556 |
| 2 | 2.6 | 0.3846 |
| 3 | 3.4 | 0.2941 |
| 4 | 4.2 | 0.2381 |
| 5 | 5 | 0.2 |
| 6 | 5.8 | 0.1724 |
| 7 | 6.6 | 0.1515 |
| 8 | 7.4 | 0.1351 |
| 9 | 8.2 | 0.122 |
| 10 | 9 | 0.1111 |

Приклад 2.

Функція при а = 1, b = 2, E=0.0001

|  |  |  |
| --- | --- | --- |
| i | xi | yi |
| 0 | 1 | 2.7183 |
| 1 | 1.1 | 2.0513 |
| 2 | 1.2 | 1.5979 |
| 3 | 1.3 | 1.277 |
| 4 | 1.4 | 1.0422 |
| 5 | 1.5 | 0.8657 |
| 6 | 1.6 | 0.7298 |
| 7 | 1.7 | 0.6231 |
| 8 | 1.8 | 0.5379 |
| 9 | 1.9 | 0.4689 |
| 10 | 2 | 0.4122 |

=

Розв’язки були оформлені за допомогою онлайн сервісу <https://math.semestr.ru/optim/rectangle.php>

Виконання за допомогою власноруч написаної програми:

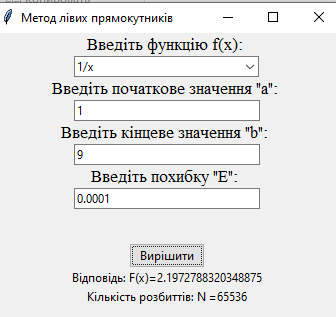


Рисунок 3.10 – Рішення для першого прикладу функції

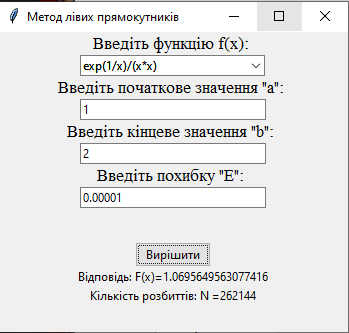


Рисунок 3.11 – Рішення для другого прикладу функції

Приклад 3.

Функція f(x) = x\*\*2, a=0, b=1, E = 0.0001

Очевидно, що даний інтеграл дорівнює 0.333333. Цей результат підтверджується розробленою програмою.

Виконання за допомогою власноруч написаної програми:

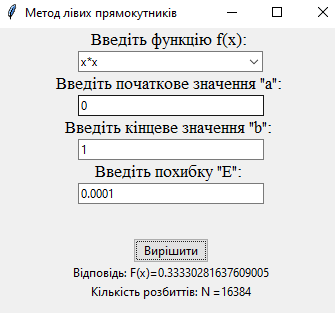


Рисунок 3.12 – Рішення для другого прикладу функції

# **ВИСНОВКИ**

В даному курсовому проекті було розроблено програмне забезпечення для розв’язування інтегралів за допомогою методу лівих прямокутників. Програму було написано на мові Python. При виконанні даної програми, користувач зможе отримати розв’язок поставленої задачі у вигляді чисельного значення та кількості інтервалів, на які було поділено даний відрізок з межами, який користувач вказує самостійно в ході виконання програми. Також, для того, щоб досягти більш високої точності виконання, користувачу було надано можливість самостійно ввести точність, яка буде використовуватись при виконанні програми.

Можна впевнено сказати, що дане програмне забезпечення являється потрібним, оскільки на даний момент для досягнення більш високих результатів при розв’язуванні інтегралів, потрібно мати програмні засоби, з допомогою яких буде можливість перевірити свій результат на правильність.

Подальшим розширенням може бути додавання нових методів обчислення інтегралів, між якими користувач зможе обрати потрібний, та забезпечення дружелюбним інтерфейсом, з яким будь-якому користувачу буде зручно працювати, а головне те, що він буде простим у використанні.

# **ЛІТЕРАТУРА**

1. [Самарский А. А.](https://www.wikiwand.com/ru/%D0%A1%D0%B0%D0%BC%D0%B0%D1%80%D1%81%D0%BA%D0%B8%D0%B9,_%D0%90%D0%BB%D0%B5%D0%BA%D1%81%D0%B0%D0%BD%D0%B4%D1%80_%D0%90%D0%BD%D0%B4%D1%80%D0%B5%D0%B5%D0%B2%D0%B8%D1%87), Гулин А. В. Численные методы: Учеб. пособие для вузов. — М.: Наука, 1989. — 432 с.
2. Каханер Д., Моулер К., Нэш С. Численные методы и программное обеспечение (пер. с англ.) — Изд. второе, стереотип — М.: Мир, 2001. — 575 с.
3. Чисельне інтегрування. – [Електронний ресурс] . — Режим доступу: <https://www.wikiwand.com/ru/%D0%A7%D0%B8%D1%81%D0%BB%D0%B5%D0%BD%D0%BD%D0%BE%D0%B5_%D0%B8%D0%BD%D1%82%D0%B5%D0%B3%D1%80%D0%B8%D1%80%D0%BE%D0%B2%D0%B0%D0%BD%D0%B8%D0%B5>, вільний.
4. Метод прямокутників [Електронний ресурс] /. — Режим доступу:   
   <http://www.mathprofi.ru/metod_prjamougolnikov.html>, вільний.
5. Метод прямокутників [Електронний ресурс] /. — Режим доступу: [http://www.cleverstudents.ru/integral/method\_of\_rectangles.html#samples](http://www.cleverstudents.ru/integral/method_of_rectangles.html%23samples), вільний.

# **ДОДАТКИ**

# **Додаток А (код програми)**

from math import \*  
import numpy as np  
import matplotlib.pyplot as plt  
from tkinter import \*  
from tkinter.ttk import \*  
  
  
  
funcText = ''  
  
  
functionLabelValue = None  
numberLabelValue = None  
errorLabel = None  
  
  
def f(x):  
 return eval(funcText)  
  
  
  
def work(f, a, b, n):  
 # print("\nТекущее число разбиений: ", n)  
 h = (b - a) / float(n)  
 # print("Текущий шаг:", h)  
 total = sum([f((a + (k \* h))) for k in range(0, n)])  
 result = h \* total  
 # print("Текущий результат: ", result)  
 return result  
  
  
  
def main(f, a, b, eps):  
 global functionLabelValue  
 global numberLabelValue  
 if (functionLabelValue):  
 functionLabelValue.destroy()  
 if (numberLabelValue):  
 numberLabelValue.destroy()  
 n = 2  
 a1 = work(f, 1, 2, n)  
 n \*= 2  
 a2 = work(f, 1, 2, n)  
  
 while abs(a1 - a2) > eps:  
 n \*= 2  
 a1 = work(f, a, b, n)  
 n \*= 2  
 a2 = work(f, a, b, n)  
  
  
 functionLabelValue = Label(text=f'Відповідь: F(x)={a2}')  
 functionLabelValue.pack()  
 numberLabelValue = Label(text=f'Кількість розбиттів: N ={n}')  
 numberLabelValue.pack()  
  
  
 f2 = np.vectorize(f)  
 x = np.linspace (-10, 10, 100)  
  
 plt.plot(x, f2(x))  
 plt.plot(a)  
 ax = plt.gca()  
 ax.spines['left'].set\_position('center')  
 ax.spines['bottom'].set\_position('center')  
 ax.spines['top'].set\_visible(False)  
 ax.spines['right'].set\_visible(False)  
  
 plt.grid(True)  
 plt.ylim(-10, 10)  
  
  
  
 plt.show()  
  
  
 return a2  
  
  
  
window = Tk()  
window.title('Метод лівих прямокутників')  
window.geometry("350x300")  
  
  
functionLabel = Label(text='Введіть функцію f(x): ',  
 font = ("Times New Roman", 13))  
functionLabel.pack()  
  
combo = Combobox(window, width = 27)  
combo['values'] = ('exp(1/x)/(x\*x)', '1/x')  
combo.pack()  
  
  
startPointLabel = Label(text='Введіть початкове значення "а": ',  
 font = ("Times New Roman", 13))  
startPointValue = Entry(window, width = 30)  
  
startPointLabel.pack()  
startPointValue.pack()  
  
endPointLabel = Label(text='Введіть кінцеве значення "b": ',  
 font = ("Times New Roman", 13))  
endPointValue = Entry(window, width = 30)  
  
endPointLabel.pack()  
endPointValue.pack()  
  
epsLabel = Label(text='Введіть похибку "E": ',  
 font = ("Times New Roman", 13))  
epsValue = Entry(window, width = 30)  
  
epsLabel.pack()  
epsValue.pack()  
  
label\_1 = Label(text='\n')  
label\_1.pack()  
  
  
def submitForm():  
 global errorLabel  
 if(errorLabel):  
 errorLabel.destroy()  
 if (functionLabelValue):  
 functionLabelValue.destroy()  
 if (numberLabelValue):  
 numberLabelValue.destroy()  
 try:  
  
 global funcText  
 global a  
 global b  
 global eps  
 funcText = combo.get()  
 print (funcText)  
  
 try:  
 a = float(startPointValue.get())  
 print (a)  
 except:  
 errorLabel = Label(text='Помилка в нижній межі "а"', foreground = "#f00")  
 errorLabel.pack()  
 return None  
  
 try:  
 b = float(endPointValue.get())  
 print (b)  
 except:  
 errorLabel = Label(text = 'Помилка в верхній межі "b"', foreground = "#f00")  
 errorLabel.pack()  
 return None  
  
 try:  
 eps = float(epsValue.get())  
 print(eps)  
 except:  
 errorLabel = Label(text = 'Помилка в точності', foreground = "#f00")  
 errorLabel.pack()  
 return None  
  
 if (eps >= 1):  
 errorLabel = Label(text='Точність має бути менше 1', foreground="#f00")  
 errorLabel.pack()  
 return None  
  
 if (a >= b):  
 errorLabel = Label(text='"а" повинно бути менше за "b"', foreground="#f00")  
 errorLabel.pack()  
 return None  
  
 global f  
 main(f, a, b, eps)  
 except :#Exception as e:  
 # print(traceback.format\_exc())  
 errorLabel = Label(text='Помилка в функції!', foreground = "#f00")  
 errorLabel.pack()  
  
  
submitButton = Button(window, text='Вирішити',  
 command=submitForm)  
submitButton.pack()  
  
  
window.mainloop()

# **Додаток Б (скріншоти виконання)**

Для функції

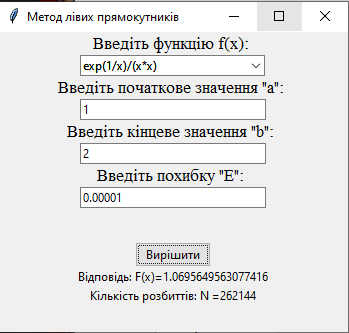


Рисунок 1. Рішення для першої функції

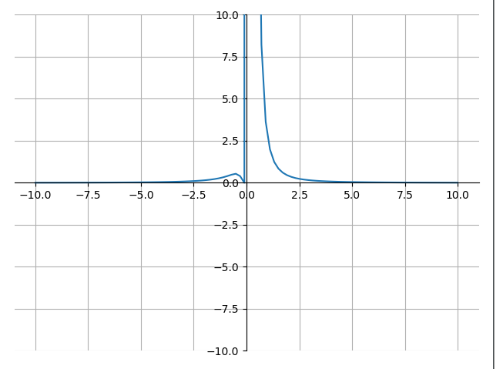


Рисунок 2 - Графік функції

Для функції

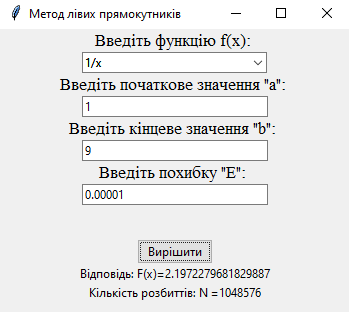


Рисунок 3. Рішення для другої функції

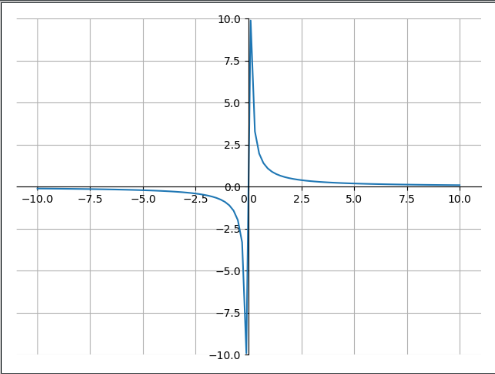


Рисунок 4 – Графік функції

# **Додаток В (скрішоти меню)**

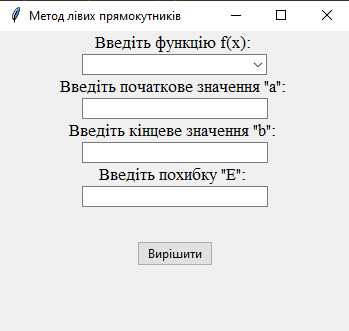


Рисунок 5. Початковий вигляд екранної форми

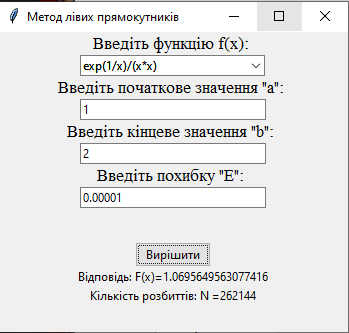


Рисунок 6. Після обчислення