

Дніпровський науковий ліцей інформаційних технологій
Дніпровської міської ради

ВИПУСКНА РОБОТА

на тему:

“Дослідження графіків функцій за допомогою похідної”

Виконала учениця 11-Г класу,

Кавуненко Юлія Сергіївна

Керівники роботи:

Боровик Людмила Іванівна,

Якименко Наталія Михайлівна

ЗМІСТ

ВСТУП.....	2
ЯК ПРАВИЛЬНО ЗАПУСТИТИ ПРОЄКТ	6
Кроки для запуску проєкту	6
ТЕОРЕТИЧНА ЧАСТИНА З МАТЕМАТИКИ.....	9
Що таке похідна?	9
Роль першої похідної y' в дослідженні графіків функцій.....	10
Роль першої похідної y'' в дослідженні графіків функцій.....	11
Алгоритм побудови графіку і його дослідження	12
Алгоритм побудови похідних функцій і виконання дослідження	14
Приклади дослідження функцій за допомогою похідних.....	17
ТЕОРЕТИЧНА ЧАСТИНА З ІНФОРМАТИКИ	19
Використані технології.....	19
Архітектура програми.....	20
Алгоритм роботи програми	24
ВИСНОВКИ	28
ДЖЕРЕЛА	29

ВСТУП

У сучасному світі графіки функцій використовуються не лише як інструмент для розв'язання математичних задач, але й як потужний засіб аналізу різноманітних явищ у науках. Використання похідної дозволяє більше зрозуміти, як змінюється функція, та визначити ключові характеристики її графіка, зокрема екстремуми, точки перетину з координатними осями та інші його особливості.

Мета проєкту полягає в дослідженні особливостей графіків квадратичних і дробово-раціональних функцій, з'ясуванні залежності між їх параметрами та характером графіка. Такий підхід дозволяє краще зрозуміти саму поведінку функцій, визначити проміжки зростання і спадання, положення екстремумів, а також побачити вплив коефіцієнтів на вигляд графіка.

Актуальність дослідження полягає в тому, що побудова та аналіз графіків функцій за допомогою похідних є невід'ємною частиною вивчення математики та інших природничих наук. Знання про властивості функцій допомагають не лише у навчанні, але й у практичних сферах, таких як прогнозування, оптимізація та моделювання.

Розробка алгоритмів для аналізу функцій стала основою для реалізації програмного забезпечення, яке демонструє прикладне значення похідних у реальних задачах. Зокрема, користувач може досліджувати, як зміна параметрів функції впливає на її графік, бачити залежність між коефіцієнтами рівнянь і особливостями графіка (екстремуми, симетрія, проміжки спадання).

У рамках проєкту було створено програму на мові Python, яка поєднує в собі зручність інтерфейсу та потужні математичні можливості. Програма використовує такі бібліотеки:

- **NumPy, SymPy** – для виконання математичних розрахунків;
- **Matplotlib, pyplot** – для побудови графіків;
- **Customtkinter, Tkinter** – для створення графічного інтерфейсу;
- **PIL** – для роботи із зображеннями;
- **fitz** - для відображення PDF-файлів;

Програма дозволяє користувачеві не лише будувати графіки функцій, а й виконувати їх аналіз. Наприклад, можна повністю дослідити запропонований графік функції, а саме визначити:

- область визначення функції;
- парність функції;
- проміжки спадання і зростання функції;
- локальні максимум і мінімум функції;
- мінімальне і максимальне значення функції;
- точки перетину з осями;
- нулі функції;
- проміжки знакосталості;
- точки перегину;
- проміжки опуклості;
- асимптоти;

Окрім цього, користувач може спокійно вибирати різні функції у спеціальному вікні та навіть приховувати або відображати певні елементи графіка, використовуючи Checkbox:

- відображення похідної y'
- відображення похідної y''

Додаток проєкту

Сам проєкт складається з 4 вікон:

- Титульне вікно – вікно з темою проєкту, іменем автора і його керівників роботи. З нього можна перейти до вікна додатку і вікна документації;
- Вікно основної програми – можна назвати основним вікном, де виконуються побудови графіків та їх дослідження;
- Вікно документації – на цьому вікні можна ознайомитися з посібником користувача, документацією роботи і додатками до неї;
- Вікно вибору функцій для роботи – другорядне вікно основної програми для вибору функцій, їх побудови і дослідження за допомогою програми.

Випускна робота. Титульна сторінка

Дніпровський науковий ліцей інформаційних технологій Дніпровської міської ради

ВИПУСКНА РОБОТА
на тему:
"Дослідження графіків функцій за допомогою похідної"

Перейти до роботи

Перейти до документації

Виконала учениця 11-Г класу,
Кавуненко Юлія Сергіївна
Керівники роботи –
Боровик Людмила Іванівна,
Якименко Наталія Михайлівна

м. Дніпро
2025

Випускна робота. Документація

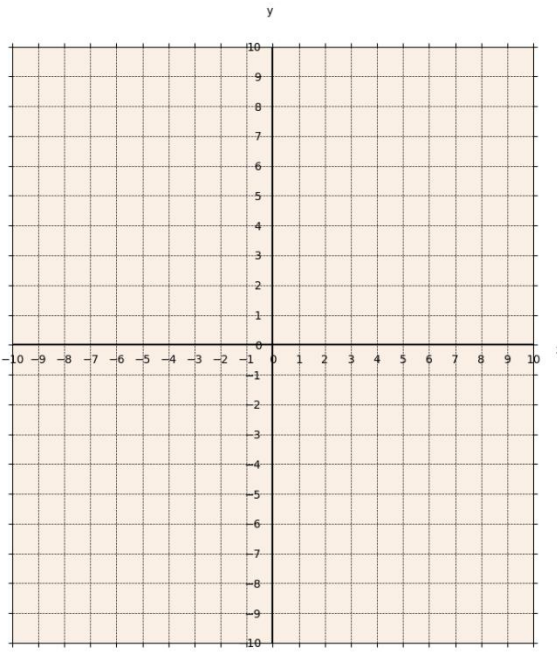
Документація до проєкту

Посібник користувача

Документація

Додаток

Випускна робота. Додаток



у

10 9 8 7 6 5 4 3 2 1 0 -1 -2 -3 -4 -5 -6 -7 -8 -9 -10

10 9 8 7 6 5 4 3 2 1 0 -1 -2 -3 -4 -5 -6 -7 -8 -9 -10

х

ОЧИСТИТИ

у =

Графік для побудови не обрано!

Обрати функцію

Дослідження графіку

- 1) Область визначення функції
- 2) Парна чи непарна ф-ція
- 3) Проміжок спадання і зростання функції
- 4) Локальний макс. і мін. функції
- 5) Мін. і макс. значення функції
- 6) Точки перетину з осями ox і oy
- 7) Нулі функції
- 8) Проміжки знакосталості ф-ції
- 9) Точки перегину
- 10) Проміжки опуклості
- 11) Похила асимптота

Вікно вибору функції для дослідження

$y = ax^3 + bx^2 + cx + d$

$y = \frac{x^2 + a}{x^2 - a}$

$y = \frac{a}{x^2} + \frac{x}{a}$

$y = \frac{x^2 + x + a}{x}$

$y = \frac{x^2 - a}{x - b}$

$y = \frac{x^2 - a^2}{x}$

$y = \frac{x}{x^2 + a}$

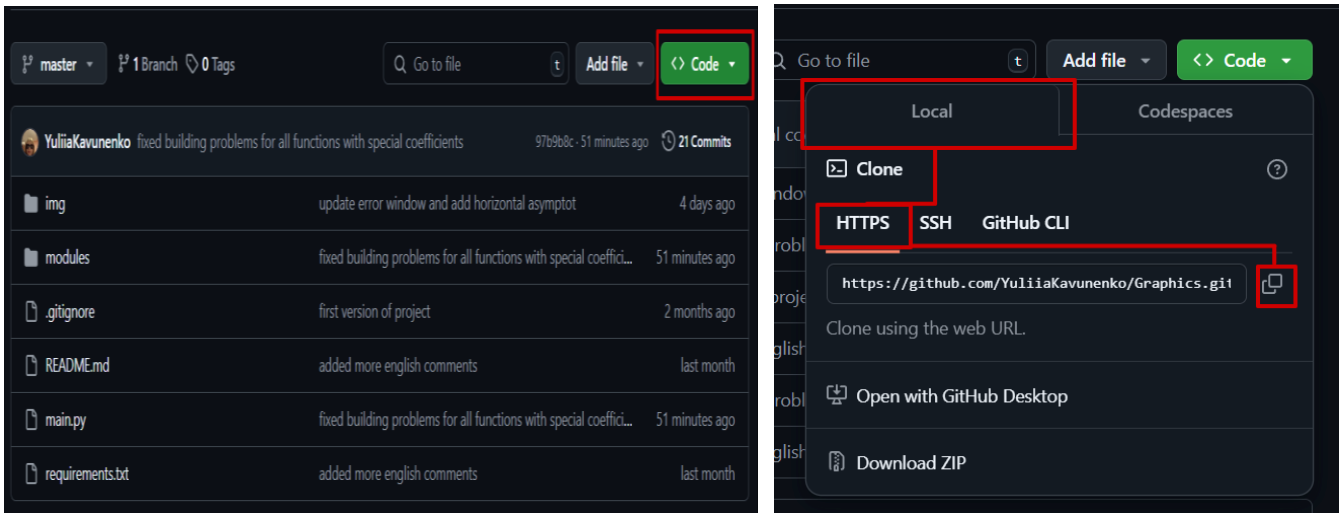
Проект має прикладну цінність, оскільки може використовуватись як навчальний інструмент для вивчення властивостей функцій. Завдяки його застосуванню учні або студенти можуть швидко побачити, як змінюються графіки при зміні параметрів, що робить процес навчання більш наочним і зрозумілим. У дослідженні було застосовано інтеграцію теоретичних знань із практичною реалізацією. Це дозволило отримати універсальний інструмент для дослідження функцій, який є корисним у навчальному процесі та може знайти застосування у реальних задачах.

ЯК ПРАВИЛЬНО ЗАПУСТИТИ ПРОЄКТ

Кроки для запуску проєкту

Проект розташований на публічному git-репозиторії, який можна знайти за посиланням: <https://github.com/YuliiaKavunenko/Graphics>. Для запуску проєкту з git-репозиторію потрібно виконати наступні кроки:

- Натиснути на кнопку «Code» і скопіювати HTTPS посилання



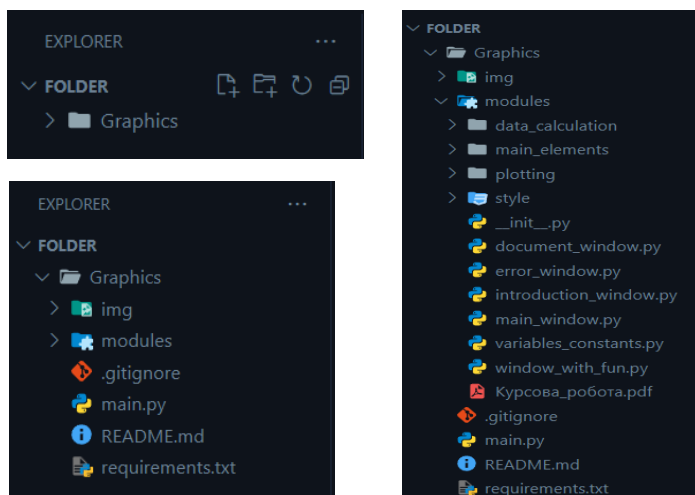
- Відкрийте відповідну папку для копіювання репозиторію у програмі для роботи з кодом (наприклад, VS code) і відкрийте термінал. Для копіювання проєкту у нашу папку потрібно використати команду **git clone посилання**. Тобто:

```
PROBLEMS  OUTPUT  DEBUG CONSOLE  TERMINAL  PORTS  COMMENTS
PS C:\Users\crazy\Downloads\Telegram Desktop\дз лицей 2024\Folder> git clone https://github.com/YuliiaKavunenko/Graphics.git
```

Після успішного копіювання ви маєте побачити наступне повідомлення:

```
PS C:\Users\crazy\Downloads\Telegram Desktop\дз лицей 2024\Folder> git clone https://github.com/YuliiaKavunenko/Graphics.git
Cloning into 'Graphics'...
remote: Enumerating objects: 343, done.
remote: Counting objects: 100% (343/343), done.
remote: Compressing objects: 100% (208/208), done.
remote: Total 343 (delta 244), reused 227 (delta 133), pack-reused 0 (from 0)
Receiving objects: 100% (343/343), 6.15 MiB | 1.74 MiB/s, done.
Resolving deltas: 100% (244/244), done.
PS C:\Users\crazy\Downloads\Telegram Desktop\дз лицей 2024\Folder> █
```

І створену папку з кодом:



- У проєкті було використано декілька модулів, яких, скоріш за все, немає на вашому пристрої. Для якісної роботи потрібно **створити віртуальну середу**. Для її створення необхідно вписати наступну команду у терміналі для:

- **Windows:**

- > `python -m venv назва_віртуального_середовища`

- **Linux/MacOS:**

- > `python3 -m venv venv`

Після успішного створення віртуального оточення ви побачите в explorer нову папку. Для його активації введіть наступні команди для:

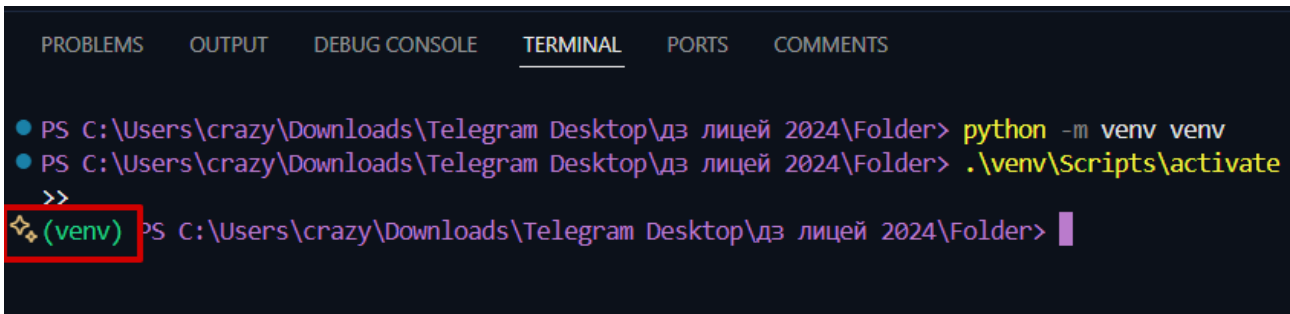
- **Windows:**

- > `.\venv\Scripts\activate`

- **Linux/MacOS:**

- > `source venv/bin/activate`

Після успішної активації ви можете побачити ваше віртуальне оточення біля директорії вашого проєкту в терміналі:

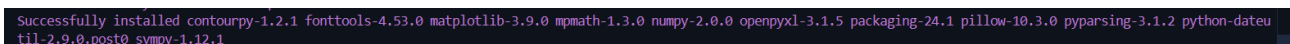


```
PROBLEMS  OUTPUT  DEBUG CONSOLE  TERMINAL  PORTS  COMMENTS
• PS C:\Users\crazy\Downloads\Telegram Desktop\дз лицей 2024\Folder> python -m venv venv
• PS C:\Users\crazy\Downloads\Telegram Desktop\дз лицей 2024\Folder> .\venv\Scripts\activate
>>
(venv) PS C:\Users\crazy\Downloads\Telegram Desktop\дз лицей 2024\Folder> |
```

- Наступним кроком буде завантаження необхідних бібліотек для роботи програми до віртуального оточення. Для цього у проєкті є файл `requirements.txt`, де зберігається лист усіх необхідних нам бібліотек з їх версіями. Далі потрібно перейти до папки скопійованого проєкту за допомогою команди `cd ./Graphics/` і ввести наступну команду у терміналі:

`pip install -r requirements.txt`

Після успішного завантаження ви маєте побачити таке повідомлення у терміналі:



```
Successfully installed contourpy-1.2.1 fonttools-4.53.0 matplotlib-3.9.0 mpmath-1.3.0 numpy-2.0.0 openpyxl-3.1.5 packaging-24.1 pillow-10.3.0 pyparsing-3.1.2 python-dateutil-2.9.0.post0 sympy-1.12.1
```

- Тепер ми можемо спокійно запустити програму у роботу. Для цього можна виконати наступний крок:

- запустити код у файлі `main.py` за допомогою термінала:

`>python main.py run`

Вітаємо! Ви запустили проєкт і можете перейти до користування!

ТЕОРЕТИЧНА ЧАСТИНА З МАТЕМАТИКИ

Основи аналізу функцій. Приклади

Що таке похідна?

Похідною функції f у точці x_0 називають число, яке дорівнює границі відношення приросту функції f у точці x_0 до відповідного приросту аргументу за умови, що приріст аргументу прямує до нуля. Похідну функції $y = f(x)$ у точці x_0 позначають так: $f'(x_0)$ (читають: «еф штрих від ікс нульового») або $y'(x_0)$.

Можна записати:

$$f'(x_0) = \lim_{\Delta x \rightarrow 0} \frac{f(x_0 + \Delta x) - f(x_0)}{\Delta x}$$

Похідна функції в точці має **важливе геометричне значення**. Вона визначає кутовий коефіцієнт дотичної до графіка функції в цій точці. Іншими словами, значення похідної показує швидкість зміни функції в околі конкретної точки.

Геометрично, якщо провести дотичну до графіка функції $y = f(x)$ у точці $(x_0, f(x_0))$, то похідна $f'(x_0)$ буде дорівнювати тангенсу кута нахилу цієї дотичної до осі абсцис. Наприклад, якщо похідна додатна, тангенс кута нахилу дотичної додатний і функція на цьому інтервалі зростає. **Якщо похідна від'ємна, тангенс кута нахилу дотичної від'ємний, що вказує на спадання функції. Якщо похідна дорівнює нулю, це означає, що дотична паралельна осі ox . Критична точка може бути точкою екстремуму (мінімумом або максимумом), якщо похідна при переході через цю точку змінює знак.**

Таким чином, перша похідна дає змогу досліджувати поведінку функції, її зростання чи спадання, а також знаходити ключові точки на графіку, які є важливими для аналізу функції та її властивостей.

Таблиця похідних

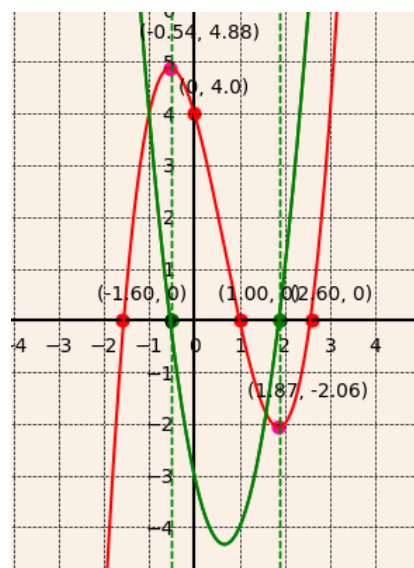
№	Функція	Похідна
1	$y = C, C - \text{стала}$	$y' = 0$
2	$y = x^n, n \in Z$	$y' = nx^{n-1}$
3	$y = \sqrt{x}$	$y' = \frac{1}{2\sqrt{x}}$
4	$y = \sin x$	$y' = \cos x$
5	$y = \cos x$	$y' = -\sin x$
6	$y = tg x$	$y' = \frac{1}{\cos^2 x}$
7	$y = ctg x$	$y' = -\frac{1}{\sin^2 x}$
8	$y = e^x$	$y' = e^x$
9	$y = a^x$	$y' = a^x * \ln a$
10	$y = \log_a x$	$y' = \frac{1}{x * \ln a}$
11	$y = \ln(x)$	$y' = \frac{1}{x}$

Роль першої похідної y' в дослідженні графіків функцій

Роль похідної полягає не лише в обчисленні її значень, а й у вивченні характеру графіка функції. Основними аспектами, що досліджуються за допомогою похідних, є **визначення точок екстремуму (локальних максимумів та локальних мінімумів), дослідження проміжків зростання і спадання.**

1. В першу чергу, похідна функції дає змогу визначити **швидкість її зміни в будь-якій точці графіка**. Якщо похідна функції в точці є невід'ємною, це означає, що функція зростає, якщо ж недодатною — функція спадає.

Власне, з цих знань і починається вивчення проміжків зростання та спадання функції, що є важливим етапом у побудові її графіка.

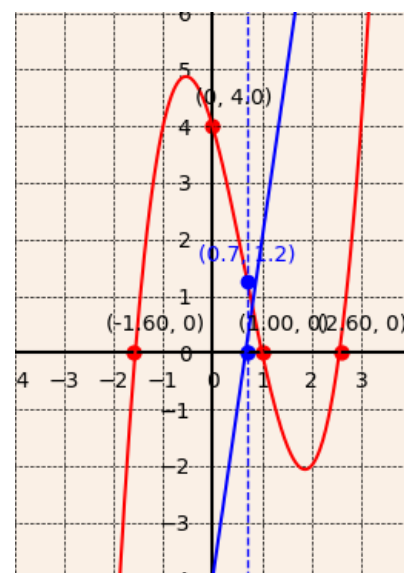


Графіки функцій
 $y = f(x)$ і $y = f'(x)$

2. Похідна також дозволяє виявляти **критичні точки**, де функція може мати екстремуми. При цьому, точка екстремуму є важливою не лише для математичного аналізу, а й для практичних застосувань, де ці точки можуть вказувати на **мінімальні або максимальні значення функцій** у реальних задачах, таких як оптимізація виробництва, фізичні процеси, економічні моделі тощо.

Роль другої похідної y'' в дослідженні графіків функцій

1. Друга похідна функції є важливим інструментом для визначення **точок перегину** та **проміжків опуклості** графіка функції. Якщо перша похідна функції дає інформацію про проміжки спадання і зростання функції, то друга похідна вказує на його **кривизну**, тобто, чи є графік опуклим вгору або вниз на певному проміжку. **Точки перегину** визначаються за допомогою другої похідної функції. **Точка перегину** — це точка, в якій функція змінює свою кривизну, тобто з опуклої вгору на опуклу вниз, або навпаки. Формально, точка перегину існує, коли друга похідна функції в



Графіки функцій
 $y = f(x)$ і $y = f''(x)$

цій точці дорівнює нулю і при переході через цю точку змінює знак. Тобто, якщо $f''(x_0) = 0$ і $f''(x)$ змінює знак при наближенні до x_0 , то **точка x_0 є точкою перегину**.

2. **Проміжки опуклості вгору та вниз** визначаються залежно від знака другої похідної. Якщо друга похідна функції додатна ($f''(x) > 0$) на певному проміжку, то графік функції на цьому проміжку є **опуклим вниз** (загинається "вгору"). Якщо ж друга похідна функції від'ємна ($f''(x) < 0$), графік на цьому проміжку є **опуклим вгору** (загинається "вниз").

Алгоритм побудови графіку і його дослідження

Побудова графіка функції є ключовим етапом у вивченні її властивостей. Вона дозволяє не лише наочно уявити, як функція змінюється при різних значеннях аргументу, а й виявити її характерні особливості, такі як критичні точки, асимптоти, проміжки зростання та спадання, тощо. Алгоритм побудови графіка включає кілька етапів дослідження функції:

1. Перед побудовою графіка важливо визначити **область визначення функції** — множину усіх допустимих значень змінної x . Це дозволяє правильно встановити межі, в яких буде здійснюватися побудова графіка, та уникнути непотрібних асимптотичних або обмежених значень.
2. Однією з важливих властивостей функції є її **парність чи непарність**.

Парна функція – функція, у якої для будь-якого $x \in D(f)$ $\exists (-x) \in D(f)$ є правильною рівністю $f(-x) = f(x)$.

Непарна функція – функція, у якої для будь-якого $x \in D(f)$ $\exists (-x) \in D(f)$ є правильною рівністю $f(-x) = -f(x)$.

3. Аналізуючи першу похідну функції, можна визначити **проміжки зростання та спадання**. Якщо похідна функції невід'ємна на певному проміжку, функція зростає на цьому проміжку, і навпаки, якщо похідна недодатна — функція спадає.

4. **Локальні максимуми та мінімуми** — це точки, де функція досягає найбільшого або найменшого значення в певному околі. Локальний максимум або мінімум можна знайти за допомогою критичних точок, де похідна дорівнює нулю або не існує.
5. **Максимальне значення функції** на певному проміжку визначається як найбільше значення функції на цьому проміжку, а **мінімальне** — найменше.
6. **Нулі функції** (x_1, x_2, x_n) - значення аргументу, при яких значення функції дорівнює нулю. Щоб знайти нулі функції, потрібно розв'язати рівняння: $y = f(x) = 0$. Нулі функції належать $D(y)$.
7. **Точки перетину функції з осями абсцис і ординат** ($x_0; y_0$) — це точки, що мають відповідні координати: $(x_0; 0)$ і $(0; y_0)$.
8. **Проміжок знакосталості** — множина значень x , на якому функція набуває значень однакового знаку.
9. **Точки перегину** — це абсциси точок, де функція змінює свою кривизну. Визначаються за допомогою другої похідної, а саме: якщо **друга похідна дорівнює нулю** і при переході через цю точку змінює знак, то дана точка є точкою перегину.
10. **Проміжки опуклості** функції визначаються за допомогою **другої похідної**. Якщо друга похідна додатна, графік **опуклий вниз**, якщо від'ємна — **опуклий вгору**. Це дозволяє зрозуміти характер кривизни функції.
11. **Асимптота функції** — це пряма, до якої значення функції наближаються необмежено близько при зростанні абсциси або ординати. Асимптоти можуть бути **вертикальними** або **похилими (горизонтальними)**.

Вертикальні асимптоти

Графік функції $y = f(x)$ при $x \rightarrow a$ має вертикальну асимптоту, якщо:

$$\lim_{x \rightarrow a} f(x) = \pm\infty;$$

Крім цього точка $x = a$ є точкою розриву II роду, а рівняння вертикальної асимптоти має вигляд $x = a$.

Похилі асимптоти

Рівняння похилої асимптоти задається рівнянням прямої:

$$y = kx + b,$$

де кутовий коефіцієнт k та вільний член b обчислюються за правилом:

$$k = \lim_{x \rightarrow \pm\infty} \frac{f(x)}{x};$$

$$b = \lim_{x \rightarrow \pm\infty} (f(x) - kx);$$

Якщо обидві границі існують і скінченні, то функція має похилу асимптоту, інакше – не має. Іноді слід окремо розглядати випадки:

$x \rightarrow +\infty$ та $x \rightarrow -\infty$.

Горизонтальні асимптоти

Функція $y = f(x)$ має горизонтальну асимптоту $y = b$, якщо

$$\lim_{x \rightarrow \pm\infty} f(x) = b.$$

Алгоритм побудови похідних функцій і виконання дослідження

Побудова похідної функції є важливим етапом аналізу поведінки функції. Похідна функції дозволяє дослідити такі характеристики, як швидкість зміни функції, зростання чи спадання, наявність екстремумів, а також інші важливі властивості, що визначають форму графіка.

Алгоритм побудови похідної функції та подальше дослідження включає кілька етапів:

1. Знаходження похідної функції

Першим етапом є безпосереднє обчислення похідної функції $f'(x)$. Це здійснюється за допомогою основних правил диференціювання.

Правила диференціювання

Правило	Формула
Правило суми та різниці	$(f(x) \pm g(x))' = f'(x) \pm g'(x)$
Правило добутку	$(f(x) * g(x))' = f'(x) * g(x) + f(x) * g'(x)$
Правило частки	$\left(\frac{f(x)}{g(x)}\right)' = \frac{f'(x)*g(x) - f(x)*g'(x)}{g(x)^2}$
Похідна складеної функції	$(f(g(x)))' = f'(g(x)) * g'(x)$

2. Після того, як похідна функції знайдена, необхідно знайти **критичні точки**, де похідна дорівнює нулю або не існує (якщо такі є). Це важливо для подальшого аналізу екстремумів функції. Точки, де $f'(x) = 0$ або похідна не існує, можуть бути **точками максимуму, мінімуму**.

Інтервали зростання та спадання функції

3. На основі знаків похідної можна визначити проміжки зростання та спадання функції:
- Якщо $f'(x) \geq 0$ на певному проміжку, **функція зростає** на цьому проміжку.
 - Якщо $f'(x) \leq 0$, **функція спадає**.
- Це дозволяє зрозуміти, на яких ділянках функція є зростаючою або спадною.

Визначення локальних екстремумів та точок перетину з осями

4. Після того, як ми знайшли критичні точки, необхідно з'ясувати, чи є ці точки екстремумами (максимумами або мінімумами). Для цього використовуємо першу похідну:
- Якщо $f'(x)$ при переході через критичну точку змінює знак з мінуса на плюс, то дана точка є **точкою локального мінімуму**.
 - Якщо $f'(x)$ при переході через критичну точку змінює знак з плюса на мінус, то дана точка є **точкою локального максимуму**.

Визначення точок перегину і проміжків опуклості функції

5. Для визначення точок перегину та проміжків опуклості використовують другу похідну:

- Якщо $f''(x) > 0$, функція є опуклою вниз.
- Якщо $f''(x) < 0$, функція є опуклою вгору.

Визначення наявності асимптот функції

6. Після дослідження похідної функції можна також визначити наявність асимптот.

Вертикальні асимптоти: графік функції $y = f(x)$ має вертикальну асимптоту при $x \rightarrow a$, якщо границя функції нескінченна

$\lim_{x \rightarrow a} f(x) = \pm\infty$. Точка $x = a$ є точкою розриву II роду, і рівняння вертикальної асимптоти має вигляд $x = a$.

Похилі асимптоти: рівняння похилої асимптоти задається $y = kx + b$, де k і b – границі, що обчислюються за правилом:

$$k = \lim_{x \rightarrow \pm\infty} \frac{f(x)}{x};$$

$$b = \lim_{x \rightarrow \pm\infty} (f(x) - kx);$$

Якщо обидві границі існують і скінченні, функція має похилу асимптоту. Інакше – не має.

Горизонтальні асимптоти: крива $y = f(x)$ має горизонтальну асимптоту $y = b$, якщо існує скінченна границя функції при $x \rightarrow \pm\infty$ і ця границя дорівнює $\lim_{x \rightarrow \pm\infty} f(x) = b$.

Приклади дослідження функцій за допомогою похідних

1. Дослідити функцію $y = \frac{x^2 + x + a}{x}$;

1) $D \in (-\infty; 0) \cup (0; \infty)$

2) $y' = \frac{(2x+1) \cdot x - 1 \cdot (x^2 + x + a)}{x^2} = \frac{2x^2 + x - x^2 - a}{x^2} = \frac{x^2 - a}{x^2} = 0;$

$$\begin{cases} u = x^2 + x + a & u' = 2x + 1 \\ v = x & v' = 1 \end{cases}$$

if $a = 0 \Rightarrow y' = \frac{x^2}{x^2} = 1 \neq 0 \Rightarrow$ крит. точок немає

$\phi \uparrow$ при $\forall x \neq 0$

if $a > 0 \Rightarrow y' = \frac{x^2 - a}{x^2} = 0; x^2 = a; x = \pm\sqrt{a}$

$\phi \uparrow$ при $x \in (-\infty; -\sqrt{a}]$ і $x \in [\sqrt{a}; \infty)$

$\phi \downarrow$ при $x \in [-\sqrt{a}; 0)$ і $x \in (0; \sqrt{a}]$

$$\begin{aligned} x_{\max} = -\sqrt{a} \Rightarrow y_{\max} &= \frac{(-\sqrt{a})^2 + (-\sqrt{a}) + a}{-\sqrt{a}} = \frac{a - \sqrt{a} + a}{-\sqrt{a}} = \frac{2a - \sqrt{a}}{-\sqrt{a}} \\ &= 1 - 2\sqrt{a} \end{aligned}$$

$$x_{\min} = \sqrt{a} \Rightarrow y_{\min} = \frac{(\sqrt{a})^2 + \sqrt{a} + a}{\sqrt{a}} = \frac{2a + \sqrt{a}}{\sqrt{a}} = 2a\sqrt{a} + 1$$

if $a < 0 y' \Rightarrow 0$

$\phi \uparrow$ при $\forall x \in (-\infty; 0)$ і $(0; \infty)$

3) $y'' = \left(\frac{x^2 - a}{x^2} \right)' = \frac{2x \cdot x^2 - 2x(x^2 - a)}{x^4} = \frac{2x(x^2 - x^2 + a)}{x^4} = \frac{2a}{x^3}$

$$\begin{aligned} u &= x^2 - a; & u' &= 2x; \\ v &= x^2; & v' &= 2x; \end{aligned}$$

if $a = 0 \Rightarrow y'' = 0; \Rightarrow$ т – к перетину не існує.

if $a > 0$ при $x \in (-\infty; 0)$ – опукла вниз;

при $x \in (0; \infty)$ – опукла вгору;

4) $k = \lim_{x \rightarrow \infty} \frac{f(x)}{x} = \lim_{x \rightarrow \infty} \frac{x^2 + x + a}{x^2} = 1;$

$$b = \lim_{x \rightarrow \infty} \left(\frac{x^2 + x + a}{x} - x \right) = \lim_{x \rightarrow \infty} \frac{x^2 + x + a - x^2}{x} = 1;$$

$y = x + 1$ – похила асимптота.

2. Дослідити функцію $y = \frac{a}{x^2} + \frac{x}{a}$;

1) $D \in (-\infty; 0) \cup (0; \infty)$

2) $y' = \left(ax^{-2} + \frac{1}{a}x \right)' = -2ax^{-3} + \frac{1}{a} = -\frac{2a}{x^3} + \frac{1}{a} = 0;$

$$\frac{2a^2 + x^3}{ax^3} = 0; \quad x^3 = 2a^2; \quad x = \sqrt[3]{2a^2} - \text{критична точка};$$

if $a > 0 \Rightarrow x_{\min} = \sqrt[3]{2a^2}$

$\phi \uparrow$ при $x \in (-\infty; 0)$ і $x \in [\sqrt[3]{2a^2}; \infty)$

if $a < 0 \Rightarrow x_{\max} = \sqrt[3]{2a^2}$

$\phi \downarrow$ при $\forall x \in (-\infty; 0)$ і $[\sqrt[3]{2a^2}; \infty)$

$\phi \uparrow$ при $\forall x \in (0; \sqrt[3]{2a^2}]$

3) $u = x^3 - 2a^2; \quad u' = 3x^2;$
 $v = ax^3; \quad v' = 3ax^2;$

$$y'' = \frac{3x^2 \cdot ax^3 - 3ax^2(x^3 - 2a^2)}{a^2x^4} = \frac{3ax^2(x^3 - x^3 + 2a^2)}{a^2x^6} = \frac{3 \cdot 2a^2 \cdot a}{a^2x^4} = \frac{6a}{x^4} = 0 \quad - \text{немає}$$

точок перетину;

if $a > 0$ при $x \in (-\infty; 0) \cup (0; \infty)$ – ϕ . опукла вниз;

if $a < 0$ при $x \in (-\infty; 0) \cup (0; \infty)$ – ϕ . опукла вгору;

4) $k = \lim_{x \rightarrow \infty} \frac{\frac{a}{x^2} + \frac{x}{a}}{x} = \lim_{x \rightarrow \infty} \frac{a^2 + x^3}{ax^3} = \frac{1}{a};$

$$b = \lim_{x \rightarrow \infty} \left(\frac{a^2 + x^3}{ax^2} - \frac{1}{a}x \right) = \lim_{x \rightarrow \infty} \left(\frac{a^2 + x^3}{ax^2} - \frac{x^3}{ax^2} \right)$$

$$= \lim_{x \rightarrow \infty} \left(\frac{a^2 + x^3 - x^3}{ax^2} \right) = \lim_{x \rightarrow \infty} \left(\frac{a^2}{ax^2} \right) = 0;$$

$y = \frac{1}{a}x$ – похила асимптота.

ТЕОРЕТИЧНА ЧАСТИНА З ІНФОРМАТИКИ

Використані технології і архітектура програми

Використані технології

У процесі створення даного проєкту було використано сучасні підходи до програмування та технології, які забезпечують зручність і ефективність розробки. Основною мовою програмування обрано **Python**, що є універсальним інструментом для реалізації математичних розрахунків, побудови графіків і створення графічного інтерфейсу. Python має широкий набір бібліотек і модулів, які суттєво полегшують роботу з математичними операціями, візуалізацією даних та розробкою користувацького інтерфейсу. Також причиною обрання саме цієї мови стала зацікавленість авторки в вивченні математичних бібліотек Python, зважаючи на те, що вона опановує цю мову вже 4 роки і прагне продовжувати розвиток у нових її напрямках.

Віртуальне середовище Python також було використано у роботі для ізоляції проєкту і керування залежностями, що забезпечує стабільність роботи застосунку.

Основні бібліотеки, які було використано у ході розробки додатку:

- **NumPy** — це основний пакет для наукових обчислень на Python. Це бібліотека Python, яка надає об'єкт багатовимірного масиву, різні похідні об'єкти (такі як замасковані масиви та матриці), а також набір процедур для швидких операцій над масивами, включаючи математичні, логічні, маніпуляції формою, сортування, вибір, введення/виведення, дискретне перетворення Фур'є, базова лінійна алгебра, основні статистичні операції, випадкове моделювання та багато іншого більше.
- **SymPy** — це бібліотека Python для символічної математики.
- **Matplotlib** — це комплексна бібліотека для створення статичних, анімованих та інтерактивних візуалізацій.

- **matplotlib.pyplot** — це інтерфейс для matplotlib на основі стану. Він забезпечує неявний, схожий на MATLAB, спосіб побудови. Він також відкриває фігури на вашому екрані та діє як менеджер графічного інтерфейсу користувача (GUI).
- **CustomTkinter** — це Desktop бібліотека інтерфейсу користувача Python на основі Tkinter, яка надає сучасні та повністю настроювані віджети. За допомогою CustomTkinter ви отримаєте узгоджений вигляд на всіх платформах настільних ПК (Windows, macOS, Linux).
- **Tkinter** — є стандартним Desktop інтерфейсом Python до інструментарію GUI Tcl/Tk. І Tk, і tkinter доступні на більшості платформ Unix, включаючи macOS, а також у системах Windows.
- **PIL** — бібліотека зображень Python додає можливості обробки зображень до інтерпретатора Python. Ця бібліотека забезпечує розширену підтримку форматів файлів, ефективне внутрішнє представлення та досить потужні можливості обробки зображень.
- **Fitz (PyMuPDF)** — це високопродуктивна бібліотека Python для вилучення даних, аналізу, перетворення та обробки документів PDF (та інших).

Архітектура програми

Проект побудовано на основі чітко структурованої архітектури, що дозволяє розділити функціональність за окремими модулями та пакетами, забезпечуючи простоту в обслуговуванні, масштабованість і читабельність коду. Основний пакет програми — **modules**, у якому організовано весь функціонал для виконання задачі.

Поділ коду на модулі та пакети забезпечує:

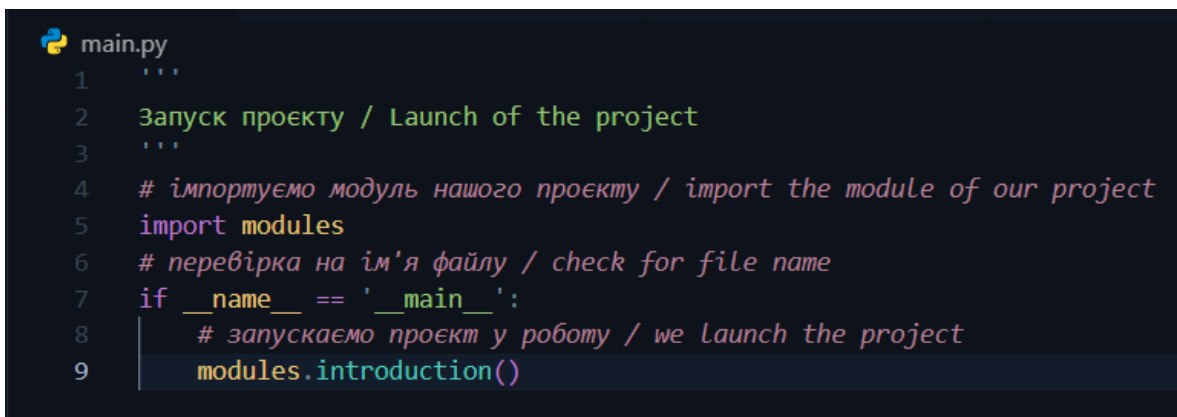
- **Логічну ізоляцію** функцій: кожен компонент відповідає за окремий аспект роботи програми;
- **Масштабованість**: легко додавати нові функції або розширювати існуючі без зміни основної архітектури;

- **Зручність тестування:** кожен модуль можна перевірити окремо, що знижує ризик помилок;
- **Повторне використання коду:** функції, які використовуються в декількох місцях, доступні через окремі модулі;

Основна структура проєкту складається з таких файлів і пакетів:

1. Файл `main.py`

Відповідає за запуск програми та ініціалізацію основного вікна. У ньому підключаються всі головний пакет й забезпечується зв'язок між компонентами:



```

main.py
1  '''
2  Запуск проєкту / Launch of the project
3  '''
4  # імпортуємо модуль нашого проєкту / import the module of our project
5  import modules
6  # перевірка на ім'я файлу / check for file name
7  if __name__ == '__main__':
8      # запускаємо проєкт у роботу / we launch the project
9      modules.introduction()

```

2. Пакет `modules`

У кореневій директорії цього пакета зберігаються:

- **Файли з реалізацією 4 основних вікон** — дозволяють відображати окремі частини функціоналу програми, організуючи її зручний інтерфейс.
- **Файл з константами** — містить глобальні змінні, що використовуються у проєкті. Це спрощує внесення змін у програму, адже всі важливі параметри зосереджені в одному місці. Також створення глобальних змінних саме таким чином допомагає уникнути можливих проблем, помилок і некоректних даних у програмі під час її використання.

3. Пакет style

Призначений для:

- Задавання стилів елементам інтерфейсу customtkinter та tkinter.
- Забезпечення єдиного дизайну програми, що покращує її візуальну узгодженість.

Завдяки цьому пакет забезпечується легке оновлення стилів для всього застосування.

4. Пакет main_elements

Містить модулі для створення:

- Основних об'єктів графічного інтерфейсу: вікон, кнопок, чекбоксів, полотен тощо.
- Компонента FigureCanvasTkAgg з matplotlib.backends.backend_tkagg, який використовується для відображення графіків у вікнах customtkinter.

Завдяки цьому розділенню полегшується підтримка та модифікація інтерфейсу.

5. Пакет data_calculation

Містить функції для математичних розрахунків і дослідження функцій, зокрема:

- область визначення функції;
- парність чи непарність;
- зростання та спадання;
- локальні максимуми та мінімуми;
- максимальне і мінімальне значення функції;
- нулі функції(x_1, x_2, x_n);
- точки перетину функції з осями абсцис і ординат ($x_0; y_0$);
- проміжок знакосталості;
- точки перегину;

- проміжки опуклості;
- асимптоти;

6. Пакет plotting

Призначений для:

- Побудови графіків функцій та їх дослідження.
- Взаємодії з функціями з пакета `data_calculation` для відображення результатів дослідження на графіку.
- Інтеграції графіків у GUI через `FigureCanvasTkAgg`.

Ця структура дозволяє легко додавати нові методи візуалізації.

Компоненти програми взаємодіють через добре структуровану систему імпортів:

- головний пакет **modules** ініціалізує програму, викликаючи модулі та функції з інших пакетів у `__init__.py`;

```
modules > __init__.py
1  """
2  Ініціалізація усіх файлів до пакету. / Initialization of all files to the package.
3  """
4  from .variables_constants import *
5  # імпортуємо функцію introduction з файлу introduction_window для запуску титульного вікна / import the introduction function from introduction_window
6  from .introduction_window import introduction
7  # імпортуємо з main_window функцію run_main для можливості запуску головного додатку / import the run_main function from main_window
8  from .main_window import run_main, on_close
9  # імпортуємо вікно документації для його роботи і відображення під час використання проекту / import the documentation window
10 from .document_window import *
11 # імпортуємо з plotting усі функції для кнопок або чекбоксів для подальшого використання / import all plotting functions
12 from .plotting import *
13 # імпортуємо з data_calculation усі функції для виконання дослідження графіків функцій / import all data_calculation functions
14 from .data_calculation import *
15 # імпортуємо усі зміни стилів для елементів з файлу style / import all style changes for elements from style
16 from .style import *
17 # імпортуємо функцію functions_window з файлу window_with_fun для запуску вікна для вибору функцій / import the functions_window function from window_with_fun
18 from .window_with_fun import functions_window
19 # імпортуємо вікно помилки для його подальшого відображення при необхідності / import the error window
20 from .error_window import show_error_window
```

- `main_elements` відповідає за створення елементів інтерфейсу, які використовують функції з **data_calculation** та **plotting** для обчислень і візуалізації;
- **style** забезпечує єдиний дизайн усіх елементів інтерфейсу;
- файл констант дозволяє узгоджено змінювати глобальні налаштування програми;

Вплив структури на проєкт:

- **покращена організація коду:** кожна функція чітко відокремлена і відповідає за свою задачу;
- **легка підтримка:** у разі змін у програмі достатньо змінити конкретний модуль, не впливаючи на решту коду;
- **зручність для роботи в команді:** завдяки поділу на пакети можна працювати над окремими модулями незалежно;

Така архітектура забезпечує надійність і ефективність програми, дозволяючи легко адаптувати її до нових задач або розширювати функціонал.

Алгоритм роботи програми

1. Ініціалізація програми

- Запускається файл `main.py`.
- Імпортуються всі необхідні модулі, пакети та бібліотеки.
- Ініціалізується основне вікно програми (`customtkinter`).
- Налаштовуються стилі інтерфейсу, використовуючи модулі з пакета `style`.

2. Створення графічного інтерфейсу

- У вікні програми створюються основні елементи інтерфейсу:
 - Поля для введення коефіцієнтів функцій.
 - Кнопки для запуску розрахунків, побудови графіків, дослідження функцій.
 - Чекбокси для вибору параметрів (наприклад, відображення похідної або критичних точок).
 - Полотна для візуалізації графіків, з використанням `FigureCanvasTkAgg`.
- Елементи створюються через модулі з пакета `main_elements`.

3. Введення даних користувачем

- Користувач вводить параметри функції (коефіцієнти, тип функції тощо).
- Дані перевіряються на коректність.

4. Розрахунки

- Після натискання кнопки викликаються функції з пакета `data_calculation`:
 - Обчислення похідної функції.
 - Пошук критичних точок (максимумів, мінімумів, точок перегину).
 - Аналіз інтервалів зростання та спадання функції.
 - Визначення особливих точок, таких як вертикальні асимптоти або точки розриву.

5. Побудова графіків

- На основі результатів розрахунків викликаються функції з пакета `plotting`:
 - Створюється графік заданої функції.
 - Накладаються графіки похідних.
 - Відображаються критичні точки та інші особливості графіку (наприклад, точки перетину з осями).
- Графік інтегрується в графічний інтерфейс через полотно (`FigureCanvasTkAgg`).

6. Відображення результатів

- Результати розрахунків виводяться у відповідних `CTkLabel` та на графіку.
- Всі графіки та точки дослідження відображаються у візуалізації.

-

7. Інтерактивна робота

- Користувач може змінювати коефіцієнти, щоб побачити, як вони впливають на графік.
- Надається можливість ввімкнути або вимкнути відображення похідних на графіку.

8. Завершення роботи

- Користувач може закрити програму через стандартний механізм закриття вікна.
- При закритті звільняються всі ресурси, і програма завершує свою роботу.

Алгоритм також передбачає гнучкість у додаванні нових функцій (наприклад, інших математичних досліджень або методів візуалізації), оскільки всі компоненти програми чітко ізольовані.

ВИСНОВКИ

У результаті виконання цього проєкту була розроблена програма для дослідження графіків функцій за допомогою похідних, яка поєднує теоретичні математичні знання з практичними інструментами для візуалізації та аналізу функцій. Програма надає користувачеві можливість не лише вводити функції та їх параметри, а й здійснювати їх детальне математичне дослідження, спостерігаючи за змінами графіків у реальному часі. Загалом можна зробити наступні висновки:

- Програма успішно реалізує основні математичні концепції, пов'язані з дослідженням графіків функцій за допомогою похідної. Вона надає можливість аналізувати функції з точки зору їх поведінки на різних інтервалах, а також допомагає візуалізувати їх графіки, що є важливим для розуміння якості математичних досліджень. Завдяки цьому користувач може вивчати різні типи функцій і знаходити критичні точки, що є корисним для вирішення задач в математичному аналізі та диференціальному численні.
- Цей проєкт є корисним інструментом для здобувачів знань, студентів і всіх тих, хто вивчає математичний аналіз. Він дозволяє швидко отримати візуальні результати і таким чином краще усвідомлювати вплив різних коефіцієнтів на поведінку функцій. Інтерактивний характер програми дозволяє гнучко змінювати параметри і бачити, як змінюються графіки, що робить навчання більш ефективним та захопливим.
- Структура програми була спроектована таким чином, щоб забезпечити максимальну гнучкість і зрозумілість. Використання модулів для кожної окремої функціональної частини програми (введення даних, розрахунки, побудова графіків) дозволяє легко модифікувати й розширювати програму в майбутньому. Це також полегшує підтримку коду та забезпечує його зручність для інших розробників, що можуть брати участь у розвитку проєкту.

- Використання Tkinter та customTkinter для створення інтерфейсу користувача є ефективним рішенням, оскільки ці бібліотеки забезпечують простоту створення графічних елементів і дозволяють зберігати високу швидкість роботи програми. Взаємодія з користувачем через кнопки, поля введення та чекбокси забезпечує зручність користування, що робить програму інтуїтивно зрозумілою навіть для користувачів без глибоких знань програмування.
- У подальшому проєкт можна розширити, додавши нові функції для більш глибокого аналізу функцій. Також можна інтегрувати підтримку більш складних математичних бібліотек, що дозволить зробити програму ще більш потужною і універсальною для різних типів задач.

Завдяки застосуванню сучасних технологій програмування та чіткій організації коду, проєкт виявився успішним інструментом для дослідження математичних функцій. Програма не тільки демонструє математичні ідеї, але й робить цей процес більш доступним і захоплюючим для користувача. Вона надає можливість швидко обчислювати, візуалізувати та аналізувати функції, що є важливим інструментом у навчальному процесі.

Сподіваюсь програма буде корисна ліцеїстам та іншим здобувачам знань при вивченні похідних і дослідження графіків функцій з їх допомогою!

ВИКОРИСТАНІ ДЖЕРЕЛА:

1. **Алгебра 10 клас: Поглиблений рівень вивчення** / А.Г. Мерзляк, Д.А. Номіровський, В.Б. Полонський, М.С. Якір. – Харків: Гімназія, 2018. – 256 с.
2. **Алгебра 9 клас: Поглиблений рівень вивчення** / А.Г. Мерзляк, В.Б. Полонський, М.С. Якір. – Харків: Гімназія, 2009. – 240 с.
3. **Математика 10 клас: Поглиблений рівень вивчення** / М.І. Бурда, Т.В. Колесник, Ю.І. Мальований. – Київ: Оріон, 2018. – 264 с.
4. Customtkinter Documentation
5. Matplotlib 3.10.0 documentation
6. Zno Academia
7. YukhymCommunity
8. NumPy Documentation
9. SymPy Documentation
10. Tkinter Documentation
11. PIL Documentation
12. PyMuPDF Documentation