НАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ УНІВЕРСИТЕТ УКРАЇНИ

"КИЇВСЬКИЙ ПОЛІТЕХНІЧНИЙ ІНСТИТУТ імені ІГОРЯ СІКОРСЬКОГО”

РОЗРАХУНКОВО-ГРАФІЧНА РОБОТА

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\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ (Олефір О. С.)

"\_\_\_" \_\_\_\_\_\_\_\_\_ 2019р.

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

КИЇВ

2020

# **ЗМІСТ**

[**ВСТУП** 2](#_Toc25604891)

[**1 ПОСТАНОВКА ЗАДАЧІ** 3](#_Toc25604892)

[**2 ОГЛЯД ЛІТЕРАТУРИ** 4](#_Toc25604893)

[**3 ПРОЕКТУВАННЯ МАТЕМАТИЧНОГО ЗАБЕЗПЕЧЕННЯ** 5](#_Toc25604894)

[**3.1 Описання тематики** 5](#_Toc25604895)

[**3.2 Математичний метод розв’язання задачі** 6](#_Toc25604896)

[**3.3 Опис користувацького інтерфейсу** 7](#_Toc25604897)

[**4 ПРОГРАМНА РЕАЛІЗАЦІЯ МАТЕМАТИЧНОЇ ЗАДАЧІ** 8](#_Toc25604898)

[**4.1 Опис розроблених алгоритмів** 8](#_Toc25604899)

[**4.2 Мова керування програмами** 8](#_Toc25604900)

[**4.3 Керівництво користувача** 9](#_Toc25604901)

[**4.4 Опис контрольних прикладів** 10](#_Toc25604902)

[**ВИСНОВКИ** 12](#_Toc25604903)

[**ЛІТЕРАТУРА** 13](#_Toc25604904)

[**ДОДАТКИ** 14](#_Toc25604905)

[**Додаток А (код програми)** 14](#_Toc25604906)

[**Додаток Б (скріншоти виконання)** 16](#_Toc25604907)

[**Додаток В (скрішоти меню)** 19](#_Toc25604908)

# **ВСТУП**

Дана розрахунково-графічна робота присвячена 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**

Дана розрахункова робота присвячена розробці програмного забезпечення для розв’язування інтегралів за допомогою методу лівих прямокутників.

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 з вказаними межами інтегрування, що задаються користувачем та виводити результат обчислення з точністю, яка також вказується користувачем.

# **2 ОГЛЯД ЛІТЕРАТУРИ**

1. <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>
2. <http://www.mathprofi.ru/metod_prjamougolnikov.html>
3. <https://ru.wikipedia.org/wiki/%D0%9C%D0%B5%D1%82%D0%BE%D0%B4_%D0%BF%D1%80%D1%8F%D0%BC%D0%BE%D1%83%D0%B3%D0%BE%D0%BB%D1%8C%D0%BD%D0%B8%D0%BA%D0%BE%D0%B2>
4. <https://studfile.net/preview/5581807/page:7/>

Поставлену задачу буде розв’язано методом лівих прямокутників, який описується за даними посиланнями.

# **3 ПРОЕКТУВАННЯ МАТЕМАТИЧНОГО ЗАБЕЗПЕЧЕННЯ**

# **3.1 Описання тематики**

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

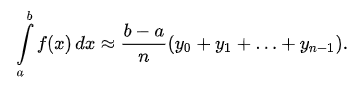
Чисельне інтегрування застосовується, коли:

* 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.

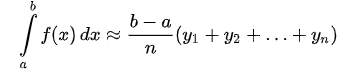
У цих двох випадках неможливо обчислення інтеграла за формулою Ньютона - Лейбніца. Також можлива ситуація, коли вид первісної настільки складний, що швидше обчислити значення інтеграла чисельним методом.

# **3.2 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.

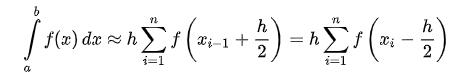


Якщо задана функція - позитивна і зростаюча, то ця формула виражає площу ступінчастою фігури, складеної з «вхідних» прямокутників, також звана формулою лівих прямокутників, а формула



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.

# **3.3 Опис користувацького інтерфейсу**

Після запуску програми користувач матиме можливість вводити межі інтегрування та точність, з якою буде обчислюватись інтеграл. Для цього буде виділено спеціальні поля, де користувача буде запрошено ввести потрібні дані. Це користувач може зробити як і в консолі, так і в екранних формах.

# **4 ПРОГРАМНА РЕАЛІЗАЦІЯ МАТЕМАТИЧНОЇ ЗАДАЧІ**

# **4.1 Опис розроблених алгоритмів**

Для розв’язання був розроблений створений 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.

# **4.2 Мова керування програмами**

Блок схема роботи даної програми виглядає наступним образом:

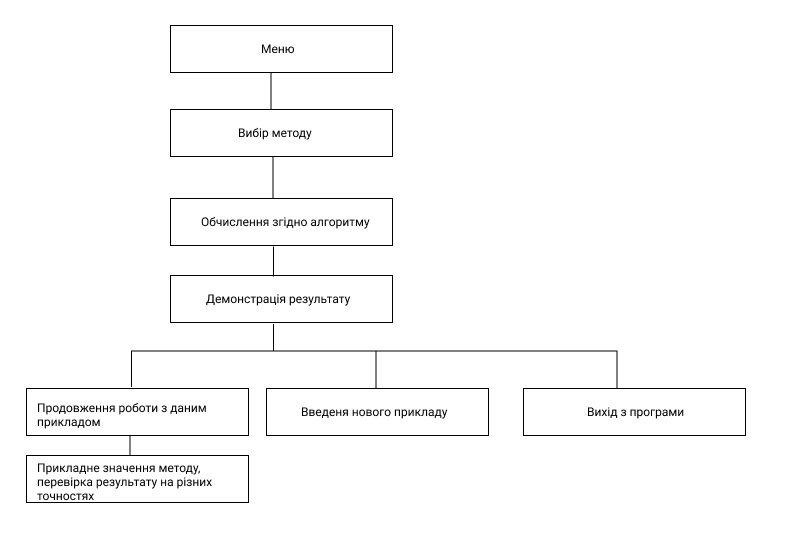


Рисунок 4.1 – Блок-схема

При запуску програми з’явиться вікно, яке буде мати назву, відповідну до теми курсової роботи.

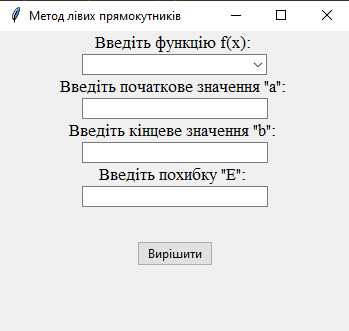


Рисунок 4.2 – Екранна форма

За допомогою даної форми у користувача буде можливість ввести функцію, яка буде підлягати інтегруванню, точність обчислень та межі інтегрування, після чого програма почне підрахунок і виведе остаточний результат.

# **4.3 Керівництво користувача**

Формат вхідних даних для вказання точності повинен бути <1 і типом точності є тип float.

Для вказання верхньої та нижньої межі використовується тип float.

Для введення певної функції потрібно використовувати спеціальні символи, щоб програма могла її інтегрувати.

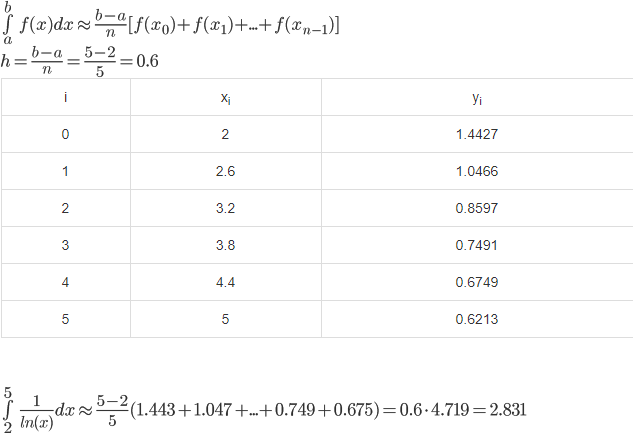
Перелік повідомлень:

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.

# **4.4 Опис контрольних прикладів**

Приклад 1.

Функція 1/ln(x), де а = 2, b = 5



Приклад 2.

Функція при а = 1, b = 2

|  |  |  |
| --- | --- | --- |
| 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>

# **ВИСНОВКИ**

В даній розрахунково графічній роботі 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 остаточне рішення 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 своє рішення 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. <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>
2. <http://www.mathprofi.ru/metod_prjamougolnikov.html>
3. <https://ru.wikipedia.org/wiki/%D0%9C%D0%B5%D1%82%D0%BE%D0%B4_%D0%BF%D1%80%D1%8F%D0%BC%D0%BE%D1%83%D0%B3%D0%BE%D0%BB%D1%8C%D0%BD%D0%B8%D0%BA%D0%BE%D0%B2>
4. <https://studfile.net/preview/5581807/page:7/>
5. <http://www.cleverstudents.ru/integral/method_of_rectangles.html#samples>

# **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 = ''  
  
mylist = None  
functionLabelValue = None  
numberLabelValue = None  
errorLabel = None  
  
#e^(1/x)/x^2  
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  
 funcText = combo.get()  
 print (funcText)  
 global a  
 a = float(startPointValue.get())  
 print (a)  
 global b  
 b = float(endPointValue.get())  
 print (b)  
 global eps  
 eps = float(epsValue.get())  
 print(eps)  
 global f  
 main(f, a, b, eps)  
 except Exception as e:  
 print(traceback.format\_exc())  
 errorLabel = Label(text='Помилка введення!', foreground = "#f00")  
 errorLabel.pack()  
  
submitButton = Button(window, text='Вирішити',  
 command=submitForm)  
submitButton.pack()  
  
  
window.mainloop()

# **Додаток Б (скріншоти виконання)**

Для функції

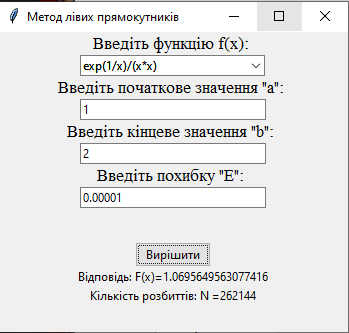


Рисунок 1. Рішення для першої функції

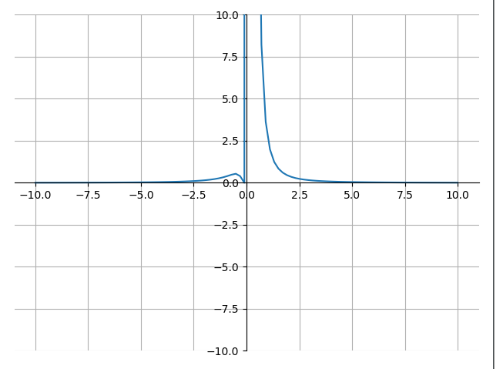


Рисунок 2 - Графік функції

Для функції

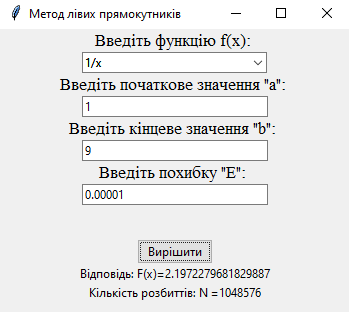


Рисунок 3. Рішення для другої функції

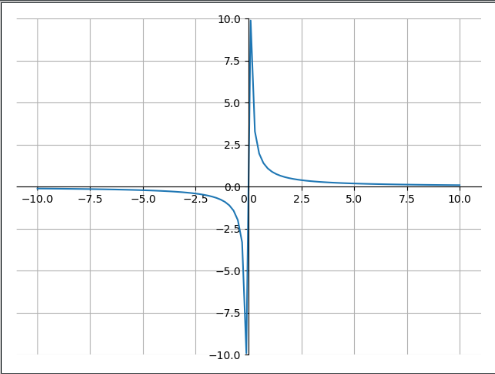


Рисунок 4 – Графік функції

# **Додаток В (скрішоти меню)**

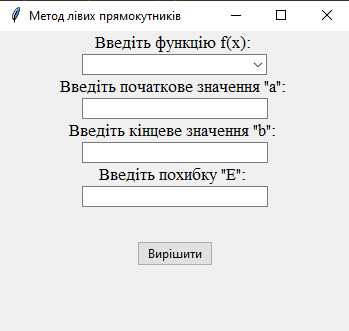


Рисунок 5. Початковий вигляд екранної форми

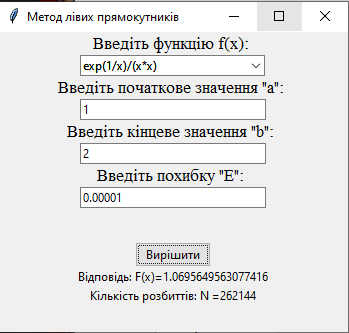


Рисунок 6. Після обчислення