

ВИЩИЙ НАВЧАЛЬНИЙ ЗАКЛАД
«УНІВЕРСИТЕТ ЕКОНОМІКИ ТА ПРАВА «КРОК»

КВАЛІФІКАЦІЙНА РОБОТА

Тема: «Калькулятор з функціями інтегрування та диференціювання»

Ступінь вищої освіти – бакалавр
Спеціальність – 122 «Комп’ютерні науки»
Освітня програма «Комп’ютерні науки»

ПОЯСНЮВАЛЬНА ЗАПИСКА

Виконав: здобувач 4 курсу
групи КН-21

Іван РАДЗИХОВСЬКИЙ

Керівник: викладач кафедри інформаційного
менеджменту, математики та
статистики

Олег МУШИНСЬКИЙ

Засвідчую, що кваліфікаційна
робота оформлена відповідно
до ДСТУ 3008:2015 та не
містить запозичень з праць
інших авторів без відповідних
посилань.

Здобувач: _____
(підпис)

ВИЩИЙ НАВЧАЛЬНИЙ ЗАКЛАД
«УНІВЕРСИТЕТ ЕКОНОМІКИ ТА ПРАВА «КРОК»

ЗАТВЕРДЖУЮ:
завідувач кафедри
комп'ютерних наук
_Сергій МІЧКІВСЬКИЙ
«_____» _____ 20__ р

ЗАВДАННЯ
НА КВАЛІФІКАЦІЙНУ РОБОТУ
Радзиховський Іван Русланович

Тема роботи	Калькулятор з функціями інтегрування та диференціювання
Номер та дата наказу про затвердження теми	№121-7 від 24 грудня 2024 року
Коротка постановка завдання	Розробити калькулятор з функціями інтегрування та диференціювання для обчислення похідних та інтегралів.
Посилання на джерела інформації (не більше п'яти найменувань, які рекомендує науковий керівник)	1. Demana, F., & Waits, B. K. (2000). Calculators in mathematics teaching and learning. Past, present, and future. In Learning Mathematics for a New Century, 51-66. 2. Irvan, I. (2024). Application of Integrals in Calculating Ball Volume using GeoGebra. Indonesian Journal of Education and Mathematical Science, 5(1), 58-63.
Вимоги до кваліфікаційної роботи	Кваліфікаційна робота має передбачити теоретичне, системотехнічне або експериментальне дослідження складного спеціалізованого завдання або практичної проблеми в галузі комп'ютерних наук, яке характеризується комплексністю та невизначеністю умов і потребує застосування теорій і методів інформаційних технологій.

Дата видачі завдання 27 грудня 2024 р.

Керівник

Олег МУШИНСЬКИЙ

Здобувач освітнього ступеня бакалавра

Іван РАДЗИХОВСЬКИЙ

КАЛЕНДАРНИЙ ПЛАН

№	Назва етапів роботи	Термін виконання	Примітка
Підготовчий етап			
1	Вибір напрямку дослідження	02.12.2024 р.	виконано
2	Формування теми та призначення керівника	16.12.2024 р.	виконано
3	Затвердження теми кваліфікаційної роботи	23.12.2024 р.	виконано
4	Затвердження завдання на кваліфікаційну роботу	27.12.2024 р.	виконано
Основний етап			
5	Розробка концепції кваліфікаційної роботи	13.01.2025 р.	виконано
6	Підбір та вивчення джерел інформації з напрямку дослідження. Огляд існуючих аналогів	20.01.2025 р.	виконано
7	Затвердження розширеної постановки завдання. Підготовка та подання керівникові розділу 1 кваліфікаційної роботи	10.03.2025 р.	виконано
8	Проектування. Підготовка та подання керівникові розділу 2 кваліфікаційної роботи	24.03.2025 р.	виконано
9	Підготовка доповіді для експертизи стану виконання кваліфікаційної роботи (проміжний контроль)	31.03-04.04.2025 р.	виконано
10	Реалізація. Підготовка та подання керівникові розділу 3 кваліфікаційної роботи	07.04.2025 р.	виконано
11	Підготовка та подання керівнику першого варіанту всієї кваліфікаційної роботи	14.04.2025 р.	виконано
12	Доопрацювання кваліфікаційної роботи з урахуванням зауважень керівника та представлення керівникові доопрацьованого варіанту кваліфікаційної роботи	21.04.2025 р.	виконано
Завершальний етап			
13	Представлення рукопису для перевірки на плагіат	28.04-04.05.2025 р.	виконано
14	Підготовка презентації та доповіді на передзахист	05.05-11.05.2025 р.	виконано
15	Передзахист кваліфікаційної роботи	12.05-16.05.2025 р.	виконано
16	Доопрацювання роботи за результатами передзахисту	19.05-06.06.2025 р.	виконано
17	Експертиза роботи керівником та зовнішнім експертом	09.06-15.06.2025 р.	виконано
18	Доопрацювання доповіді та презентації для захисту	09.06-15.06.2025 р.	виконано
19	Захист кваліфікаційної роботи	16.06-22.06.2025 р.	виконано

Керівник

Олег МУШИНСЬКИЙ

Здобувач освітнього ступеня бакалавра

Іван РАДЗИХОВСЬКИЙ

Радзиховський І.Р. Калькулятор з функціями інтегрування та диференціювання

Кваліфікаційна робота присвячена розробці програмного калькулятора з підтримкою операцій інтегрування та диференціювання математичних функцій. У роботі буде проведено аналіз предметної області, розглянуто існуючі аналоги, обґрунтовано вибір алгоритмів і технологій для реалізації. Описано етапи проєктування, включаючи архітектуру програми, алгоритми обчислення та інтерфейс користувача.

Розроблений програмний продукт дозволяє автоматизувати процес розрахунку похідних і визначених інтегралів, що може бути корисним для студентів, викладачів і науковців у сфері математичних обчислень.

Ключові слова: калькулятор, інтегрування, диференціювання, алгоритми, програмування, математичні обчислення, автоматизація.

Radzikhovsky I.R. Calculator with integration and differentiation functions

The qualification work is devoted to the development of a software calculator with support for integration and differentiation of mathematical functions. The work analyzes the subject area, considers existing analogues, justifies the choice of algorithms and technologies for implementation. The design stages are described, including the program architecture, calculation algorithms, and user interface.

The developed software product allows you to automate the process of calculating derivatives and definite integrals, which can be useful for students, teachers and scientists in the field of mathematical calculations.

Keywords: calculator, integration, differentiation, algorithms, programming, mathematical calculations, automation.

ЗМІСТ

ВСТУП.....	3
РОЗДІЛ 1 АНАЛІЗ ПРЕДМЕТНОЇ ОБЛАСТІ	5
1.1 Опис предметної області.....	5
1.2 Огляд існуючих рішень для обчислення інтегралів та похідних.....	6
1.3 Постановка задачі.....	18
Висновок до розділу 1.....	20
РОЗДІЛ 2 ПРОЄКТУВАННЯ ТА РОЗРОБКА КАЛЬКУЛЯТОРА ДЛЯ ОБЧИСЛЕННЯ ІНТЕГРАЛІВ ТА ПОХІДНИХ	21
2.1 Опис проєкту створення калькулятора	21
2.2 Основні бібліотеки Python для реалізації обчислень	24
Висновок до розділу 2.....	27
РОЗДІЛ 3 ПРОГРАМНА РЕАЛІЗАЦІЯ КАЛЬКУЛЯТОРА ДЛЯ ОБЧИСЛЕННЯ ІНТЕГРАЛІВ ТА ПОХІДНИХ	28
3.1 Огляд застосованих бібліотек при реалізації калькулятора.....	28
3.2 Основні фрагменти коду реалізації функцій калькулятора для обчислення інтегралів та похідних	30
3.3 Результати роботи калькулятора по обчисленню інтегралів та похідних.....	39
Висновок до розділу 3.....	56
ВИСНОВОК	58
ПЕРЕЛІК ДЖЕРЕЛ ПОСИЛАННЯ	60

ВСТУП

Сучасний розвиток комп'ютерних технологій дозволив створити велику кількість прикладних програм, для розв'язку різноманітних математичних задач. В основі таких програм лежить знаходження точного та швидкого розв'язання задач, які зазвичай важко розв'язати аналітично за розумну кількість часу. Використання таких програм може допомогти при перевірці вже існуючих розв'язків, для перевірки існування розв'язків поставленої задачі, тощо.

Використання таких пакетів при розв'язуванні задач, пов'язаних з інтегральним та диференціальним численням є досить поширеним, так як дані розділи математичного аналізу представлені у великій кількості задач прикладного значення, які представлені не тільки в математиці.

Розробка калькулятора, який включав би в собі функції обчислення інтегралів та похідних, є одним із практичних рішень при знаходженні розв'язків різних задач, що зумовлює актуальність обраної теми.

Метою роботи є розробка калькулятора для розв'язування задач пов'язаних з інтегральним та диференціальним численням функцій однієї та декількох змінних.

Для досягнення поставленої мети, потрібно виконати наступні завдання:

- провести аналіз предметної області;
- розглянути особливості різних прикладних математичних пакетів при розв'язуванні задач, пов'язаних з інтегралами та похідними;
- провести постановку задачі розробки калькулятора для обчисленні інтегралів та похідних;
- надати обґрунтування вибору засобу реалізації калькулятора для обчислення інтегралів та похідних;
- представити основні функції, які буде виконувати розроблений програмний продукт;

- провести тестування розробленого калькулятора для обчислення інтегралів та похідних.

Об'єктом дослідження є методи комп'ютеризованого розв'язування задач інтегрального та диференціального числення.

Предметом дослідження є створення калькулятора для обчислення інтегралів та похідних.

Методами дослідження є аналіз, узагальнення, систематизація, класифікація, моделювання.

Практичне значення отриманих результатів полягає в отриманні вузько направленою програмного продукту для розв'язування задач, пов'язаних із обчисленням інтегралів та похідних.

Структура роботи. Кваліфікаційна робота складається зі вступу, трьох основних розділів роботи, висновків, переліку джерел посилання із 25 найменувань, 66 рисунків, одного додатку. Загальний обсяг роботи становить 63 сторінки, з них основного тексту 58 сторінок.

В першому розділі роботи проводиться аналіз предметної області, виконується огляд програмних прикладних пакетів для обчисленні інтегралів та похідних, виконується постановка задачі.

В другому розділі роботи проводиться обґрунтування вибору засобу створення калькулятора для обчислення інтегралів та похідних, наводиться функціонал калькулятору.

В третьому розділі проводиться огляд основних бібліотек, за допомогою яких реалізується калькулятор, наводяться фрагменти коду програми калькулятора, проводиться тестування розробленого програмного продукту.

РОЗДІЛ 1

АНАЛІЗ ПРЕДМЕТНОЇ ОБЛАСТІ

1.1 Опис предметної області

Інтегральне та диференціальне числення є досить добре вивченими розділами математичного аналізу, які мають велику кількість застосувань як в математиці, так і в інших науках, таких як фізика, хімія, біологія, медицина, географія і т.д.

Однак на практиці, виникають задачі, які досить важко розв'язати швидко за допомогою наявних аналітичних методів. Як правило, такі задачі виникають під час обчислення інтегралів. Така ситуація виникає тому, що далеко не всі функції мають відносно просту первісну, що значно ускладнює знаходження значень визначених інтегралів [1].

Тому, для вирішення даної задачі існують прикладні програмні пакети, які дозволяють набагато швидше знаходити первісні і обчислювати значення інтегралів. Також такі пакети підходять для обчислення похідних різного рівня складності для різних видів задання функції.

Функціонал сучасних програмних пакетів досить широкий і постійно оновлюється. Серед основних функцій, які виконують прикладні програмні пакети виділяють:

- спрощення виразів;
- дії над матрицями;
- розв'язування рівнянь, нерівностей та їх систем;
- операції з векторами;
- побудова кривих та поверхонь заданих в різних системах координат;
- знаходження границь, похідних, інтегралів;
- програмування алгоритмів розв'язку різних прикладних задач.

В залежності від спрямованості прикладного пакету, змінюється і його функціонал, але представлені функції є базовими для кожного з них.

Для знаходження похідних, прикладні програмні пакети мають набір команд, які виводять операцію диференціювання на екран, після введення функції, похідну якої потрібно знайти, кінцевий результат виводиться на екран. Також, можливо зробити перевірку знайденої похідної, виконавши операцію інтегрування. Як правило, можна знаходити похідні стандартних функцій, функцій заданих в полярній системі координат, функцій заданих параметрично, функцій заданих неявно, спеціальних функцій [19].

При обчисленні інтегралів, ситуація така ж сама як і для знаходження похідних, на екран можна вивести знак інтегрування, ввести потрібну функцію та знайти первісну або знайти значення визначеного інтегралу. Результат знаходження первісної можна перевірити за допомогою диференціювання. Також, якщо вид отриманої первісної є складним, можна провести операцію спрощення, що представить отриманий результат в більш компактній формі, що іноді буває досить зручно для швидких обчислень [6].

1.2 Огляд існуючих рішень для обчислення інтегралів та похідних

З розвитком обчислювальної математики нерозривно пов'язаний розвиток програмування, яке дозволяє спростити способи взаємодії людини і ЕОМ. За багато років накопичені великі бібліотеки наукових підпрограм на різних алгоритмічних мовах, призначених для розв'язання типових задач обчислювальної математики. Крім того, є цілий ряд різних математичних пакетів, що реалізують різноманітні чисельні методи і проводять аналітичні математичні перетворення.

Найбільш відомими сьогодні є пакети прикладних програм (ППП) і математичні бібліотеки: MatLab [25] (компанія The MathWorks), Maple [22] (компанія Waterloo Maple Inc), Mathematica [24] (компанія Wolfram Research), MathCAD [23] (компанія MathSoft Inc). Пакет Maple також популярний у наукових колах. Крім аналітичних перетворень, пакет вирішує завдання чисельно. Характерною особливістю пакету є те, що він дозволяє

конвертувати документи в формат LaTeX-стандартний формат переважної більшості наукових видавництв [16].

Крім того, ряд інших програмних продуктів використовують інтегрований символічний процесор Maple. Наприклад, пакет підготовки наукових публікацій Scientific WorkPlace (компані TCI Software Research) дозволяє звертатися до символічного процесора Maple, проводити аналітичні перетворення і вбудовувати отримані результати в документ.

Пакет MatLab фактично являє собою своєрідну мову програмування високого рівня, орієнтований на вирішення наукових завдань. Характерною особливістю пакету є те, що він дозволяє зберігати документи у форматі мови програмування C [25].

Пакет Mathematica є сьогодні найбільш популярним у наукових колах, особливо серед теоретиків. Пакет надає широкі можливості в проведенні символічних (аналітичних) перетворень, проте вимагає значних ресурсів комп'ютера. Система команд пакета багато в чому нагадує звичайну мову програмування [17].

Пакет MathCAD популярний, мабуть, більше в інженерному, ніж у науковому, середовищі. Характерною особливістю пакету є використання звичних стандартних математичних позначень, тобто документ на екрані виглядає точно так само, як звичайний математичний розрахунок. Для використання пакета не потрібно вивчати будь-яку систему команд, як, наприклад, у випадку пакетів Mathematica або Maple [23].

Пакет орієнтований в першу чергу на проведення чисельних розрахунків, але має вбудований символічний процесор Maple, що дозволяє виконувати аналітичні перетворення. В останніх версіях передбачена можливість створювати зв'язки документів MathCAD з документами MathLab. На відміну від згаданих вище пакетів, MathCAD є середовищем візуального програмування, тобто не вимагає знання специфічного набору команд. Останнім часом проглядається тенденція до зближення та інтеграції різних пакетів. Наприклад, останні випуски пакетів Mathematica і Maple мають

хороші можливості для візуального програмування; в MatLab включена бібліотека аналітичних перетворень Maple; MathCAD дозволяє працювати спільно з MatLab.

Пакет MathCAD найбільш відповідає вимогам обчислювальної середовища при вивченні обчислювальної математики: звичні стандартні математичні позначення, середовище візуального програмування, можливість символічних обчислень, дружній інтерфейс, відносна невибагливість до можливостей комп'ютера.

Для більше детального ознайомлення варто розглянути декілька прикладних пакетів в контексті знаходження інтегралів та похідних.

MatLab. Продукт компанії MathWorks, Inc., що є мовою високого рівня для науково-технічних обчислень. Серед основних областей його застосування — математичні розрахунки, розробка алгоритмів, моделювання, аналіз даних та візуалізація, наукова та інженерна графіка, розробка програм, включаючи графічний інтерфейс користувача [12]. MatLab вирішує безліч комп'ютерних завдань - від збору та аналізу даних до розробки готових додатків. Середовище поєднує в собі математичні обчислення, візуалізацію та потужну технічну мову. Вбудовані універсальні інтерфейси дозволяють легко працювати із зовнішніми інформаційними джерелами, а також здійснювати інтеграцію з процедурами, написаними мовами високого рівня (C, C++, Java та ін.). Мультиплатформність MatLab зробила його одним із найпоширеніших продуктів, також він має широкий спектр застосувань, у тому числі цифрову обробку сигналів та зображень, проектування систем управління, природничі науки, фінанси, економіку, приладобудування тощо [3].

У мові програмування Matlab існує безліч інструментів для роботи з диференціювання та інтегрування функцій. Ці операції є основними для аналізу математичних моделей та вирішення завдань, пов'язаних із чисельними методами. У Matlab можна використовувати як символічні обчислення, і чисельні методи [4].

Для виконання диференціювання в Matlab використовується символічний двигун, який надається пакетом Symbolic Math Toolbox. Для початку потрібно створити символічну змінну за допомогою команди `syms` (рис. 1.1).

```
syms x
f = x^3 + 5*x^2 - 2*x + 1;
df = diff(f, x);
disp(df);
```

Рисунок 1.1 – Створення символічної змінної в MatLab

Джерело: [4]

Цей код створює символічну змінну x , потім визначає функцію f , і здійснює її диференціювання по x з допомогою функції `diff`. Результат: похідна функція буде виведена як вираз.

Функція `diff` використовується для знаходження похідних:

- `diff(f, x)`- Похідна функції f по змінній x ;
- можна також обчислювати похідні вищих порядків, наприклад, `diff(f, 2)` друга похідна [4].

Приклад обчислення похідної (рис. 1.2)

```
syms x
f = x^4 - 3*x^2 + 2*x - 5;
df = diff(f, x);
d2f = diff(f, 2);
disp(df);
disp(d2f);
```

Результат:

```
df = 4*x^3 - 6*x + 2
d2f = 12*x^2 - 6
```

Рисунок 1.2 – Приклад обчислення похідної в MatLab

Джерело: [4]

В Matlab можна використовувати функцію `gradient`, яка обчислює наближені похідні на основі кінцевих різниць (рис. 1.3) [7].

```
x = linspace(0, 10, 100);
y = sin(x);
dy = gradient(y, x);
plot(x, dy);
```

Рисунок 1.3 – Приклад чисельного диференціювання в MatLab

Джерело: [4]

Для інтегрування Matlab також використовується пакет Symbolic Math Toolbox. Символічне інтегрування виконується через функцію `int`, яка обчислює невизначений чи визначений інтеграл [4].

Для початку потрібно створити символічну змінну за допомогою команди `syms` (рис. 1.4).

```
syms x
f = x^2 + 2*x + 1;
I = int(f, x);
disp(I);
```

Рисунок 1.4 – Приклад створення символічної змінної для інтегрування в MatLab

Джерело: [4]

Для обчислення визначеного інтеграла необхідно зазначити межі інтегрування. Наприклад, обчислити визначений інтеграл функції від 0 до 2 (рис. 1.5):

```
syms x
f = x^2 + 2*x + 1;
I_definite = int(f, x, 0, 2);
disp(I_definite);
```

Результат:

```
I_definite = 10
```

Рисунок 1.5 Обчислення визначеного інтеграла в MatLab

Джерело: [4]

У випадку, коли інтеграл неможливо висловити аналітично, або коли функція задана чисельно, Matlab пропонує чисельні методи інтегрування. Для цього використовується функція `integral`, яка застосовує чисельні методи обчислення інтегралів (рис. 1.6) [8].

```
f = @(x) x.^2 + 2.*x + 1;
I_num = integral(f, 0, 2);
disp(I_num);
```

Рисунок 1.6 – Приклад реалізації чисельного інтегрування в MatLab

Джерело: [4]

Компанії Wolfram Research, Inc. має надзвичайно широкий набір засобів, що перекладають складні математичні алгоритми програми [11]. По суті, всі алгоритми, що містяться в курсі вищої математики технічного вузу, закладені на згадку про комп'ютерну систему Mathematica. Величезна перевага системи Mathematica полягає в тому, що її оператори та способи запису алгоритмів прості та природні. Mathematica має потужний графічний пакет, за допомогою якого можна будувати графіки дуже складних функцій однієї та двох змінних. Головна перевага Mathematica, що робить її безперечним лідером серед інших

систем високого рівня, полягає в тому, що ця система набула сьогодні дуже широкого поширення у всьому світі, охопивши величезні сфери застосування в наукових та інженерних дослідженнях, а також у сфері освіти [16].

Продукт компанії Waterloo Maple Software, Inc., який часто називають системою символьних обчислень або комп'ютерної алгебри. Maple дозволяє виконувати як чисельні, і аналітичні розрахунки з можливістю редагування тексту і формул на робочому листі [10]. Завдяки представленню формул у поліграфічному форматі, чудовій дво- та тривимірній графіці та анімації Maple є одночасно й потужним науковим графічним редактором. Простий та ефективний мову-інтерпретатор, відкрита архітектура, можливість перетворення кодів у коди C робить його дуже ефективним засобом створення нових алгоритмів. Володіє інтуїтивно зрозумілим інтерфейсом, простими правилами роботи та широким функціоналом [3].

Для обчислення звичайних і частинних похідних в Maple використовується команда (функція) `diff`, перший аргумент, в якій є диференційована функція, а другий - змінна, за якою потрібно брати похідну (рис. 1.7) [17].

```
> diff(sqrt(x^2+3), x);
```

$$\frac{x}{\sqrt{x^2+3}}$$

```
> diff(diff(z*sin(z), z), z); diff(z*sin(z), z, z);
```

$$2 \cos(z) - z \sin(z)$$

$$2 \cos(z) - z \sin(z)$$

Рисунок 1.7 – Приклад знаходження звичайної та частинної похідних в Maple

Джерело: [17]

Для знаходження визначених і невизначених інтегралів в Maple використовується функція `int`, що залежить від двох аргументів, перший з яких

являє собою підінтегральний вираз, а другий - змінну, за якою ведеться інтегрування. Якщо для другої змінної вказано діапазон зміни змінної, то Maple шукає визначений інтеграл, а в протилежному випадку – невизначений (рис. 1.8) [17].

> **int(sin(x)*x^2, x);**

$$-\frac{x^2}{2} \cos(x) + 2 \cos(x) + 2x \sin(x)$$

> **Int(x/(x^2+x+1),x)=int(x/(x^2+x+1),x);**

$$\int \frac{x}{x^2+x+1} dx = \frac{1}{2} \ln(x^2+x+1) - \frac{1}{3} \sqrt{3} \arctan\left(\frac{1}{3}(2x+1)\sqrt{3}\right)$$

> **int(x^3-t*x^2+4*t, x=1..4);**

$$\frac{255}{4} - 9t$$

> **int(sin(x)/x, x);**

$$\text{Si}(x)$$

Рисунок 1.8 – Приклади обчислення визначених і невизначених інтегралів в Maple
Джерело: [17]

Аналогічно більшості інших найбільш важливих математичних операцій, у MathCad існує чисельне і символічне диференціювання. Символьний метод має переваги в тому плані, що результат можна одержати у вигляді функції, яку можна буде використовувати в подальших розрахунках. Чисельний же підхід має переваги в деяких специфічних задачах. MathCad дозволяє обчислювати як звичайну похідну, так і похідні більш високих порядків, а також часткові похідні (рис. 1.9) [2].

Оператор простого диференціювання на панелі Calculus для обчислення першої похідної має два маркери, принцип заповнення яких такий: у верхній вводиться функція, у нижній – змінна, за якою обчислюється похідна.

Результат може бути представлено в символьному вигляді, якщо використати оператор символьного виведення $\rightarrow$, а потім звернутися до символьного процесора Symbolic/Evaluate (Символіка/Обрахувати в символах) (рис. 1.10) [13].

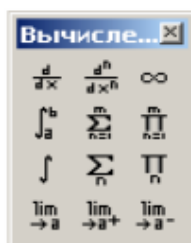


Рисунок 1.9 – Діалогове вікно для обчислення похідних і інтегралів
Джерело: [2]

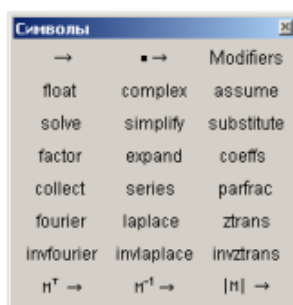


Рисунок 1.10 – Меню символьного процесора Symbolic для обчислення
в символах

Джерело: [13]

Під час символьного диференціювання можна оперувати з функціями декількох змінних. Оператор диференціювання може поєднуватися з будь-яким обчислювальним або символьним оператором. Особливо корисним є оператор Simplify, так як вираз похідної подається в неспрощеному вигляді. Для спрощення відповіді варто використовувати оператори Collect (наводити подібні), Factor (розкладає вирази на множники) і Expand (розкриває дужки).

Приклад обчислення невизначеного інтегралу (рис. 1.11) [14].

$$\int \left(x^3 + \sqrt[3]{x} - \frac{2}{x^3} \right) dx \rightarrow \frac{1}{4} \cdot x^4 + \frac{3}{4} \cdot x^{\frac{4}{3}} + \frac{1}{x^2} \quad \int \frac{1}{\sqrt{4-x^2}} dx \rightarrow \operatorname{asin}\left(\frac{1}{2} \cdot x\right)$$

Рисунок 1.11 – Обчислення невизначеного інтегралу в MathCad

Джерело: [14]

Приклад обчислення визначених інтегралів (рис. 1.12):

$$\begin{aligned} \int_2^3 x^2 dx &\rightarrow \frac{19}{3} & \int_2^3 x^2 dx &= 6.33 & \int_2^3 x^2 dx &= 6.33333 & \int_2^3 x^2 dx &= 6.33333333333333 \\ \int_0^\pi \sin(x) dx &\rightarrow 2 & \int_0^\pi \sin(x) dx &= 2 \\ \int_1^{\ln(2)} x \cdot \exp(x) dx &\rightarrow 2 \cdot \ln(2) - 2 & \int_1^{\ln(2)} x \cdot \exp(x) dx &= -0.61371 \\ \int_0^\pi x^4 \sin\left(\frac{x}{2}\right) dx &\rightarrow 16 \cdot \pi^3 - 384 \cdot \pi + 768 & \int_0^\pi x^4 \sin\left(\frac{x}{2}\right) dx &= 57.7288 \\ \int_1^2 x^3 \exp(2x) dx &\rightarrow \frac{17}{8} \cdot \exp(4) - \frac{1}{8} \cdot \exp(2) & \int_1^2 x^3 \exp(2x) dx &= 115.0974 \end{aligned}$$

Рисунок 1.12 – Приклади обчислення визначених інтегралів в MathCad

Джерело: [14]

Окремо, слід розглянути Python. Це високорівнева мова програмування загального призначення, що підтримує структурне, узагальнене, об'єктно-орієнтоване, функціональне та аспектно-орієнтоване програмування. Для виконання наукових досліджень використовуються додаткові пакети: Numpy, Scipy, Matplotlib та ін. [5].

Дана мова програмування добре підходить для виконання швидких розрахунків з високим ступенем точності, що буває дуже критично для розв'язування певного кола задач, не тільки математичних, а й фізичних, біологічних, тощо.

Порівняльний аналіз систем комп'ютерної математики наведено у таблиці нижче (табл. 1.1).

Таблиця 1.1 – Порівняльний аналіз систем комп'ютерної математики

Критерій порівняння	MatLab	Mathematica	Maple	Python
Інтерфейс	містить три вікна: командне вікно, всі змінні та їх типи та вікно підказок; є рядок запрошення, що позначається знаком ">>"; на відміну від Mathcad, всі функції доводиться вводити з клавіатури.	рядок запрошення на відміну від MatLab поділено на дві області: введення та виведення, які становлять разом область всього виразу; область введення можна редагувати; є палітра з грецькими літерами, різними символами та панель мат аналізу.	можливість роботи в режимі інтерпретатора; дозволяє вводити електронні таблиці, що містять як числа, так і символи; є опції розбиття на параграфи та розділи, а також додавання гіперпосилань; робочі аркуші можна організувати ієрархічно	залежить від середовища розробки.
Робота з масивами та матрицями	аналогічно Mathematica матриці та вектора формуються за допомогою списку елементів. функції вводяться з клавіатури.	багатовимірний набір даних створюється за допомогою списку.	може знаходити власні значення та власні вектори операторів, обчислювати криволінійні координати	Python не має вбудованого типу даних для матриць. Для лінійної алгебри прийнято використовувати numpy, вектора та матриці якого типізовані.
Математичні оператори	на відміну від Mathcad, всі оператори вводяться з клавіатури у вигляді окремих символів і функцій. Дано відносно докладний список операторів.	як і в MatLab оператори доводиться вводити з клавіатури, але деякі можна знайти і на панелі інструментів.	здатний виконувати арифметичні дії у символьному вигляді.	для наукових обчислень використовується бібліотека scipy.

Продовження Таблиці 1.1

Критерій порівняння	MatLab	Mathematica	Maple	Python
Розв'язання рівнянь	вирішує рівняння та системи рівнянь функцій з різними параметрами.	містить кілька функцій для вирішення рівнянь та систем рівнянь. Ви можете знайти корені рівнянь з параметром. Є функція для спеціальних рішень.	пропонує різні способи подання, скорочення та перетворення виразів, наприклад такі операції, як спрощення та розкладання на множники алгебраїчних виразів та приведення їх до різного виду	Для використання символьних обчислень можна використовувати бібліотеку sympy. Однак код, написаний за допомогою sympy, поступається в красі коду, написаному на Mathematica, яка спеціалізована на символьних обчисленнях.
Вбудовані функції	в основному використовуються лише функції, які вводяться з клавіатури.	наведено багато функцій різного призначення з різним числом параметрів.	включені пакети для вирішення завдань лінійної та тензорної алгебри, евклідової та аналітичної геометрії, теорії чисел, теорії ймовірностей та мат.статистики, комбінаторики,	існує дуже багато арифметичних, тригонометричних, гіперболічних, логарифмічних функцій, які стають доступними після імпортування бібліотеки <code>null></code>

З порівняльного аналізу можна бачити, що всі математичні пакети схожі між собою. Але серед них особливо виділяється Python, у якого можна виділити такі основні переваги:

- 1) вільно розповсюджується та працює практично на всіх відомих платформах;
- 2) простий для вивчення і має такі важливі властивості, як інтерактивність та інтерпретованість;
- 3) має багатий набір функцій стандартної бібліотеки та широкий набір зовнішніх бібліотек, що дозволяють вирішувати практично будь-які завдання;

4) простір імен: наприклад, Matlab підтримує простір імен для функцій, які використовуються в програмі, але ядро Matlab не має просторів імен (тобто кожна функція визначена в глобальному просторі імен). Python працює з модулями, які потрібно імпортувати, якщо вони будуть використовуватися;

5) самоаналіз. Оскільки програма має чітку структуру, самоаналіз є дуже простим. Закриті змінні існують лише за згодою, тому можна отримати доступ до будь-якої частини програми, включаючи деякі внутрішні компоненти Python.

6) інструментарій GUI, що дозволяє створювати зовнішній інтерфейс.

Таким чином, Python – це, найбільш вдало збалансована система та безперечний лідер щодо можливостей символьних обчислень для математики.

1.3 Постановка задачі

Метою даної дипломної роботи є розробка калькулятора з функціями інтегрування та диференціювання для обчислення інтегралів та похідних. Для досягнення цієї мети необхідно вирішити такі завдання:

Визначення функціональних вимог. Для досягнення мети розробки калькулятора для обчислення інтегралів та похідних необхідно розробити функціонал для створення, редагування та видалення завдань. Важливо забезпечити можливість налаштування умов завдання для обчислення інтегралів та похідних.

Користувацькі вимоги. Інтерфейс користувача повинен бути інтуїтивно зрозумілим та зручним, щоб забезпечити легкість у використанні калькулятора. Необхідно також передбачити можливість індивідуалізації налаштувань відповідно до потреб кожного користувача.

Технічні вимоги. Необхідно визначити технології та інструменти, які будуть використані для розробки калькулятора, зокрема мову програмування. Забезпечення високої продуктивності та стабільності роботи калькулятора є ключовим аспектом, що передбачає оптимізацію коду та використання ефективних алгоритмів.

Інтеграція та тестування. Інтеграція всіх функціональних компонентів калькулятора є важливим етапом розробки, що забезпечує їх узгоджену роботу та взаємодію. Для цього необхідно розробити детальний тестовий план, який включатиме перевірку всіх функціональних можливостей калькулятора. Після цього потрібно виконати тестування калькулятора на різних прикладах, щоб гарантувати його стабільну роботу та відповідність заданим вимогам.

Перед тим як проводити реалізацію розробки калькулятора для обчислення інтегралів та похідних, потрібно окреслити коло основних задач, які повинен виконувати даний калькулятор.

В розділі інтегральне числення існує досить велика кількість задач, які розв'язуються за допомогою інтегралів. Серед основних задач виділяють обчислення невизначених та визначених інтегралів, знаходження площі фігури, обмеженої кривими за допомогою визначених інтегралів, обчислення об'єму тіла навколо різних осей обертання, обчислення довжини дуги лінії, обчислення подвійних та потрійних інтегралів та ін.. Коло задач, досить широке, але всі вони будуються на кінцевому обчисленні інтегралів.

В розділі диференціальне числення виділяють ряд важливих задач, таких як обчислення похідних, обчислення похідних в точці, обчислення похідних вищого порядку, обчислення частинних похідних функції декількох змінних. За допомогою похідних розв'язуються задачі дослідження функцій і побудови графіків, знаходження рівняння дотичної та нормалі, знаходження градієнту функції декількох змінних, знаходження похідної за напрямом, наближені обчислення за допомогою диференціалів та 19н..

Обчислення, які проводитиме калькулятор, направлені на перевірку отриманих раніше результатів за допомогою аналітичного розв'язку. З допомогою калькулятора, можна звірити відповіді і вразі відмінностей, знайти помилки в аналітичному розв'язанні поставленої задачі.

Основними задачами, які буде розв'язувати калькулятор будуть наступні:

- обчислення невизначених інтегралів;

- обчислення визначених інтегралів;
- обчислення подвійних інтегралів;
- обчислення потрійних інтегралів;
- обчислення площі фігури, обмеженої кривими;
- обчислення об'єму тіла обертання навколо координатних осей;
- обчислення похідних;
- обчислення похідних вищих порядків;
- обчислення частинних похідних.

Функціонал даного калькулятора дозволить проводити перевірку розв'язків поставлених задач, що є досить зручним інструментом забезпечення правильного результату розв'язування.

Висновок до розділу 1

В результаті написання даного розділу було розглянуто основні пакети прикладних програм, які використовуються для обчислення похідних та інтегралів різного рівня складності. Кожен прикладний пакет має свої особливості здійснення символічних та чисельних обчислень, які можна використовувати для розв'язку різноманітних задач. Також, була здійснена постановка завдання по розробці калькулятора з функціями інтегрування та диференціювання, який дасть можливість швидко знаходити значення похідних та інтегралів, після введення відповідних функцій. Було описано всі необхідні умови, які потрібно реалізувати при розробці калькулятора.

РОЗДІЛ 2

ПРОЄКТУВАННЯ ТА РОЗРОБКА КАЛЬКУЛЯТОРА ДЛЯ ОБЧИСЛЕННЯ ІНТЕГРАЛІВ ТА ПОХІДНИХ

2.1 Опис проєкту створення калькулятора

Для проєктування калькулятора для обчислення інтегралів та похідних, потрібно детально охарактеризувати всі етапи його створення.

Калькулятор створюватиметься із використанням мови програмування Python.

В загальному, схема створення калькулятора має вигляд (рис. 2.1):

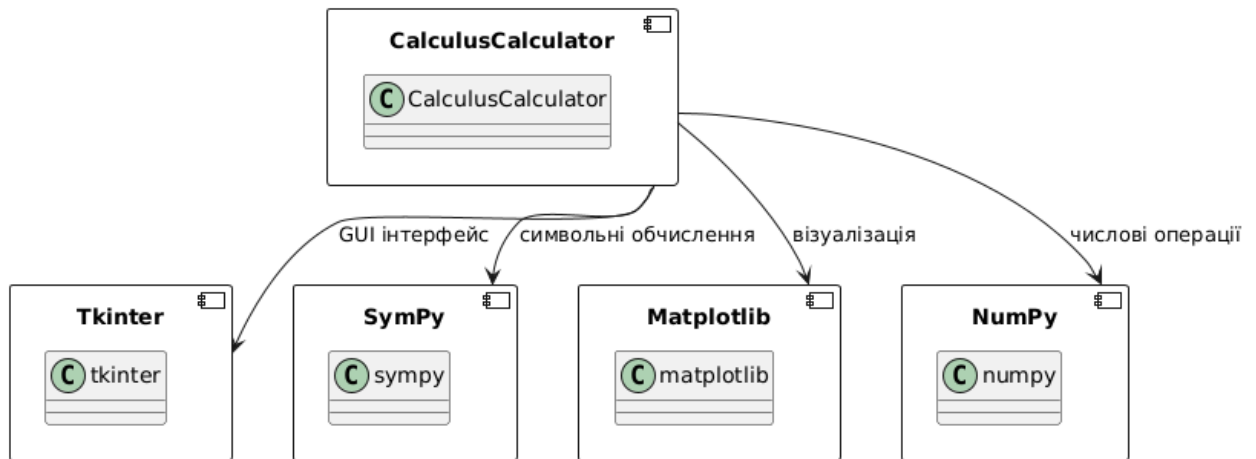


Рисунок 2.1 – Схема створення калькулятора для обчислення інтегралів та похідних

Джерело: розроблено автором

Калькулятор повинен безпосередньо займатись обчисленням інтегралів та похідних, а також, виконувати супутні задачі, які включають в себе інтегрування та диференціювання. Основні функції, які виконуватиме калькулятор, після взаємодії з користувачем (рис. 2.2):



Рисунок 2.2 – Основні функції, які виконуватиме калькулятор при взаємодії з користувачем

Джерело: розроблено автором

Даний набір функції охоплює переважну більшість задач, які стосуються інтегрального та диференціального числення. Обчислення площі та об'єму передбачає використання визначених інтегралів. Обчислення об'ємів просторових тіл передбачає використання подвійних та потрійних інтегралів. Також, потрібна графічна інтерпретація кривих для обчислення площі плоских фігур. Що стосується обчислення похідних, окрім похідних функції однієї змінної існують похідні функцій декількох змінних. Також, дуже часто

розв'язуються задачі на знаходження похідних вищих порядків та похідних в точках.

Обчислення інтегралів відбуватиметься наступним чином (рис. 2.3).

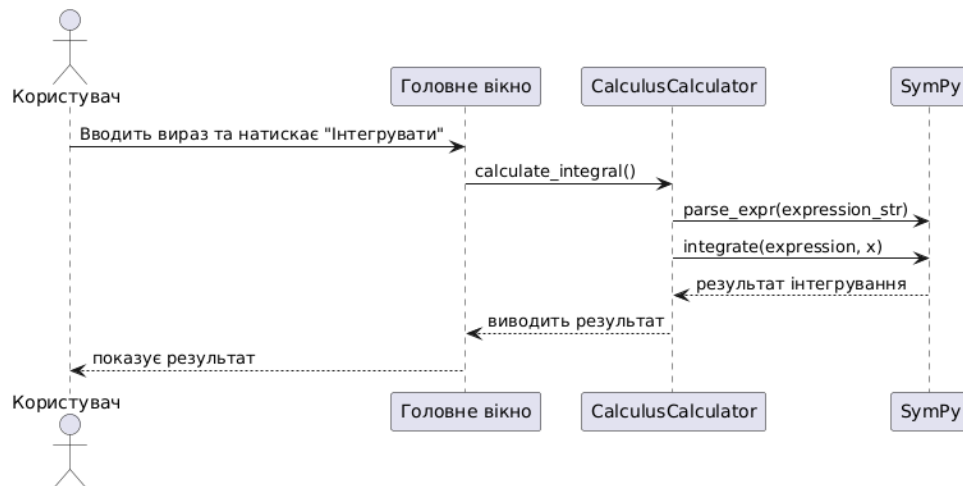


Рисунок 2.3 – Алгоритм обчислення інтегралів

Джерело: розроблено автором

Обчислення похідних реалізовуватиметься наступним чином (рис. 2.4).

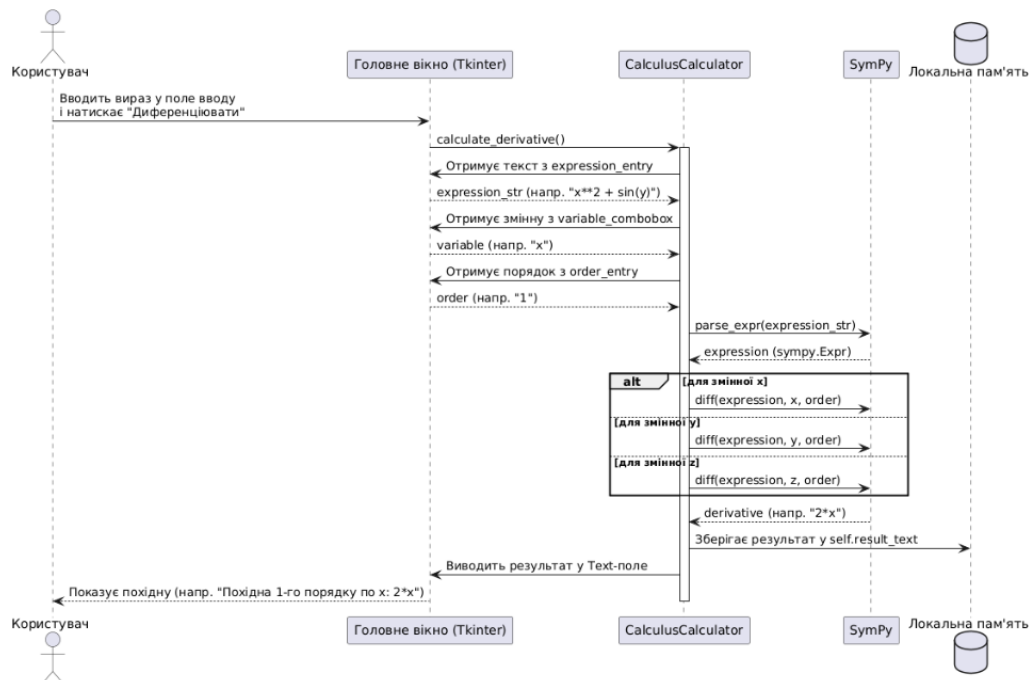


Рисунок 2.4 – Алгоритм обчислення похідних

Джерело: розроблено автором

2.2 Основні бібліотеки Python для реалізації обчислень

Наукові обчислення відіграють ключову роль в аналізі даних, моделюванні та вирішенні складних математичних завдань. Python став однією з найпопулярніших мов для цих цілей завдяки своїй простоті та потужним бібліотекам [5].

Python пропонує широкий спектр інструментів для наукових обчислень, що робить його ідеальним вибором для дослідників, аналітиків даних та інженерів. Завдяки відкритим бібліотекам та активній спільноті розробників, Python постійно розвивається та пропонує нові можливості для вирішення складних завдань.

NumPy (Numerical Python) – це фундаментальна бібліотека для наукових обчислень на Python. Вона надає підтримку багатовимірних масивів та матриць, а також функції для виконання різних математичних операцій. NumPy є основою більшості інших бібліотек, які у наукових обчисленнях [9].

Основні функції NumPy:

1. Масиви. NumPy дозволяє створювати та працювати з багатовимірними масивами, що є основою для більшості наукових обчислень. Масиви NumPy мають високу продуктивність і дозволяють ефективно виконувати операції з великими обсягами даних;
2. Лінійна алгебра. Бібліотека включає функції для виконання операцій лінійної алгебри, таких як множення матриць, знаходження власних значень та векторів. Ці функції необхідні вирішення багатьох завдань у галузі математики та фізики [9];
3. Статистика. NumPy надає функції для обчислення статистичних показників, таких як середнє, медіана та стандартне відхилення. Ці функції корисні для аналізу даних та виявлення закономірностей.

NumPy також підтримує різні типи даних, включаючи цілі числа, числа з плаваючою точкою та комплексні числа. Це дозволяє використовувати

бібліотеку для вирішення широкого спектру завдань, від простих арифметичних операцій до складних математичних обчислень.

Pandas – це бібліотека для роботи з даними, яка надає високорівневі структури даних та інструменти для аналізу даних. Основні структури даних у Pandas – це Series та DataFrame. Pandas дозволяє легко маніпулювати даними, виконувати їх фільтрацію, агрегацію та перетворення [15].

Основні функції Pandas:

1. Series. Одновимірний масив, аналогічний масиву NumPy, але з індексами. Series дозволяє легко звертатися до елементів масиву за їх індексами, що полегшує роботу з даними.
2. DataFrame. Двовимірна таблиця даних, аналогічна таблиці у базі даних або електронних таблицях. DataFrame дозволяє зберігати дані у структурованому вигляді та виконувати над ними різні операції.
3. Маніпуляція даними. Pandas надає функції для фільтрації, агрегації та перетворення даних. Ці функції дозволяють легко обробляти великі обсяги даних та витягувати з них корисну інформацію.

Pandas також підтримує роботу з тимчасовими рядами, що робить його ідеальним інструментом для аналізу даних, пов'язаних з часом. Бібліотека дозволяє легко виконувати операції з датами та часовими інтервалами, що корисно для аналізу фінансових даних, даних про продаж та інших часових рядів [18].

SciPy (Scientific Python) – це бібліотека, яка розширює можливості NumPy, надаючи додаткові функції для наукових обчислень. Вона включає модулі для оптимізації, інтеграції, інтерполяції, лінійної алгебри та статистики. SciPy є потужним інструментом для вирішення складних математичних завдань.

Основні функції SciPy:

1. Оптимізація: Функції знаходження мінімумів та максимумів функцій. Ці функції корисні для вирішення задач оптимізації, таких як мінімізація витрат або максимізація прибутку;

2. Інтегрування: Інструменти для чисельного інтегрування. SciPy надає функції для обчислення певних та невизначених інтегралів, що корисно для вирішення завдань у галузі математики та фізики;

3. Інтерполяція: Методи для інтерполяції даних. SciPy дозволяє виконувати інтерполяцію даних, що корисно для відновлення значень, що відсутні, або згладжування даних [18];

4. Статистика: Розширені статистичні функції та розподіли. SciPy надає функції для виконання різних статистичних тестів та аналізу даних.

SciPy також включає модулі для роботи з сигналами, зображеннями та багатьма іншими типами даних. Це робить бібліотеку універсальним інструментом для вирішення широкого спектру завдань у галузі наукових обчислень.

Візуалізація даних – це важливий аспект аналізу даних, який допомагає краще зрозуміти та інтерпретувати результати. Matplotlib та Seaborn – це дві популярні бібліотеки для створення графіків та діаграм. Вони дозволяють створювати гарні та інформативні візуалізації даних.

Основні функції Matplotlib:

1. Графіки. Створення різних типів графіків, таких як лінійні графіки, гістограми та діаграми розсіювання. Matplotlib дозволяє створювати графіки будь-якої складності та налаштовувати їх відповідно до ваших потреб.

2. Налаштування графіків. Можливість налаштування осей, заголовків, легенд та інших елементів графіків. Matplotlib пропонує широкий спектр інструментів для налаштування графіків, що дозволяє створювати професійні візуалізації даних [18].

Основні функції Seaborn:

1. Статистичні графіки. Створення складних статистичних графіків з мінімальними зусиллями. Seaborn дозволяє легко створювати графіки, які показують розподіл даних, кореляції та інші статистичні характеристики.

2. Інтеграція з Pandas. Легка інтеграція з DataFrame для візуалізації даних. Seaborn дозволяє легко візуалізувати дані, що зберігаються в DataFrame, що спрощує аналіз даних.

Matplotlib і Seaborn дозволяють створювати графіки будь-якої складності та налаштовувати їх відповідно до ваших потреб. Ці бібліотеки є потужними інструментами для візуалізації даних та допомагають краще зрозуміти та інтерпретувати результати аналізу даних.

Висновок до розділу 2

В результаті проведеної роботи було проведено обґрунтування вибору засобу створення калькулятора для обчислення інтегралів та похідних. Використання мови програмування Python є одним із найоптимальніших варіантів для створення калькулятора для обчислення інтегралів та похідних. Дана мова програмування має декілька важливих бібліотек, які відповідають за математичні обчислення. Це бібліотеки NumPy, SymPy та Matplotlib. Саме ці три бібліотеки будуть покладені в основу створення калькулятора для обчислення інтегралів та похідних.

Також, в даному розділі було наведено основний функціонал калькулятора, який буде створено, описані задачі, які він зможе розв'язувати. Окрім основних задач по обчисленню інтегралів та похідних, калькулятор зможе обчислювати площі фігур та об'єми тіл обертання, подвійні та потрійні інтеграли, частинні похідні, похідну функції в точці, похідні вищих порядків.

РОЗДІЛ 3

ПРОГРАМНА РЕАЛІЗАЦІЯ КАЛЬКУЛЯТОРА ДЛЯ ОБЧИСЛЕННЯ ІНТЕГРАЛІВ ТА ПОХІДНИХ

3.1 Огляд застосованих бібліотек при реалізації калькулятора

В основі розробленого калькулятора лежать наступні бібліотеки:

- 1) NumPy;
- 2) SymPy;
- 3) Matplotlib;

Бібліотека NumPy.

NumPy – це бібліотека наукових обчислень для Python. Вона пропонує високорівневі математичні функції та багатовимірну структуру (відому як `ndarray`) для маніпулювання великими наборами даних.

Основні можливості NumPy [18]:

1) Масиви. NumPy дозволяє створювати масиви будь-якої розмірності, що робить його ідеальним для роботи з даними. Масиви NumPy є більш ефективними за списки Python, оскільки вони займають менше пам'яті і забезпечують швидший доступ до даних;

2) Математичні функції. NumPy надає функції для виконання арифметичних операцій, таких як додавання, віднімання, множення та ділення;

3) Статистичні функції: Включає функції для обчислення середнього, медіани, стандартного відхилення та інших статистичних показників.

Лінійна алгебра: NumPy має вбудовані функції для роботи з матрицями, включаючи обчислення детермінантів, обернених матриць та розв'язування систем лінійних рівнянь. NumPy надає можливості для генерації випадкових чисел з різних розподілів (нормальний, рівномірний тощо). NumPy дозволяє легко отримувати доступ до елементів масиву, а також виконувати зрізи для вибору під масивів [18].

Бібліотека SymPy. SymPy (Symbolic Python) – це бібліотека Python для символної математики. Вона надає можливості для виконання обчислень з символними виразами, подібно до таких систем, як Mathematica або Maple. Замість обчислення числових значень, SymPy працює з математичними символами, що дозволяє виконувати алгебраїчні перетворення, обчислювати похідні, інтеграли, розв'язувати рівняння та багато іншого.

Основні можливості SymPy:

- 1) Створення символних змінних: SymPy дозволяє визначати символні змінні (наприклад, x , y , z), які використовуються для представлення математичних змінних;
- 2) Визначення функцій: можна визначати функції, які залежать від символних змінних;
- 3) Алгебраїчні обчислення: SymPy може спрощувати математичні вирази, розкривати дужки, збирати подібні члени та виконувати інші алгебраїчні перетворення;
- 4) Диференціальне числення: SymPy дозволяє знаходити похідні функцій за однією або кількома змінними;
- 5) Інтегральне числення: SymPy може обчислювати як невизначені, так і визначені інтеграли, а також, інтеграли від декількох змінних;
- 6) Розв'язування рівнянь: SymPy може розв'язувати алгебраїчні рівняння (лінійні, квадратні, кубічні тощо), системи рівнянь та диференціальні рівняння;
- 7) Лінійна алгебра: SymPy дозволяє створювати, виконувати операції з матрицями (додавання, віднімання, множення, обчислення детермінантів, обернених матриць тощо) [18].

Matplotlib – це фундаментальна бібліотека для візуалізації даних у Python, яка надає широкий спектр інструментів для створення статичних, анімованих та інтерактивних графіків. Вона є основою для багатьох інших бібліотек візуалізації в Python, таких як Seaborn, що робить її важливим

інструментом для аналізу даних, представлення результатів та створення інформативних візуалізацій.

Основні компоненти та можливості Matplotlib [18]:

1) Архітектура та організація:

Figure (Фігура): Верхній рівень, який представляє вікно або сторінку, де відображаються графіки. Може містити один або кілька об'єктів Axes.

Axes (Осі): Містить осі (x, y), мітки, заголовки та інші елементи графіка.

Artist (Об'єкти): Все, що відображається на графіку (лінії, кола, текст, зображення тощо). Artist — це базовий клас для всіх графічних елементів.

2) Типи графіків та візуалізацій: Відображення трендів та залежностей між змінними; візуалізація взаємозв'язків між двома або більше змінними.

3) Налаштування та стилізація.

4) Робота з текстом: Додавання тексту, анотацій, підписів, математичних формул (підтримка LaTeX).

3.2 Основні фрагменти коду реалізації функцій калькулятора для обчислення інтегралів та похідних

Розробка функціоналу калькулятора вимагає використання різних команд та функцій, тому для більш чіткого розуміння варто розглянути основні фрагменти коду, який реалізує даний калькулятор для обчислення інтегралів та похідних.

Для розробки коду використовувались відомі бібліотеки Python, а також, бібліотека Tkinter, для розробки інтерфейсу.

Розробка коду на мові програмування Python починається з імпорту необхідних бібліотек (рис. 3.1).

Для обчислення інтегралів використовується наступний набір команд (рис. 3.2).

Обчислення визначеного інтегралу відбувається наступним чином (рис. 3.3).

```
import tkinter as tk
from tkinter import ttk, simpledialog
from sympy import integrate, diff, symbols, parse_expr, pi
import matplotlib.pyplot as plt
import numpy as np
```

Рисунок 3.1 – Імпорт необхідних бібліотек для обчислення інтегралів та похідних

Джерело: розроблено автором

```
def calculate_integral(self):
    try:
        expression_str = self.expression_entry.get()
        expression = parse_expr(expression_str)
        integral = integrate(expression, self.x)
        self.result_text.delete(1.0, tk.END)
        self.result_text.insert(tk.END, f"Інтеграл: {integral}")
    except Exception as e:
        self.result_text.delete(1.0, tk.END)
        self.result_text.insert(tk.END, f"Помилка: {e}")
```

Рисунок 3.2 – Фрагмент коду для обчислення інтегралів

Джерело: розроблено автором

```
def calculate_definite_integral(self):
    try:
        expression_str = self.expression_entry.get()
        expression = parse_expr(expression_str)

        # Запит меж інтегрування
        lower_bound = simpledialog.askfloat("Межі інтегрування", "Введіть нижню межу:")
        upper_bound = simpledialog.askfloat("Межі інтегрування", "Введіть верхню межу:")

        definite_integral = integrate(expression, (self.x, lower_bound, upper_bound))
        if definite_integral < 0:
            self.result_text.delete(1.0, tk.END)
            self.result_text.insert(tk.END, "Помилка: Площа не може бути від'ємною.")
        else:
            self.result_text.delete(1.0, tk.END)
            self.result_text.insert(tk.END, f"Визначений інтеграл: {definite_integral}")
    except Exception as e:
        self.result_text.delete(1.0, tk.END)
        self.result_text.insert(tk.END, f"Помилка: {e}")
```

Рисунок 3.3 – Обчислення визначеного інтегралу

Джерело: розроблено автором

Для обчислення подвійних інтегралів використовуються такі команди (рис. 3.4):

```

def calculate_double_integral(self):
    try:
        expression_str = self.expression_entry.get()
        expression = parse_expr(expression_str)

        # Запит меж інтегрування
        x_lower = simpledialog.askfloat("Межі інтегрування", "Введіть нижню межу для x:")
        x_upper = simpledialog.askfloat("Межі інтегрування", "Введіть верхню межу для x:")
        y_lower = simpledialog.askfloat("Межі інтегрування", "Введіть нижню межу для y:")
        y_upper = simpledialog.askfloat("Межі інтегрування", "Введіть верхню межу для y:")

        double_integral = integrate(integrate(expression, (self.y, y_lower, y_upper)), (self.x, x_lower, x_upper))
        if double_integral < 0:
            self.result_text.delete(1.0, tk.END)
            self.result_text.insert(tk.END, "Помилка: Площа не може бути від'ємною.")
        else:
            self.result_text.delete(1.0, tk.END)
            self.result_text.insert(tk.END, f"Подвійний інтеграл: {double_integral}")
    except Exception as e:
        self.result_text.delete(1.0, tk.END)
        self.result_text.insert(tk.END, f"Помилка: {e}")

```

Рисунок 3.4 Обчислення подвійних інтегралів

Джерело: розроблено автором

Обчислення потрійних інтегралів проводиться за схожим принципом (рис. 3.5):

```

def calculate_triple_integral(self):
    try:
        expression_str = self.expression_entry.get()
        expression = parse_expr(expression_str)

        # Запит меж інтегрування
        x_lower = simpledialog.askfloat("Межі інтегрування", "Введіть нижню межу для x:")
        x_upper = simpledialog.askfloat("Межі інтегрування", "Введіть верхню межу для x:")
        y_lower = simpledialog.askfloat("Межі інтегрування", "Введіть нижню межу для y:")
        y_upper = simpledialog.askfloat("Межі інтегрування", "Введіть верхню межу для y:")
        z_lower = simpledialog.askfloat("Межі інтегрування", "Введіть нижню межу для z:")
        z_upper = simpledialog.askfloat("Межі інтегрування", "Введіть верхню межу для z:")

        triple_integral = integrate(integrate(integrate(expression, (self.z, z_lower, z_upper)), (self.y, y_lower, y_upper)), (self.x, x_lower, x_upper))
        if triple_integral < 0:
            self.result_text.delete(1.0, tk.END)
            self.result_text.insert(tk.END, "Помилка: Об'єм не може бути від'ємним.")
        else:
            self.result_text.delete(1.0, tk.END)
            self.result_text.insert(tk.END, f"Потрійний інтеграл: {triple_integral}")
    except Exception as e:
        self.result_text.delete(1.0, tk.END)
        self.result_text.insert(tk.END, f"Помилка: {e}")

```

Рисунок 3.5 – Реалізація обчислення потрійних інтегралів

Джерело: розроблено автором

Також, представлений калькулятор може знаходити розв'язок задачі знаходження площі фігури, обмеженої двома кривими (рис. 3.6):

Також, калькулятор може виводити графіки функцій, які обмежують фігуру, площу якої потрібно знайти (рис. 3.7):

```

def calculate_area(self):
    try:
        # Введення функцій
        func1_str = simpledialog.askstring("#функція 1", "Введіть першу функцію:")
        func2_str = simpledialog.askstring("#функція 2", "Введіть другу функцію:")
        func1 = parse_expr(func1_str)
        func2 = parse_expr(func2_str)

        # Запит меж інтегрування
        x_lower = simpledialog.askfloat("Межі інтегрування", "Введіть нижню межу для x:")
        x_upper = simpledialog.askfloat("Межі інтегрування", "Введіть верхню межу для x:")

        # Обчислення площі
        area = integrate(func1 - func2, (self.x, x_lower, x_upper))
        if area < 0:
            self.result_text.delete(1.0, tk.END)
            self.result_text.insert(tk.END, "Помилка: Площа не може бути від'ємною.")
        else:
            self.result_text.delete(1.0, tk.END)
            self.result_text.insert(tk.END, f"Площа фігури: {area}")

        # Візуалізація графіка функцій
        self.plot_graph(func1, func2, x_lower, x_upper)

    except Exception as e:
        self.result_text.delete(1.0, tk.END)
        self.result_text.insert(tk.END, f"Помилка: {e}")

```

Рисунок 3.6 Реалізація обчислення площі фігури

Джерело: розроблено автором

```

def plot_graph(self, func1, func2, x_lower, x_upper):
    # Створення значень для побудови графіка
    x_vals = np.linspace(x_lower, x_upper, 400)
    y_vals1 = [func1.subs(self.x, val) for val in x_vals]
    y_vals2 = [func2.subs(self.x, val) for val in x_vals]

    # Побудова графіка
    plt.figure(figsize=(8, 6))
    plt.plot(x_vals, y_vals1, label=f'y = {func1}', color='blue')
    plt.plot(x_vals, y_vals2, label=f'y = {func2}', color='orange')
    plt.axhline(0, color='black', lw=0.5, ls='--')
    plt.axvline(0, color='black', lw=0.5, ls='--')
    plt.title("Графік функцій")
    plt.xlabel("x")
    plt.ylabel("y")
    plt.legend()
    plt.grid()
    plt.show()

```

Рисунок 3.7 – Побудова графіків функцій, що обмежують фігуру

Джерело: розроблено автором

Також, калькулятор має функціонал для обчислення об'ємів тіл, які обертаються навколо осей (рис. 3.8):

```

def calculate_volume(self):
    try:
        func_str = simpledialog.askstring("#функція", "Введіть функцію для об'єму обертання:")
        func = parse_expr(func_str)

        # Запит меж інтегрування
        x_lower = simpledialog.askfloat("Межі інтегрування", "Введіть нижню межу для x:")
        x_upper = simpledialog.askfloat("Межі інтегрування", "Введіть верхню межу для x:")
        axis = simpledialog.askstring("Ось обертання", "Навколо якої осі обертається? (x або y)")

        # Обчислення об'єму
        if axis.lower() == 'x':
            volume = integrate(pi * func**2, (self.x, x_lower, x_upper))
        elif axis.lower() == 'y':
            volume = integrate(pi * func, (self.y, x_lower, x_upper))
        else:
            volume = "Невідома вісь обертання"

        if isinstance(volume, str): # Перевірка на помилку
            self.result_text.delete(1.0, tk.END)
            self.result_text.insert(tk.END, volume)
        elif volume < 0:
            self.result_text.delete(1.0, tk.END)
            self.result_text.insert(tk.END, "Помилка: Об'єм не може бути від'ємним.")
        else:
            self.result_text.delete(1.0, tk.END)
            self.result_text.insert(tk.END, f"Об'єм тіла обертання: {volume}")
    except Exception as e:
        self.result_text.delete(1.0, tk.END)
        self.result_text.insert(tk.END, f"Помилка: {e}")

```

Рисунок 3.8 – Обчислення об'єму тіла обертання

Джерело: розроблено автором

Що стосується диференціального числення, то в даному калькуляторі реалізовано декілька важливих функцій.

Обчислення похідної функції однієї змінної (рис. 3.9):

```

def calculate_derivative(self):
    try:
        expression_str = self.expression_entry.get()
        expression = parse_expr(expression_str)
        variable = self.variable_combobox.get()
        order = int(self.order_entry.get())
        if variable == 'x':
            derivative = diff(expression, self.x, order)
        elif variable == 'y':
            derivative = diff(expression, self.y, order)
        elif variable == 'z':
            derivative = diff(expression, self.z, order)
        else:
            derivative = "Невідома змінна"
    
```

Рисунок 3.9 Обчислення похідних функції однієї змінної

Джерело: розроблено автором

Також, калькулятор може обчислювати похідні від функції однієї змінної вищих порядків (рис. 3.10):

```
def calculate_derivative(self):
    try:
        expression_str = self.expression_entry.get()
        expression = parse_expr(expression_str)
        variable = self.variable_combobox.get()
        order = int(self.order_entry.get())
        if variable == 'x':
            derivative = diff(expression, self.x, order)
        elif variable == 'y':
            derivative = diff(expression, self.y, order)
        elif variable == 'z':
            derivative = diff(expression, self.z, order)
        else:
            derivative = "Невідома змінна"

        self.result_text.delete(1.0, tk.END)
        self.result_text.insert(tk.END, f"Похідна {order}-го порядку по {variable}: {derivative}")
    except Exception as e:
        self.result_text.delete(1.0, tk.END)
        self.result_text.insert(tk.END, f"Помилка: {e}")
```

Рисунок 3.10 – Обчислення похідних вищих порядків

Джерело: розроблено автором

В функціоналі калькулятора також реалізовано знаходження частинних похідних функції декількох змінних (рис. 3.11):

```
def calculate_partial_derivative(self):
    try:
        expression_str = self.expression_entry.get()
        expression = parse_expr(expression_str)
        variable = self.variable_combobox.get()
        order = int(self.order_entry.get())

        if variable == 'x':
            derivative = diff(expression, self.x, order)
        elif variable == 'y':
            derivative = diff(expression, self.y, order)
        elif variable == 'z':
            derivative = diff(expression, self.z, order)
        else:
            derivative = "Невідома змінна"
```

Рисунок 3.11 – Обчислення частинних похідних

Джерело: розроблено автором

Як у випадку із функціями однієї змінної, калькулятор може обчислювати частинні похідні вищих порядків (рис. 3.12):

```
def calculate_partial_derivative(self):
    try:
        expression_str = self.expression_entry.get()
        expression = parse_expr(expression_str)
        variable = self.variable_combobox.get()
        order = int(self.order_entry.get())

        if variable == 'x':
            derivative = diff(expression, self.x, order)
        elif variable == 'y':
            derivative = diff(expression, self.y, order)
        elif variable == 'z':
            derivative = diff(expression, self.z, order)
        else:
            derivative = "Невідома змінна"

        self.result_text.delete(1.0, tk.END)
        self.result_text.insert(tk.END, f"Частинна похідна {order}-го порядку по {variable}: {derivative}")
    except Exception as e:
        self.result_text.delete(1.0, tk.END)
        self.result_text.insert(tk.END, f"Помилка: {e}")
```

Рисунок 3.12 – Обчислення частинних похідних вищих порядків

Джерело: розроблено автором

Не менш важливою є реалізація обчислення похідної функції в точці, так як це корисно при розв’язуванні різних задач з допомогою похідної (рис. 3.13):

```
def calculate_derivative_at_point(self):
    try:
        expression_str = self.expression_entry.get()
        expression = parse_expr(expression_str)
        variable = self.variable_combobox.get()
        order = int(self.order_entry.get())

        if variable == 'x':
            derivative = diff(expression, self.x, order)
            point_str = simpledialog.askstring("Точка", "Введіть значення x:")
            if point_str is None:
                return
            try:
                point = float(point_str)
                derivative_at_point = derivative.subs(self.x, point)
                self.result_text.delete(1.0, tk.END)
                self.result_text.insert(tk.END, f"Похідна {order}-го порядку по {variable} в точці x={point}: {derivative_at_point}")
            except ValueError:
                self.result_text.delete(1.0, tk.END)
                self.result_text.insert(tk.END, "Помилка: Невірне значення точки.")
```

Рисунок 3.13 – Обчислення значень похідних в точці

Джерело: розроблено автором

Варто зазначити, що калькулятор також обчислює значення похідних вищих порядків в заданій точці.

Також, калькулятор обчислює значення частинних похідних і частинних похідних вищих порядків у вказаній точці (рис. 3.14):

Аналогічно, для змінної «z» (рис. 3.15).

```
elif variable == 'y':
    derivative = diff(expression, self.y, order)
    point_str = simpdialog.askstring("Точка", "Введіть значення y:")
    if point_str is None:
        return
    try:
        point = float(point_str)
        derivative_at_point = derivative.subs(self.y, point)
        self.result_text.delete(1.0, tk.END)
        self.result_text.insert(tk.END, f"Похідна {order}-го порядку по {variable} в точці y={point}: {derivative_at_point}")
    except ValueError:
        self.result_text.delete(1.0, tk.END)
        self.result_text.insert(tk.END, "Помилка: Невірне значення точки.")
```

Рисунок 3.14 – Обчислення частинних похідних і частинних похідних вищих порядків у вказаній точці по змінній «y»

Джерело: розроблено автором

```
elif variable == 'z':
    derivative = diff(expression, self.z, order)
    point_str = simpdialog.askstring("Точка", "Введіть значення z:")
    if point_str is None:
        return
    try:
        point = float(point_str)
        derivative_at_point = derivative.subs(self.z, point)
        self.result_text.delete(1.0, tk.END)
        self.result_text.insert(tk.END, f"Похідна {order}-го порядку по {variable} в точці z={point}: {derivative_at_point}")
    except ValueError:
        self.result_text.delete(1.0, tk.END)
        self.result_text.insert(tk.END, "Помилка: Невірне значення точки.")
else:
    self.result_text.delete(1.0, tk.END)
    self.result_text.insert(tk.END, "Помилка: Невідома змінна")
except Exception as e:
    self.result_text.delete(1.0, tk.END)
    self.result_text.insert(tk.END, f"Помилка: {e}")
```

Рисунок 3.15 – Обчислення частинних похідних і частинних похідних вищих порядків у вказаній точці по змінній «z»

Джерело: розроблено автором

Якщо при введенні даних завдання, відповідь до якого треба перевірити виникла помилка, можна повністю видалити неправильну умову, за допомогою кнопки (рис. 3.16).

Для виведення всіх функцій калькулятора було розроблено графічний інтерфейс, який реалізує наступний фрагмент коду (рис. 3.17-3.19).


```
def clear_entry(self):
    self.expression_entry.delete(0, tk.END)
    self.result_text.delete(1.0, tk.END)
    self.order_entry.delete(0, tk.END)
    self.order_entry.insert(0, "1")
```

Рисунок 3.16 – Функція видалення неправильних умов завдань

Джерело: розроблено автором

```
# Елементи інтерфейсу
self.create_widgets()

def create_widgets(self):
    # Введення виразу
    self.expression_label = ttk.Label(self.master, text="Введіть вираз:")
    self.expression_label.grid(row=0, column=0, padx=5, pady=5, sticky=tk.W)

    self.expression_entry = ttk.Entry(self.master, width=40)
    self.expression_entry.grid(row=0, column=1, columnspan=3, padx=5, pady=5, sticky=tk.EW)

    # Вибір змінної для диференціювання
    self.variable_label = ttk.Label(self.master, text="Змінна для частинної похідної:")
    self.variable_label.grid(row=1, column=0, padx=5, pady=5, sticky=tk.W)

    self.variable_combobox = ttk.Combobox(self.master, values=['x', 'y', 'z'], state="readonly")
    self.variable_combobox.set('x') # Default variable
    self.variable_combobox.grid(row=1, column=1, padx=5, pady=5, sticky=tk.EW)

    # Порядок похідної
    self.order_label = ttk.Label(self.master, text="Порядок похідної:")
    self.order_label.grid(row=1, column=2, padx=5, pady=5, sticky=tk.W)

    self.order_entry = ttk.Entry(self.master, width=5)
    self.order_entry.insert(0, "1") # Default order
    self.order_entry.grid(row=1, column=3, padx=5, pady=5, sticky=tk.EW)
```

Рисунок 3.17 – Реалізація інтерфейсу калькулятора

Джерело: розроблено автором

```
# Кнопки
self.integrate_button = ttk.Button(self.master, text="Інтегрувати", command=self.calculate_integral)
self.integrate_button.grid(row=2, column=0, padx=5, pady=5, sticky=tk.EW)

self.differentiate_button = ttk.Button(self.master, text="Диференціювати", command=self.calculate_derivative)
self.differentiate_button.grid(row=2, column=1, padx=5, pady=5, sticky=tk.EW)

self.partial_derivative_button = ttk.Button(self.master, text="Частинна похідна", command=self.calculate_partial_derivative)
self.partial_derivative_button.grid(row=2, column=2, padx=5, pady=5, sticky=tk.EW)

self.derivative_at_point_button = ttk.Button(self.master, text="Похідна в точці", command=self.calculate_derivative_at_point)
self.derivative_at_point_button.grid(row=2, column=3, padx=5, pady=5, sticky=tk.EW)

self.double_integral_button = ttk.Button(self.master, text="Подвійний інтеграл", command=self.calculate_double_integral)
self.double_integral_button.grid(row=3, column=0, padx=5, pady=5, sticky=tk.EW)

self.triple_integral_button = ttk.Button(self.master, text="Потрійний інтеграл", command=self.calculate_triple_integral)
self.triple_integral_button.grid(row=3, column=1, padx=5, pady=5, sticky=tk.EW)

self.definite_integral_button = ttk.Button(self.master, text="Визначений інтеграл", command=self.calculate_definite_integral)
self.definite_integral_button.grid(row=3, column=2, padx=5, pady=5, sticky=tk.EW)

self.area_button = ttk.Button(self.master, text="Площа фігури", command=self.calculate_area)
self.area_button.grid(row=4, column=0, padx=5, pady=5, sticky=tk.EW)

self.volume_button = ttk.Button(self.master, text="Об'єм тіла обертання", command=self.calculate_volume)
self.volume_button.grid(row=4, column=1, padx=5, pady=5, sticky=tk.EW)

self.clear_button = ttk.Button(self.master, text="Очистити", command=self.clear_entry)
self.clear_button.grid(row=4, column=2, padx=5, pady=5, sticky=tk.EW)
```

Рисунок 3.18 – Кнопки інтерфейсу калькулятора

Джерело: розроблено автором

```
# Виведення результату
self.result_label = ttk.Label(self.master, text="Результат:")
self.result_label.grid(row=5, column=0, padx=5, pady=5, sticky=tk.W)

self.result_text = tk.Text(self.master, height=5, width=50)
self.result_text.grid(row=5, column=1, columnspan=3, padx=5, pady=5, sticky=tk.EW)
```

Рисунок 3.19 – Виведення результату

Джерело: розроблено автором

Графічний інтерфейс калькулятора для обчислення інтегралів та похідних має вигляд (рис. 3.20):

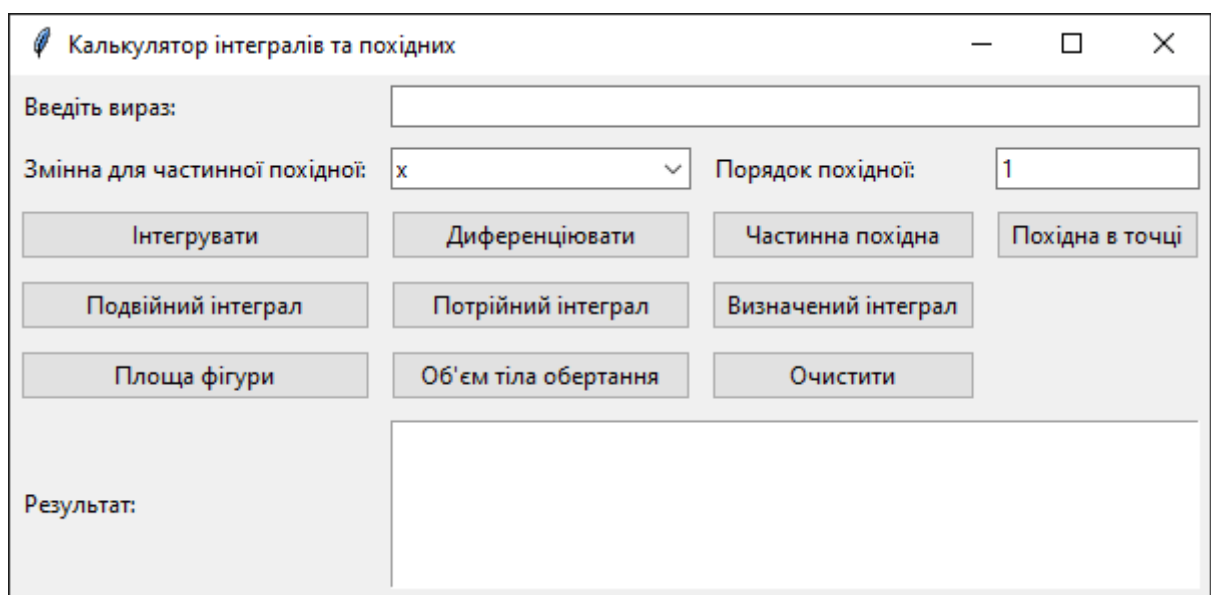


Рисунок 3.20 – Графічний інтерфейс калькулятора інтегралів та похідних

Джерело: розроблено автором

3.3 Результати роботи калькулятора по обчисленню інтегралів та похідних

Для демонстрації роботи калькулятора, варто розглянути приклади різних функцій.

Обчислення невизначених інтегралів.

Задача 1. Обчислити невизначені інтеграли:

а)

$$\int \left(x^2 + 3\cos x + \frac{1}{1 + \frac{x^2}{2}} \right) dx$$

Калькулятор інтегралів та похідних

Введіть вираз:

Змінна для частинної похідної: Порядок похідної:

Інтегрувати Диференціювати Частинна похідна Похідна в точці

Подвійний інтеграл Потрійний інтеграл Визначений інтеграл

Площа фігури Об'єм тіла обертання Очистити

Результат: Інтеграл: $x^3/3 + 3\sin(x) + \sqrt{2} \cdot \operatorname{atan}(\sqrt{2} \cdot x/2)$

Рисунок 3.21 – Результат інтегрування інтегралу а)

Джерело: розроблено автором

б)

$$\int \frac{dx}{\sin x + \cos x}$$

Калькулятор інтегралів та похідних

Введіть вираз:

Змінна для частинної похідної: Порядок похідної:

Інтегрувати Диференціювати Частинна похідна Похідна в точці

Подвійний інтеграл Потрійний інтеграл Визначений інтеграл

Площа фігури Об'єм тіла обертання Очистити

Результат: Інтеграл: $\sqrt{2} \cdot \log(\tan(x/2) - 1 + \sqrt{2})/2 - \sqrt{2} \cdot \log(\tan(x/2) - \sqrt{2} - 1)/2$

Рисунок 3.22 – Результат інтегрування інтегралу б)

Джерело: розроблено автором

в)

$$\int \ln \sqrt{x^2 + 1} dx$$

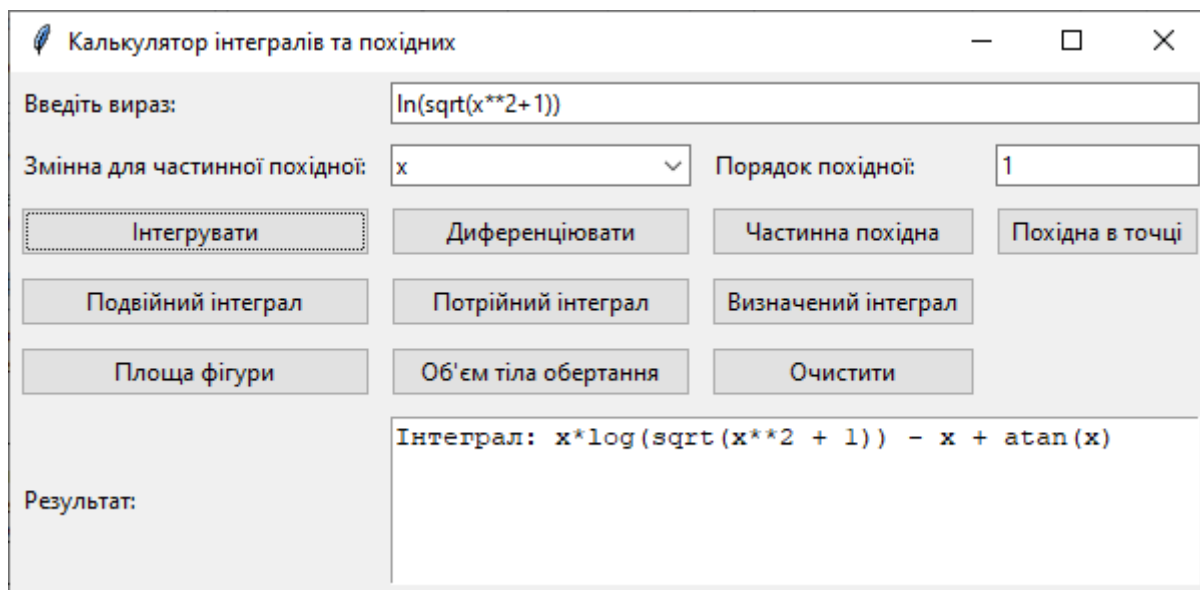


Рисунок 3.23 – Результат інтегрування інтегралу в)

Джерело: розроблено автором

Також, калькулятор може обчислювати інтеграли спеціального виду, первісні яких не можуть бути виражені за допомогою елементарних алгебраїчних функцій (рис. 3.24):

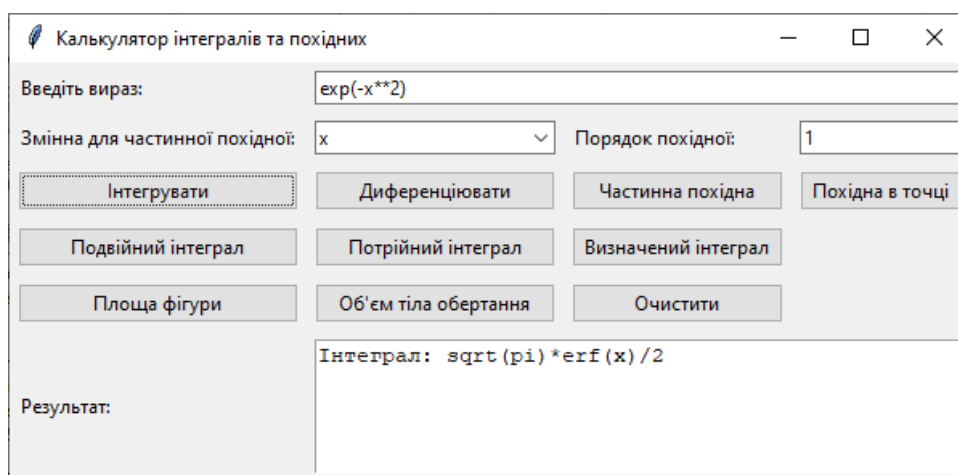


Рисунок 3.24 Результат інтегрування інтегралу спеціального виду

Джерело: розроблено автором

Обчислення визначених інтегралів.

Задача 2. Обчислити визначені інтеграли.

Спочатку вводиться функція (рис. 3.25).

a)

$$\int_0^1 \frac{\sin^5 x}{\cos^4 x} dx$$

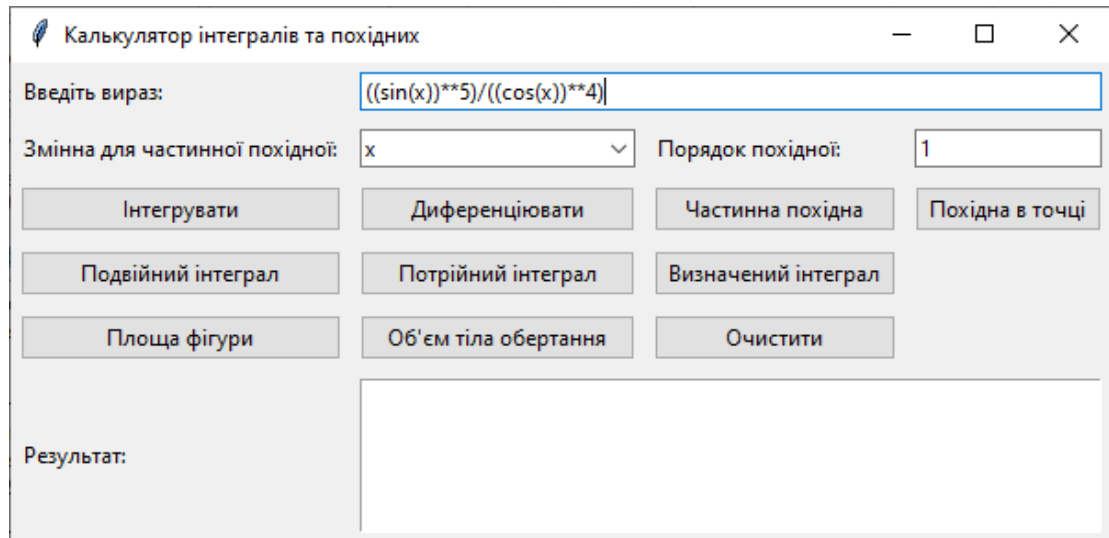


Рисунок 3.25 – Введення функції для обчислення визначеного інтегралу

Джерело: розроблено автором

Після натискання кнопки «Визначений інтеграл» з'являються наступні вікна для введення меж інтегрування (рис. 3.26).

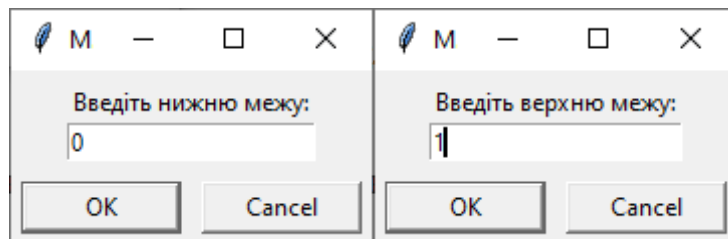


Рисунок 3.26 – Введення меж інтегрування

Джерело: розроблено автором

Результат обчислення (рис. 3.27-3.29).

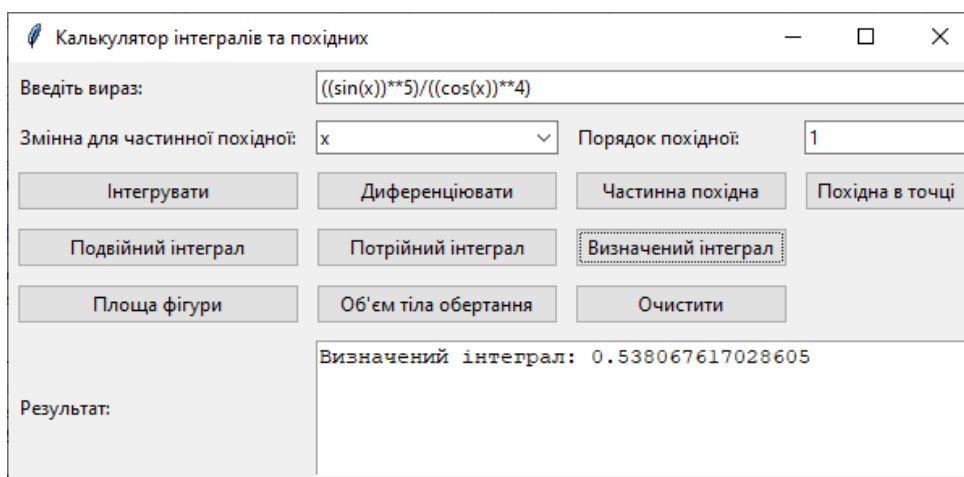


Рисунок 3.27 – Результат обчислення визначеного інтегралу а)

Джерело: розроблено автором

б)

$$\int_0^{\pi} \arcsin\left(\frac{x}{4}\right) dx$$

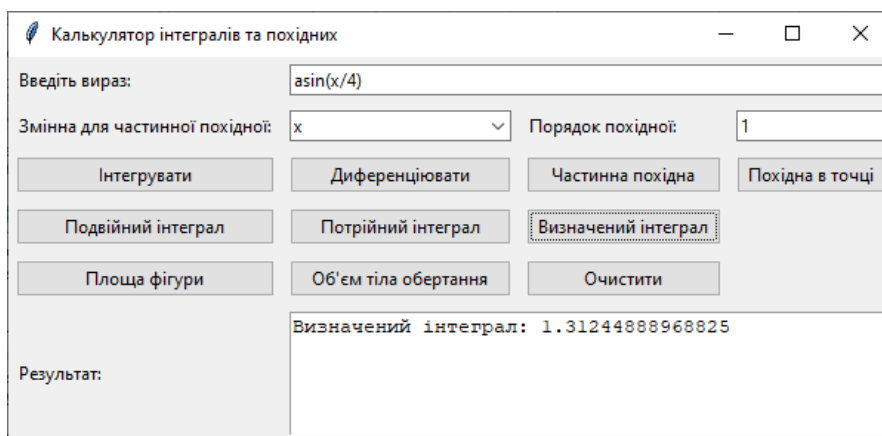


Рисунок 3.28 – Обчислення визначеного інтегралу б)

Джерело: розроблено автором

Обчислення площі фігури, обмеженої кривими.

Задача 3. Обчислити площу фігури, обмеженої заданими кривими:

$$y = x^2 + 2, y = x, x = 0, x = 1$$

Натиснувши на кнопку «Площа фігури», калькулятор запропонує наступні вікна для вводу функцій (рис. 3.29).

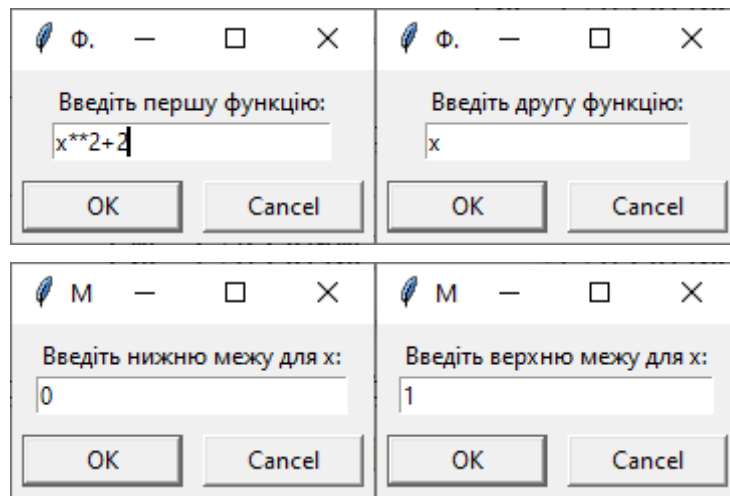


Рисунок 3.29 – Введення даних для обчислення площі фігури

Джерело: розроблено автором

Далі, калькулятор побудує графіки функцій, які обмежують фігуру (рис. 3.30):

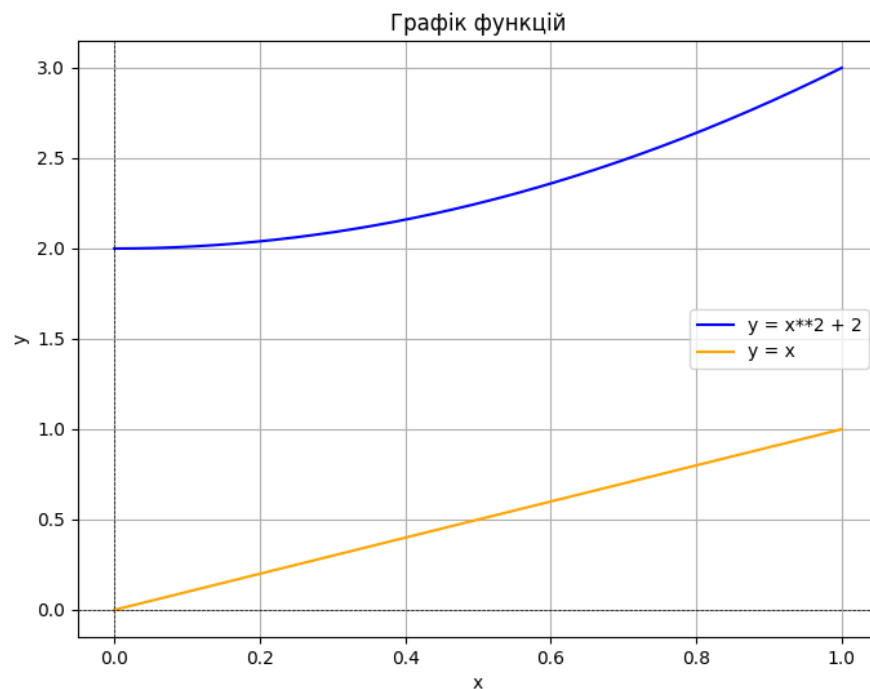


Рисунок 3.30 – Побудова графіків функцій, що обмежують фігуру

Джерело: розроблено автором

Результат обчислення площі (рис. 3.31):

Калькулятор інтегралів та похідних

Введіть вираз:

Змінна для частинної похідної: Порядок похідної:

Інтегрувати Диференціювати Частинна похідна Похідна в точці

Подвійний інтеграл Потрійний інтеграл Визначений інтеграл

Площа фігури Об'єм тіла обертання Очистити

Результат: Площа фігури: 1.83333333333333

Рисунок 3.31 – Результат обчислення площі фігури

Джерело: розроблено автором

Задача 4. Обчислити площу фігури, обмеженої заданими кривими:

$$y = x^2, y = x, x = 0, x = 1$$

Результат обчислення наступний (рис. 3.32):

Калькулятор інтегралів та похідних

Введіть вираз:

Змінна для частинної похідної: Порядок похідної:

Інтегрувати Диференціювати Частинна похідна Похідна в точці

Подвійний інтеграл Потрійний інтеграл Визначений інтеграл

Площа фігури Об'єм тіла обертання Очистити

Результат: Помилка: Площа не може бути від'ємною.

Рисунок 3.32 – Результат обчислення площі фігури

Джерело: розроблено автором

Тобто, якщо результатом обчислення площі є від'ємне значення, калькулятор видасть помилку.

Обчислення об'єму тіла обертання.

Задача 5. Обчислити об'єм тіла обертання $y = x^2, x = 0, x = 1$

При натисканні на кнопку «Об'єм тіла обертання» калькулятор пропонує наступні вікна (рис. 3.33). Результат обчислення (рис. 3.34).

Обчислення подвійних інтегралів.

Задача 6. Обчислити подвійний інтеграл

$$\int_0^1 dx \int_0^2 (x^2 + y) dy$$

Після вводу функції, яку потрібно проінтегрувати, калькулятор пропонує наступні вікна для вводу меж інтегрування (Рисунок 3.35):

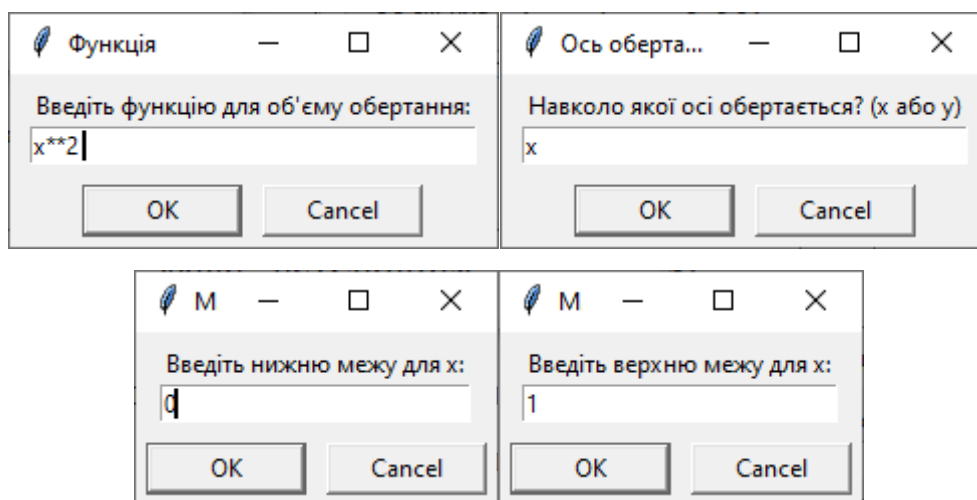


Рисунок 3.33 – Обчислення об'єму тіла обертання

Джерело: розроблено автором

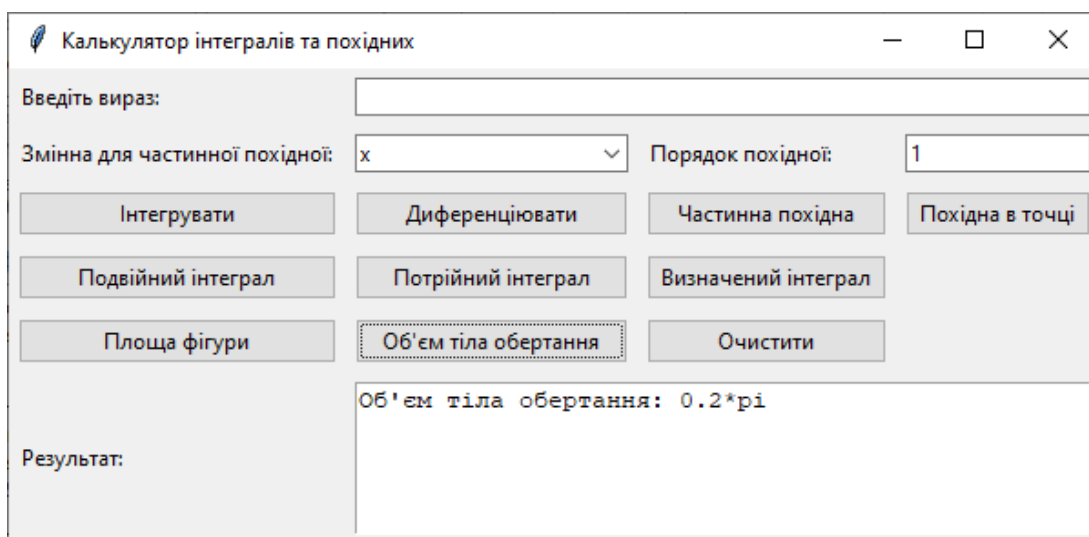


Рисунок 3.34 – Результат обчислення об'єму тіла обертання

Джерело: розроблено автором

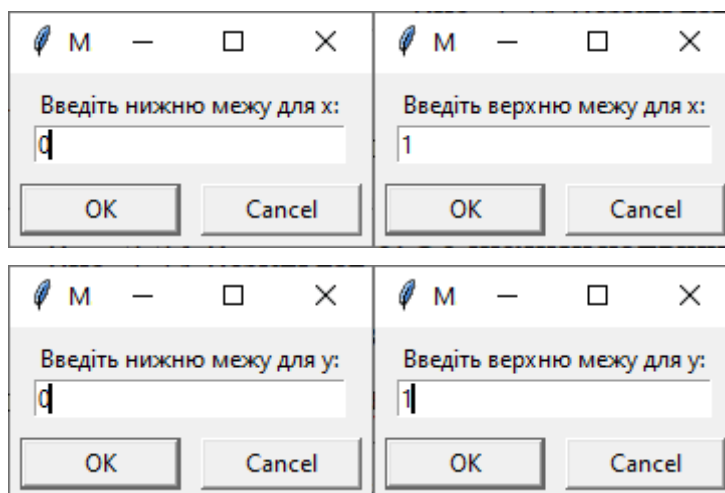


Рисунок 3.35 – Обчислення подвійного інтегралу

Джерело: розроблено автором

Результат інтегрування (рис. 3.36):

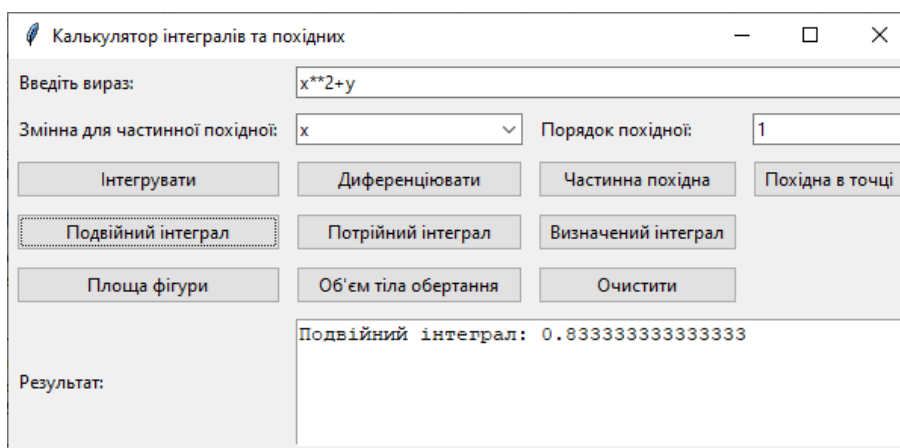


Рисунок 3.36 – Результат обчислення подвійного інтегралу

Джерело: розроблено автором

Обчислення потрійного інтегралу.

Задача 7. Обчислити потрійний інтеграл

$$\int_0^1 dx \int_0^2 dy \int_1^3 (x^3 + e^{xy} + xyz) dz$$

Після вводу функції, яку потрібно проінтегрувати, калькулятор пропонує наступні вікна для вводу меж інтегрування (рис. 3.37). Результат обчислення потрійного інтегралу (рис. 3.38).

Обчислення похідних.

Задача 8. Обчислити похідні наступних функцій:

а)

$$y = e^{\cos x} + \arctg(\arcsin(x)) + \ln\left(\sin \frac{1}{x}\right) - \sqrt{2^{\lg x}}$$

Результат диференціювання (рис. 3.39).

The figure shows six small dialog boxes arranged in a 3x2 grid. Each dialog box has a title bar with a feather icon, a minus sign, a maximize button, and a close button. The dialogs are for setting the lower and upper limits for variables x, y, and z.

- Top-left:** "Введіть нижню межу для x:" (Enter lower limit for x:). Input field contains "0". Buttons: OK, Cancel.
- Top-right:** "Введіть верхню межу для x:" (Enter upper limit for x:). Input field contains "1". Buttons: OK, Cancel.
- Middle-left:** "Введіть нижню межу для y:" (Enter lower limit for y:). Input field contains "0". Buttons: OK, Cancel.
- Middle-right:** "Введіть верхню межу для y:" (Enter upper limit for y:). Input field contains "1". Buttons: OK, Cancel.
- Bottom-left:** "Введіть нижню межу для z:" (Enter lower limit for z:). Input field contains "1". Buttons: OK, Cancel.
- Bottom-right:** "Введіть верхню межу для z:" (Enter upper limit for z:). Input field contains "3". Buttons: OK, Cancel.

Рисунок 3.37 – Обчислення подвійного інтегралу

Джерело: розроблено автором

The screenshot shows a window titled "Калькулятор інтегралів та похідних" (Integral and Derivative Calculator). The interface includes the following elements:

- Введіть вираз:** (Enter expression:). Input field contains $x^3 + \exp(x \cdot y) + x \cdot y \cdot z$.
- Змінна для частинної похідної:** (Variable for partial derivative:). Dropdown menu shows "x".
- Порядок похідної:** (Order of derivative:). Input field contains "1".
- Buttons:**
 - Інтегрувати (Integrate)
 - Диференціювати (Differentiate)
 - Частинна похідна (Partial derivative)
 - Похідна в точці (Derivative at point)
 - Подвійний інтеграл (Double integral)
 - Потрійний інтеграл** (Triple integral) - highlighted with a dashed border
 - Визначений інтеграл (Definite integral)
 - Площа фігури (Area of figure)
 - Об'єм тіла обертання (Volume of solid of revolution)
 - Очистити (Clear)
- Результат:** (Result:). Output field shows: "Потрійний інтеграл: $-2 \cdot \log(2) - 2 \cdot \text{EulerGamma} + 14.9084687120038$ ".

Рисунок 3.38 – Результат обчислення потрійного інтегралу

Джерело: розроблено автором

Калькулятор інтегралів та похідних

Введіть вираз: $\exp(\cos(x)) + \operatorname{atan}(\sin(x)) + \ln(\sin(1/x)) - \sqrt{2 \cdot \tan(x)}$

Змінна для частинної похідної: x Порядок похідної: 1

Інтегрувати Диференціювати Частинна похідна Похідна в точці

Подвійний інтеграл Потрійний інтеграл Визначений інтеграл

Площа фігури Об'єм тіла обертання Очистити

Результат:

Похідна 1-го порядку по x : $-(\tan(x)^2 + 1) \cdot \sqrt{2 \cdot \tan(x)} \cdot \log(2)/2 - \exp(\cos(x)) \cdot \sin(x) + 1/(\sqrt{1 - x^2}) \cdot (\operatorname{asin}(x)^2 + 1) - \cos(1/x) / (x^2 \cdot \sin(1/x))$

Рисунок 3.39 Результат диференціювання а)

Джерело: розроблено автором

Калькулятор інтегралів та похідних

Введіть вираз: $(\tan(x))^{\ln(x)}$

Змінна для частинної похідної: x Порядок похідної: 1

Інтегрувати Диференціювати Частинна похідна Похідна в точці

Подвійний інтеграл Потрійний інтеграл Визначений інтеграл

Площа фігури Об'єм тіла обертання Очистити

Результат:

Похідна 1-го порядку по x : $((\tan(x))^2 + 1) \cdot \log(x) / \tan(x) + \log(\tan(x)) / x \cdot \tan(x)^{\log(x)}$

Рисунок 3.40 – Результат диференціювання б)

Джерело: розроблено автором

б)

$$y = (\operatorname{tg}(x))^{\ln(x)}$$

Результат диференціювання (рис. 3.40):

Обчислення похідних вищого порядку.

Задача 9. Обчислити похідну 5-го порядку функції $y = \cos(\cos(x))$

Результат диференціювання (рис. 3.41):

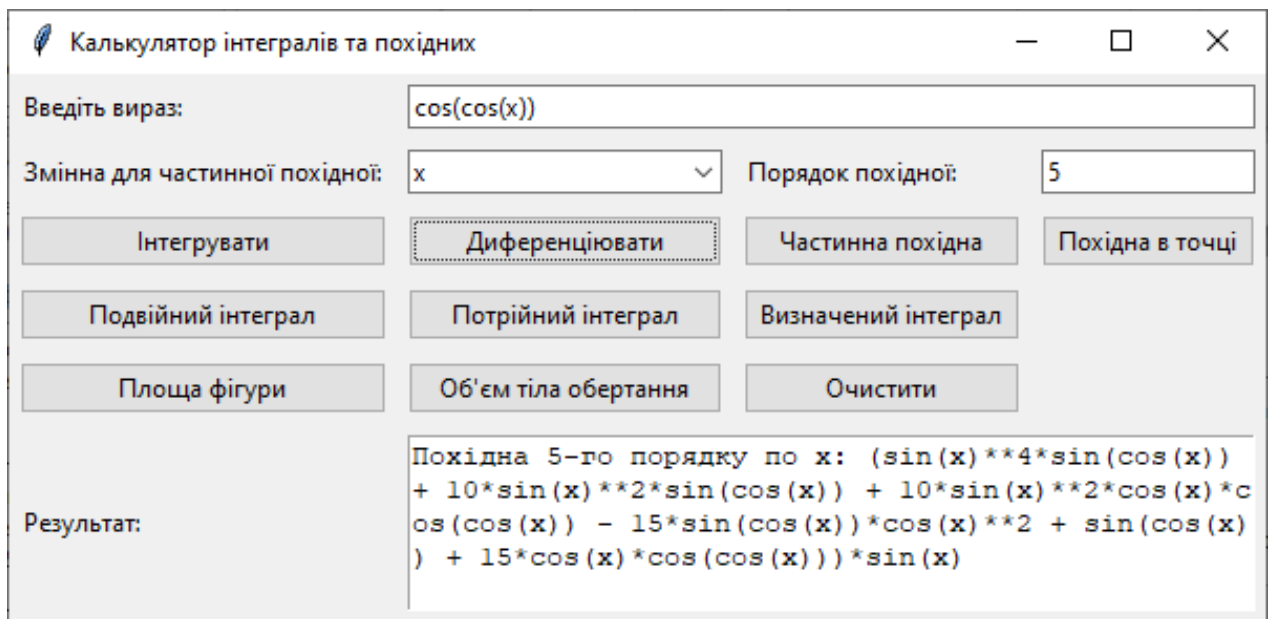


Рисунок 3.41 – Результат обчислення похідної 5-го порядку

Джерело: розроблено автором

Обчислення частинних похідних.

Задача 10. Обчислити частинну похідну наступних функцій:

а)

$$z(x, y) = x^3 + e^{\cos xy}$$

Результат обчислення похідної по змінній «х» (рис. 3.42):

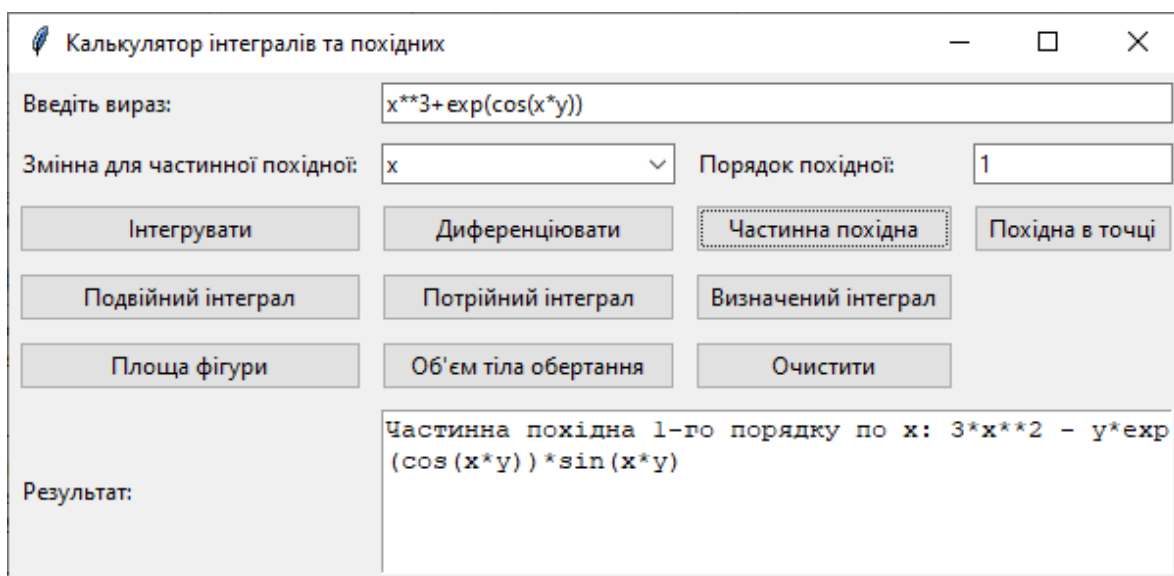


Рисунок 3.42 Обчислення частинної похідної по змінній «x»

Джерело: розроблено автором

Результат обчислення похідної по змінній «y» (рис. 3.43):

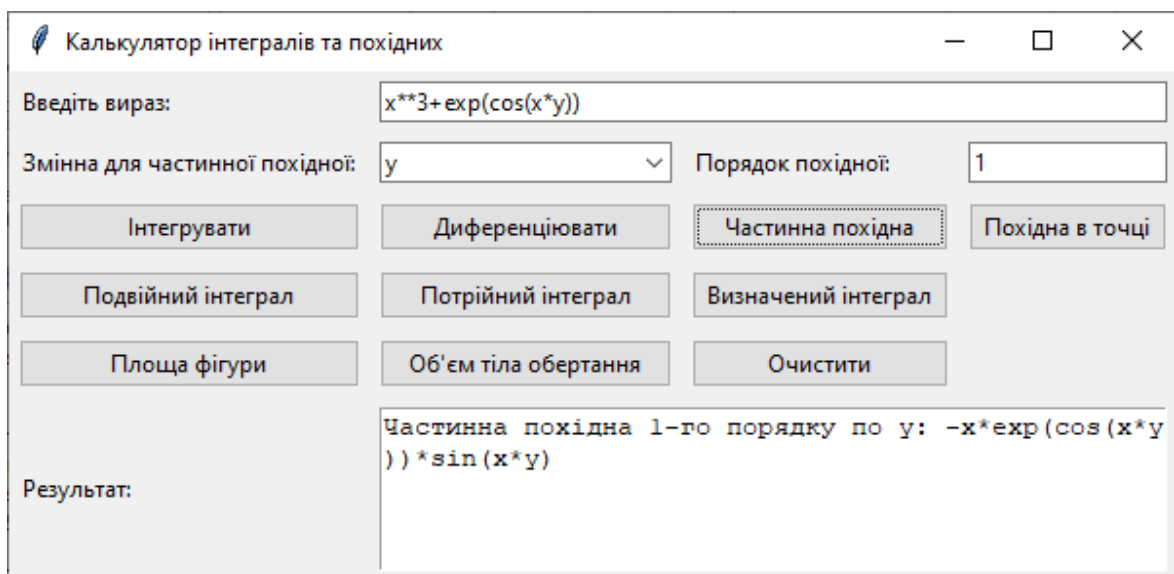


Рисунок 3.43 – Обчислення частинної похідної по змінній «y»

Джерело: розроблено автором

б)

$$u(x, y, z) = xy^2z^5 + \ln \sqrt{x^2 + y^2 + z^2}$$

Результат обчислення похідної по змінній «х» (рис. 3.44):

Калькулятор інтегралів та похідних

Введіть вираз: $x*y^{**2}*z^{**5}+\ln(\text{sqrt}(x^{**2}+y^{**2}+z^{**2}))$

Змінна для частинної похідної: x Порядок похідної: 1

Інтегрувати Диференціювати Частинна похідна Похідна в точці

Подвійний інтеграл Потрійний інтеграл Визначений інтеграл

Площа фігури Об'єм тіла обертання Очистити

Результат: Частинна похідна 1-го порядку по x : $x/(x^{**2} + y^{**2} + z^{**2}) + y^{**2}*z^{**5}$

Рисунок 3.44 Обчислення частинної похідної по змінній «х»

Джерело: розроблено автором

Результат обчислення похідної по змінній «у» (рис. 3.45):

Калькулятор інтегралів та похідних

Введіть вираз: $x*y^{**2}*z^{**5}+\ln(\text{sqrt}(x^{**2}+y^{**2}+z^{**2}))$

Змінна для частинної похідної: y Порядок похідної: 1

Інтегрувати Диференціювати Частинна похідна Похідна в точці

Подвійний інтеграл Потрійний інтеграл Визначений інтеграл

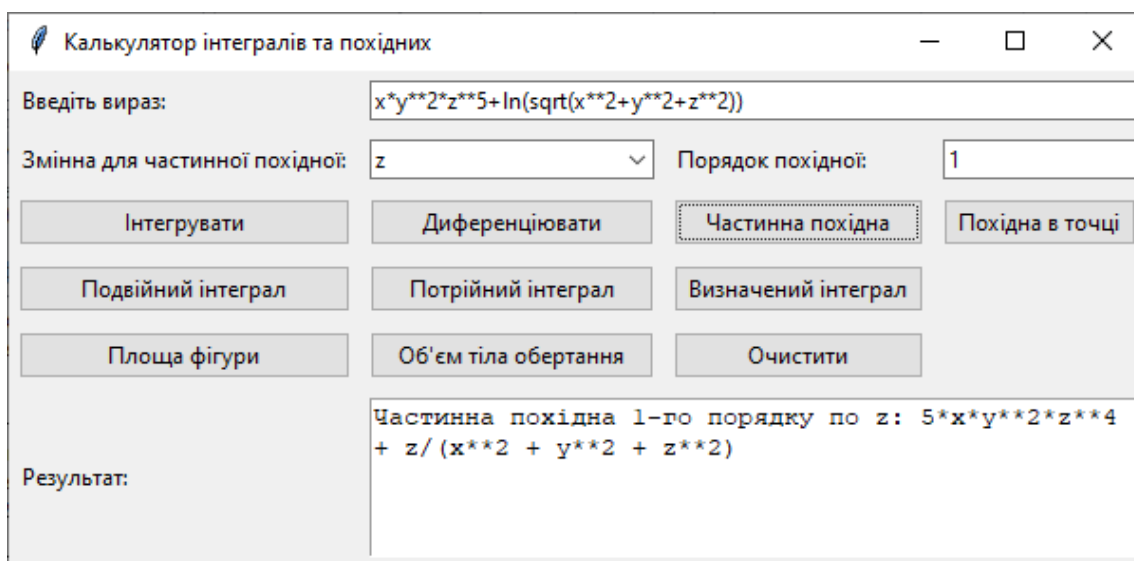
Площа фігури Об'єм тіла обертання Очистити

Результат: Частинна похідна 1-го порядку по y : $2*x*y*z^{**5} + y / (x^{**2} + y^{**2} + z^{**2})$

Рисунок 3.45 – Обчислення частинної похідної по змінній «у»

Джерело: розроблено автором

Результат обчислення похідної по змінній «z» (рис. 3.46):



Калькулятор інтегралів та похідних

Введіть вираз: $x*y^{**2}*z^{**5}+\ln(\sqrt{x^{**2}+y^{**2}+z^{**2}})$

Змінна для частинної похідної: z Порядок похідної: 1

Інтегрувати Диференціювати **Частинна похідна** Похідна в точці

Подвійний інтеграл Потрійний інтеграл Визначений інтеграл

Площа фігури Об'єм тіла обертання Очистити

Результат: Частинна похідна 1-го порядку по z : $5*x*y^{**2}*z^{**4} + z/(x^{**2} + y^{**2} + z^{**2})$

Рисунок 3.46 – Обчислення частинної похідної по змінній «z»

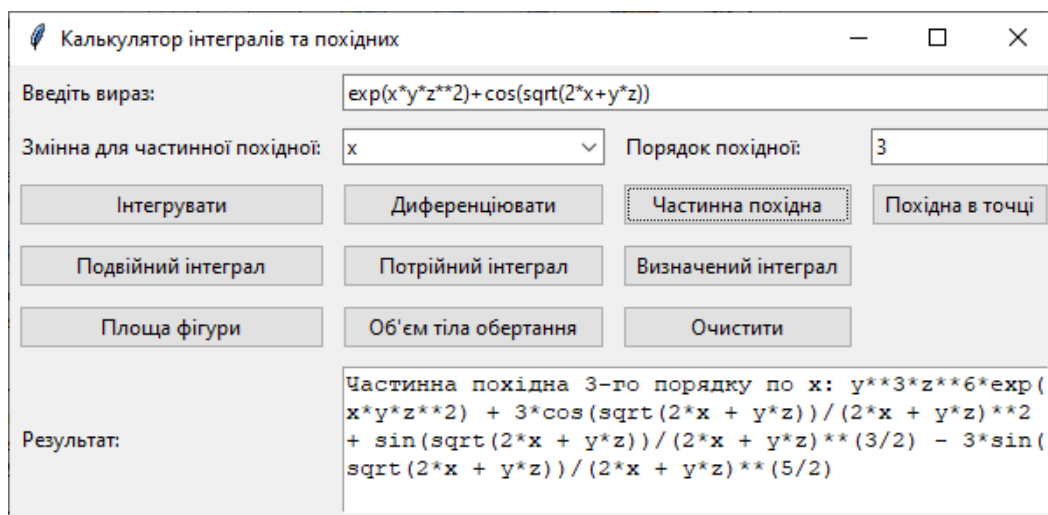
Джерело: розроблено автором

Обчислення частинних похідних вищих порядків.

Задача 11. Обчислити частинну похідну 3 порядку наступної функції

$$u(x, y, z) = e^{xyz^2} + \cos(\sqrt{2x + yz})$$

Результат обчислення похідної по змінній «x» (рис. 3.47):



Калькулятор інтегралів та похідних

Введіть вираз: $\exp(x*y*z^{**2})+\cos(\sqrt{2*x+y*z})$

Змінна для частинної похідної: x Порядок похідної: 3

Інтегрувати Диференціювати **Частинна похідна** Похідна в точці

Подвійний інтеграл Потрійний інтеграл Визначений інтеграл

Площа фігури Об'єм тіла обертання Очистити

Результат: Частинна похідна 3-го порядку по x : $y^{**3}*z^{**6}*\exp(x*y*z^{**2}) + 3*\cos(\sqrt{2*x + y*z})/(2*x + y*z)^{**2} + \sin(\sqrt{2*x + y*z})/(2*x + y*z)^{**(3/2)} - 3*\sin(\sqrt{2*x + y*z})/(2*x + y*z)^{**(5/2)}$

Рисунок 3.47 – Обчислення частинної похідної по змінній «х»

Джерело: розроблено автором

Результат обчислення похідної по змінній «у» (рис. 3.48):

Калькулятор інтегралів та похідних

Введіть вираз: $\exp(x*y*z^{**2})+\cos(\sqrt{2*x+y*z})$

Змінна для частинної похідної: y Порядок похідної: 3

Інтегрувати Диференціювати Частинна похідна Похідна в точці

Подвійний інтеграл Потрійний інтеграл Визначений інтеграл

Площа фігури Об'єм тіла обертання Очистити

Результат:

Частинна похідна 3-го порядку по y : $z^{**3}*(x^{**3}*z^{**3}*\exp(x*y*z^{**2}) + 3*\cos(\sqrt{2*x + y*z}))/ (8*(2*x + y*z)^{**2}) + \sin(\sqrt{2*x + y*z}) / (8*(2*x + y*z)^{** (3/2)}) - 3*\sin(\sqrt{2*x + y*z}) / (8*(2*x + y*z)^{** (5/2)})$

Рисунок 3.48 – Обчислення частинної похідної по змінній «у»

Джерело: розроблено автором

Результат обчислення похідної по змінній «z» (рис. 3.49):

Калькулятор інтегралів та похідних

Введіть вираз: $\exp(x*y*z^{**2})+\cos(\sqrt{2*x+y*z})$

Змінна для частинної похідної: z Порядок похідної: 3

Інтегрувати Диференціювати Частинна похідна Похідна в точці

Подвійний інтеграл Потрійний інтеграл Визначений інтеграл

Площа фігури Об'єм тіла обертання Очистити

Результат:

Частинна похідна 3-го порядку по z : $y^{**2}*(8*x^{**3}*y*z^{**3}*\exp(x*y*z^{**2}) + 12*x^{**2}*z*\exp(x*y*z^{**2}) + 3*y*\cos(\sqrt{2*x + y*z}) / (8*(2*x + y*z)^{**2}) + y*\sin(\sqrt{2*x + y*z}) / (8*(2*x + y*z)^{** (3/2)}) - 3*y*\sin(\sqrt{2*x + y*z}) / (8*(2*x + y*z)^{** (5/2)})$

Рисунок 3.49 – Обчислення частинної похідної по змінній «z»

Джерело: розроблено автором

Обчислення похідної в точці.

Задача 12. Обчислити значення похідної 7-ого порядку в точці $x = 1$ для функції:

$$y = \ln\left(e^{x \cdot 2^x} + \ln\left(\frac{1}{x^3 + 1}\right)\right)$$

Результат обчислення (рис. 3.50):

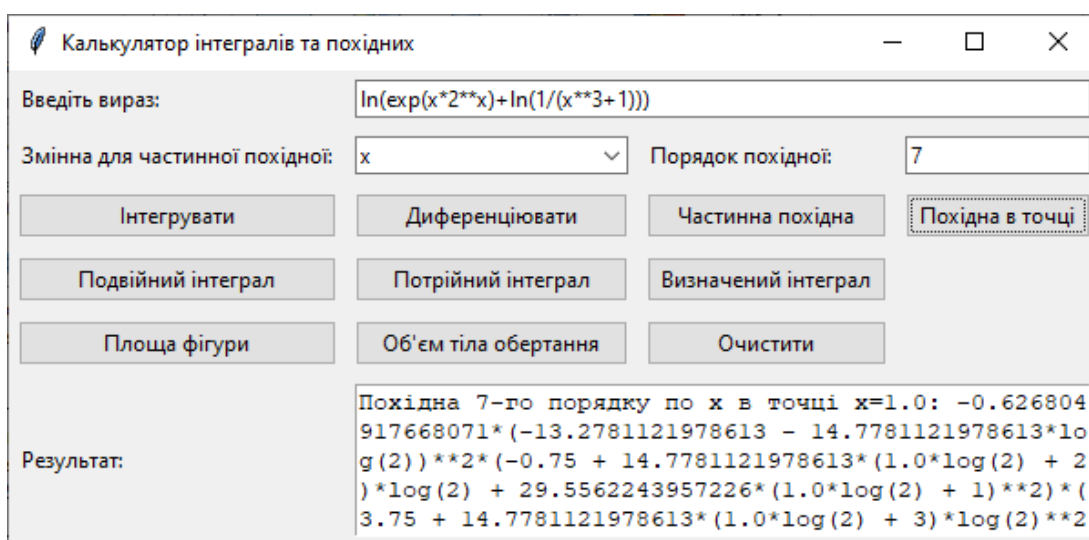


Рисунок 3.50 – Результат обчислення похідної вищого порядку в заданій точці

Джерело: розроблено автором

Висновок до розділу 3

В результаті проведеної роботи було проведено розробку калькулятора для обчислення інтегралів та похідних.

В рамках даного розділу було надано поглиблений опис використовуваних бібліотек, а саме NumPy, SymPy, Matplotlib, приведено фрагменти коду програми, яка реалізує кожну функцію створеного калькулятора, наведено загальний вигляд графічного інтерфейсу, який

дозволяє вводити функції та проводити необхідні обчислення з виведенням кінцевого результату. Представлений калькулятор може обчислювати невизначені та визначені інтеграл, обчислювати площу фігури, обмеженою кривим, обчислювати об'єм тіла обертання навколо координатних осей, обчислювати подвійні та потрійні інтеграл, знаходити похідні функції, причому похідні різних порядків, знаходити частинні похідні, знаходити значення похідних в різних точках.

В результаті тестування, було розглянуто достатню кількість прикладів для демонстрації роботи калькулятора. Калькулятор справляється з досить складними обчисленнями для нетривіальних функцій, що дозволяє йому перевіряти достовірність відповідей, отриманих при аналітичному розв'язку.

ВИСНОВОК

Використання високоточних калькуляторів при розв'язуванні різних практичних задач, є одним із важливих засобів, так як дозволяє значно пришвидшити процес вирішення, провести перевірку раніше отриманих результатів.

За час розвитку комп'ютерних технологій було розроблено досить велику кількість прикладних математичних пакетів, які дозволяють розв'язувати велику кількість задач, які навіть виходять за межі математики. До таких пакетів можна віднести Maple, Matlab, MathCad, Maxima, Wolfram Alpha та інші. Вони мають досить великий та потужний функціонал, що дозволяє досить швидко розв'язувати велику кількість прикладних задач. Кожен з пакетів відрізняється складністю функціоналу, в одних пакетах обчислення виконуються за допомогою команд та коду, в інших, вже є вбудовані функції, які дозволяють проводити швидкі обчислення.

Однак, попри велику кількість прикладних пакетів, існує досить мало рішень, пов'язаних тільки із обчислення інтегралів та похідних, що зумовило актуальність обраної теми.

Для реалізації калькулятора для обчислення інтегралів та похідних було обрано мову програмування Python, так як вона містить достатню кількість бібліотек для здійснення математичних обчислень. Такими бібліотеками є NumPy, SymPy, Matplotlib. Кожна з цих бібліотек відповідає за певні процеси, які сприяють більш швидкому обчисленню інтегралів та похідних.

Реалізація даного калькулятора була підкріплена фрагментами коду для виконання певної математичної дії. До функціоналу створеного калькулятора входить обчислення невизначених та визначених інтегралів, обчислення площі фігури, обмеженої графіками функцій, обчислення об'єму тіла обертання навколо координатної осі, обчислення подвійних та потрійних інтегралів, виведення графіків кривих при обчисленні площі фігури, обчислення похідних, обчислення похідних вищих порядків, обчислення

частинних похідних, обчислення частинних похідних вищих порядків, обчислення похідних в точці.

Подальший розвиток представленого калькулятора можна реалізувати за рахунок збільшення кількості задач, які він може розв'язувати, тобто розширюючи його функціонал. Інтегральне та диференціальне числення є двома фундаментальними розділами математики, які використовуються в багатьох прикладних задачах.

Проведене тестування представленого калькулятора показало його ефективність при знаходженні інтегралів та похідних від досить складних функцій. Даний калькулятор може використовуватись для перевірки вже розв'язаних задач, для більш швидкого знаходження розв'язку задачі, яка потребує багато часу при аналітичному розв'язуванні, для перевірки наявності розв'язку задачі і т.д.

ПЕРЕЛІК ДЖЕРЕЛ ПОСИЛАННЯ

1. Вовкодав Н. І., Овчарук В. О., Ющук І. В. Інформаційні технології: навч. посібник. К.: НУХТ, 2019. 115 с.
2. Дзісь В. Г., Левчук О.В., Новицька Л.І. , Смілянець О.Г., Бубновська.І.А. Довідник . Програмування в Mathcad . – Вінниця. Видавничий центр ВНАУ 2015.– 187 с.
3. Дрозденко В.О. Maple в математиці: навчальний посібник для студентів вищих навчальних закладів III та IV рівнів акредитації / В.О. Дрозденко. – Біла Церква, 2019. – 328 с.
4. Жученко А.І. Оцінювання параметрів та перевірка статистичних гіпотез. Теорія та практика роботи з MathCAD, MatLab, MS Excel: навчальний посібник для студентів вищих навчальних закладів, які навчаються за спеціальністю "Автоматизоване управління технологічними процесами" / А.І. Жученко, Л.Д. Ярошук; М-во освіти і науки, молоді та спорту України, НТУУ "КПІ". – Київ: НТУУ "КПІ", 2012. – 154 с.
5. Златопольський Д.М. Основи програмування мовою Python. 2017. - 284 с.
6. Комп'ютерне моделювання систем та процесів. Методи обчислень. Частина 1 : навчальний посібник / Кветний Р. Н., Богач І. В., Бойко О. Р., Софіна О. Ю., Шушура О.М.; за заг. ред. Р.Н. Кветного. – Вінниця: ВНТУ, 2012. – 193 с
7. Кравченко І. В., Микитенко В. І., Інформаційні технології. Системи комп'ютерної математики: навч. посіб. для студ. спеціальності «Автоматизація та комп'ютерно-інтегровані технології». Київ : КПІ ім. Ігоря Сікорського, 2018. 243с.
8. Кузяєв І.М. Основи комп'ютерного моделювання технічних систем: навчальний посібник / І.М. Кузяєв, Ситар В.І.; Міністерство освіти і науки України, Державний вищий навчальний заклад "Український державний хіміко-технологічний університет". – Дніпро: ДВНЗ УДХТУ, 2020. – 392 с.

9. Любанович Білл Простий Python. Сучасний стиль програмування. 2016. - 480 с.: - (Серія "Бестселлери O'Reilly").

10. Методичні вказівки до лабораторного практикуму з дисципліни „Математичні пакети прикладних програм“ для студентів спеціальностей 125 „Кібербезпека та захист інформації“, освітні програми „Системи технічного захисту інформації, автоматизація її обробки“, „Безпека інформаційних і комунікаційних систем“; 175 „Інформаційно-вимірювальні технології“, освітня програма „Якість, стандартизація та сертифікація“, 175 „Мікро- та наносистемна техніка“ освітня програма „Мікро- та наноелектронні прилади і пристрої“ денної і заочної форм навчання / Укл.: В.І. Рева, Н.І Павлице. – Запоріжжя: НУ «Запорізька політехніка», 2024. – 93 с.

11. Методичні рекомендації до практичних робіт з дисципліни «Сучасні методи математичної обробки інформації» для здобувачів освітньо-наукового ступеня «доктор філософії» зі спеціальності 151 Автоматизація та комп’ютерно-інтегровані технології, освітньо-наукової програми «Автоматизація та комп’ютерно-інтегровані технології» усіх форм навчання. [Електронний ресурс] / [Я.В. Корпань, О.В. Нечипоренко, Є.Є. Федоров]; М-во освіти і науки України, Черкас. держ. технол. ун-т. – Черкаси : ЧДТУ, 2019.

12. Методологія та технологія створення складальних програмних систем: навчально-методичний посібник / Укладачі : Кінах Я.І., Бойко І.В. – Тернопіль : Тернопільський національний технічний університет імені Івана Пулюя, 2017 – 40 с

13. Основи роботи з MathCad: методичні вказівки до виконання комп’ютерного практикуму з дисципліни «Комп’ютерні технології та програмування – 2. Програмні засоби для числового аналізу», для студентів напряму підготовки «Автоматизація та комп’ютерно-інтегровані технології» / Національний технічний університет України «Київський політехнічний інститут»; укладач О.В. Ситніков. – Київ: НТУУ «КПІ», 2013. – 103 с.

14. Основи роботи з Mathcad: Методичні вказівки до виконання лабораторних робіт з курсу "Комп'ютерні технології" / Уклад. Т. В. Аверіна, Л. Д. Ярошук, І. В. Ярошук ; НТУУ "КПІ". – К. : Політехніка, 2004. – 44с
15. Рейтц К., Шлюссер Т. Автостопом з Python. 2017. - 336 с.: іл. – (Серія "Бестселери O'Reilly").
16. Собчук В.В., Чичурін О.В., Кальчук І.В., Жигалло Т.В. Розв'язування задач аналізу та диференціальних рівнянь засобами комп'ютерної алгебри Mathematica. Київ : Міленіум, 2021. – 420 с.
17. Терлецький А.І., Фрик О.Б. Використання математичного пакету Maple для розв'язування та моделювання задач. Методичні рекомендації до виконання лабораторних робіт з дисципліни "Комп'ютерне моделювання та оптимізація" для студентів напрямку "Комп'ютерна інженерія". - Івано-Франківськ, 2012. - 78 с.
18. Федоров, Д. Ю. Програмування мовою високого рівня Python: навчальний посібник для прикладного бакалаврату [Електронний ресурс]. - URL: <https://urait.com/bcode/437489>
19. Demana, F., & Waits, B. K. (2000). Calculators in mathematics teaching and learning. Past, present, and future. In Learning Mathematics for a New Century, 51-66.
20. Emmanuel Kwadzo Sallah. Use of Maple Software to Reduce Senior High School Students' Errors in Integral Calculus.- African Journal of Educational Studies in Mathematics and Sciences Vol. 17, No. 2. 2021.
21. Irvan, I. (2024). Application of Integrals in Calculating Ball Volume using GeoGebra. Indonesian Journal of Education and Mathematical Science, 5(1), 58-63.
22. Maple [Електронний ресурс]. Режим доступу: <http://www.maplesoft.com/products/Maple/index.aspx>
23. MathCAD [Електронний ресурс]. Режим доступу: <http://www.ptc.com/products/mathcad>.
24. Mathematica [Електронний ресурс]. Режим доступу: <http://www.wolfram.com/products/mathematica/index.html>.

25. Matlab [Электронный ресурс]. Режим доступа:
<http://www.mathworks.com/products/matlab/>