# КИЇВСЬКИЙ НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ ІМЕНІ ТАРАСА ШЕВЧЕНКА

**ФАКУЛЬТЕТ ІНФОРМАЦІЙНИХ ТЕХНОЛОГІЙ КАФЕДРА ІНТЕЛЕКТУАЛЬНИХ ТЕХНОЛОГІЙ**

# ПОЯСНЮВАЛЬНА ЗАПИСКА

**ДО КУРСОВОЇ РОБОТИ З ПРОЄКТУВАННЯ АЛГОРИТМІВ ТА ПРОГРАМУВАННЯ**

# на тему:

“Візуалізація графічної інтерпретації чисельних методів пошуку коренів нелінійних рівнянь. Метод Ньютона (метод дотичних)”

Виконав студент 2 курсу групи КН - 21

Домбровський Артур Юліанович

Засвідчую, що в курсовій роботі немає запозичень з праць інших авторів без відповідних посилань





Керівник курсової роботи:

к.т.н., доцент Тмєнова Наталія Пилипівна



Оцінка за курсову роботу:



Члени комісії:



Київ – 2024

# КИЇВСЬКИЙ НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ ІМЕНІ ТАРАСА ШЕВЧЕНКА

**ФАКУЛЬТЕТ ІНФОРМАЦІЙНИХ ТЕХНОЛОГІЙ КАФЕДРА ІНТЕЛЕКТУАЛЬНИХ ТЕХНОЛОГІЙ**

## ЗАВДАННЯ

на курсову роботу з проєктування алгоритмів та програмування студенту **Домбровському Артуру Юліановичу**

1. Тема роботи:

Візуалізація графічної інтерпретації чисельних методів пошуку коренів нелінійних рівнянь. Метод Ньютона (метод дотичних)

1. Термін здачі закінченого проєкту: «27» травня 2024 р.
2. Вихідні дані до проєкту: немає
3. Зміст роботи:

Розділ 1. Аналіз алгоритмів пошуку коренів нелінійних рівнянь.

Розділ 2. Проєктування програмного застосунку пошуку коренів нелінійних рівнянь Методом Ньютона.

Розділ 3. Розробка програмного застосунку пошуку коренів нелінійних рівнянь Методом Ньютона.

1. Перелік презентаційного матеріалу:

1. Мета та завдання курсової роботи (1 - 2 слайди).

2. Характеристика та класифікація алгоритмів пошуку коренів нелінійних рівнянь (5-10 слайдів).

3. Узагальнений алгоритм пошуку коренів нелінійних рівнянь Методом Ньютона. (1-2 слайди).

4. Проєктування програмного застосунку пошуку коренів нелінійних рівнянь Методом Ньютона. (5-8 слайдів).

5. Реалізація програмного застосунку пошуку коренів нелінійних рівнянь Методом Ньютона. (1-2 слайди).

6. Висновки (1-2 слайд).

1. Дата видачі завдання: «14» березня 2024 р

Графік виконання курсової роботи

| **№** | **Назва етапу** | | **Терміни** | | **Примітки / відмітка про виконання** | |
| --- | --- | --- | --- | --- | --- | --- |
| 1 | Вибір теми та керівника курсової роботи | | 26 лютого 7 березня | – | Заява на виконання курсової роботи, що підписана студентом та керівником | |
| 2 | Обговорення з керівником постановки завдання та змісту пояснювальної записки до курсової роботи | | 8 березня 14 березня | – | Заповнений бланк завдання на курсову роботу, що підписаний студентом та керівником роботи | |
| 3 | Аналіз постановки задачі, формалізація задачі, вибір методів та засобів реалізації поставленої задачі, аналіз літературних джерел | | 15 березня  20 березня | – | Сформований матеріал до розділу 1 пояснювальної записки курсової роботи, оформлення списку джерел | |
| 4 | *Перше узгодження з керівником* | | 21 березня 27 березня | – |  | |
| 5 | Розробка алгоритму, вибір структур даних, проектування програмного інтерфейсу з користувачем. | | 28 березня 09 квітня | – | Сформований матеріал розділу 2 пояснювальної | до |
| 6 | *Друге узгодження з керівником* | | 10 квітня 14 квітня | – |  | |
| 7 | Розробка та тестування програмного продукту. | | 15 квітня 23 квітня | – | Готовий програмний продукт | |
| 8 | *Демонстрація базового програмного продукту. узгодження з керівником* | *варіанту*  *Третє* | 24 квітня 30 квітня | – |  | |
| 9 | Доопрацювання програмного продукту, всебічне заключне тестування, розробка керівництва користувача. | | 01 травня 07 травня | – | Сформований матеріал до розділу 3 пояснювальної записки, підготовлений демонстраційний приклад роботи програми | |
| 10 | Оформлення пояснювальної записки, підготовка презентації | | 08 квітня 14 квітня | – | Готова пояснювальна записка та презентація для захисту курсової роботи | |

Керівник роботи Наталія ТМЄНОВА



Завдання прийняв до виконання

Артур ДОМБРОВСЬКИЙ

## Зміст

[**Вступ**](#_gjdgxs)5

# [**РОЗДІЛ 1. АНАЛІЗ ЧИСЕЛЬНИХ МЕТОДІВ ПОШУКУ КОРЕНІВ НЕЛІНІЙНИХ РІВНЯНЬ**](#_yiod5j6h9qlm)7

* 1. Поняття чисельних методів пошуку коренів нелінійних рівнянь 7
  2. Аналіз методу Ньютона (методу дотичних) 8
  3. Порівняння з іншими методами пошуку коренів та їх графічним представленням 11
  4. [Висновки до розділу 1](#_3znysh7) 14

# [РОЗДІЛ 2. ПРОЄКТУВАННЯ ПРОГРАМНОГО ЗАСТОСУНКУ ПОШУКУ КОРЕНІВ НЕЛІНІЙНИХ РІВНЯНЬ МЕТОДОМ НЬЮТОНА](#_2et92p0) 15

* 1. Постановка задачі проєктування 15
  2. Опис архітектури та взаємодія між модулями програмного засобу 16
  3. Математична модель алгоритму чисельного методу пошуку коренів нелінійних рівнянь 19
  4. Створення графічного представлення алгоритму чисельного методу 20
  5. Розробка графічного інтерфейсу для роботи ПЗ 23
  6. Висновки до розділу 2 25

[**Перелік використаних джерел**](#_2s8eyo1)26

## Вступ

**Мета роботи:** дослідження ефективності, швидкості роботи, аналіз властивостей конвергенції чисельних методів пошуку коренів нелінійних рівнянь та, зокрема, візуалізувати його графічне представлення роботи алгоритму.

**Цілі роботи:**

1. Дослідження продуктивності та швидкості чисельних методів: оцінити продуктивність і швидкість різних чисельних методів пошуку коренів нелінійних рівнянь, з акцентом на роботу чисельного метода Ньютона.
2. Візуалізація роботи алгоритму: розробити повне графічне представлення методу Ньютона в дії, що дозволяє чітко візуалізувати його ітераційний процес і поведінку наближення до кореня рівняння.

Для досягнення поставленої мети необхідно вирішити такі завдання:

1. Дослідити літературні джерела, що описують чисельні методи пошуку коренів та його видів: метод Ньютона, метод Січних, метод Дихотомії, метод Хорд, Комбінований метод;
2. Розробити модель графічного подання роботи чисельних методів пошуку коренів нелінійних рівнянь на різних наборах даних;
3. Розробити програмний застосунок, що реалізує візуалізацію ітераційного процесу та розрахунки за методом Ньютона ;
4. Провести обчислювальні експерименти для нелінійних рівнянь: поліноміальних, степеневих, логарифмічних, тригонометричних, трансцендентних та складених рівнянь. Протестувати працездатність розробленої програми, проаналізувати отримані результати.

У першому розділі буде проведено аналіз літературних джерел, що описують чисельні методи пошуку коренів нелінійних рівнянь, проаналізовано їх переваги та недоліки, що дозволить отримати базові знання для подальшої розробки.

У другому розділі буде проведено проєктування програмного застосунку, який реалізує візуалізацію та розрахунки за методом Ньютона, що дозволить відобразити роботу чисельного методу на різних наборах даних.

В третьому розділі буде розроблено програмний застосунок та проведено обчислювальні експерименти для практичного використання методу Ньютона у вирішенні завдань пошуку коренів рівнянь. Результати роботи будуть візуально представлені для кращого розуміння та оцінки ефективності алгоритму.

**Опис використаних засобів розробки:** для виконання курсової роботи була обрана мова програмування Python, а сама розробка велась в IDE Visual Studio Code.

**Практичне значення отриманих результатів:** візуалізація чисельних методів пошуку алгоритмів може бути використана в навчальних цілях в рамках курсу “Чисельні методи”, а також в практичних цілях пошуку коренів рівнянь.

## РОЗДІЛ 1. АНАЛІЗ ЧИСЕЛЬНИХ МЕТОДІВ ПОШУКУ КОРЕНІВ НЕЛІНІЙНИХ РІВНЯНЬ

# 1.1 Поняття чисельного методу та пошуку коренів

[1]**Чисельні методи** — це алгоритми, які використовуються для наближення до точного значення функції або розв'язання задачі. Ці методи застосовуються, коли аналітичне розв'язання неможливе або є занадто складним.

[2]Чисельні методи використовуються для побудови математичних моделей реальних явищ, які описуються складними рівняннями. Наприклад, моделювання траєкторії польоту ракети, розрахунок міцності будівельних конструкцій, прогнозування погодних умов.

**Задача пошуку коренів** полягає в знаходженні значень змінної, при яких нелінійна функція дорівнює нулю.

**Наближений корінь** — це значення змінної, яке з заданою точністю наближає точний корінь нелінійної функції.

[3]**Нелінійне рівняння** — це рівняння, в якому змінна з'являється в ступені, більшому за одиницю, або в добутку з іншими змінними. Наприклад:

[3]**Види нелінійних рівнянь:**

1. **Алгебраїчні рівняння:** Містять лише змінні, піднесені до цілих степенів. Наприклад:
2. **Трансцендентні рівняння:** Містять змінні у функціях, таких як експоненти, логарифми, тригонометричні функції. Наприклад: **Диференціальні рівняння:** Містять похідні змінних за часом або простором. Наприклад:
3. **Інтегральні рівняння:** Містять інтеграли змінних. Наприклад:

[2] Існує багато різних чисельних методів для пошуку коренів нелінійних рівнянь. Деякі з найпоширеніших методів включають:

1. **Метод дихотомії:** Цей метод ділить інтервал, у якому знаходиться корінь, навпіл на кожному кроці, поки не буде знайдено значення змінної з заданою точністю.
2. **Метод Ньютона (метод дотичних):** Ефективний метод для знаходження коренів однозмінних нелінійних рівнянь.
3. **Метод січних:** Простий метод, який використовує лінійну інтерполяцію для наближення кореня.
4. **Метод хорд:** Подібний до методу січних, але використовує прямі, що проходять через точки перетину функції з віссю x.
5. **Комбінований метод:** Поєднує в собі різні методи для кращої точності та стійкості. Наприклад, метод може використовувати метод Ньютона для швидкого наближення кореня, а потім метод січних для уточнення цього наближення.
6. **Метод ітерацій:** Цей метод використовує ітераційний процес, де на кожному кроці значення змінної оновлюється за допомогою заданої формули.

# 1.2 Аналіз методу Ньютона-Рафсона (методу дотичних)

[3]Метод Ньютона-Рафсона, також відомий як метод дотичних, є ітеративним методом для наближеного знаходження коренів нелінійних рівнянь f(x) = 0. Цей метод використовується для розв'язання рівнянь з однією змінною.

[4]Для застосування методу Ньютона необхідні наступні початкові дані:

1. **Рівняння f(x) = 0**, яке потрібно розв'язати.
2. **Початкове наближення x0**: Це перше наближення до кореня рівняння, воно є значенням, з якого починається ітеративний процес методу.
3. **Проміжок [a, b]**: Це проміжок, який містить корінь рівняння.
4. **Точність (ε)**: Це граничне значення абсолютної величини різниці між послідовними наближеннями.

Важливо зазначити, що вибір початкового наближення може суттєво впливати на збіжність методу, адже якщо початкове наближення знаходиться занадто далеко від кореня, метод може не збігатися.

[2] Проміжок [а, b] можна визначити за Теоремою Больцано-Коші: Якщо неперервна функція f(x) набуває на кінцях відрізка [а, b] значень різних знаків, тобто f(a) · f(b) < 0, то всередині цього відрізка міститься принаймні один корінь. Цей корінь буде єдиним, якщо похідна f'(x) існує і зберігає постійний знак усередині відрізка [а, b].

[5] **Алгоритм методу (1.2.1):**

1. **Перевірити умови збіжності:**
   1. f(x) та f'(x) неперервні та диференційовані в проміжку [a, b].
   2. Початкове наближення x0 знаходиться в проміжку [a, b].
2. **Вибрати початкове наближення x0.**
3. **На кожному кроці ітерації**:
   1. Обчислити похідну в точці xi.
   2. Обчислити наступне наближення xi+1 за формулою:

(1.2.1.1)

1. **Повторити крок “с)”, поки не буде досягнуто заданої точності ε:**

(1.2.1.2)

[6] Метод Ньютона-Рафсона має квадратичну швидкість збіжності. Однак метод не гарантує збіжності для будь-якого початкового наближення. Тому важливо перевіряти умови збіжності методу перед його застосуванням .

Збіжність залежить від:

1. **Гладкості функції f(x):** Метод потребує, щоб f(x) та f'(x) були неперервними та диференційованими в точці кореня.
2. **Початкового наближення:** Якщо початкове наближення знаходиться занадто далеко від кореня, метод може не збігатися.
3. **Існування єдиного кореня:** Метод гарантує збіжність лише до одного кореня в заданій області.

[2] Існують різні модифікації методу Ньютона-Рафсона , які розроблені для покращення його стійкості, швидкості збіжності, для розширення його можливостей або спрощення чисельних обрахунків. Ось два поширених приклади:

1. **Метод січних:**

Спрощена версія методу Ньютона-Рафсона, подібний до методу Хорд, яка не потребує обчислення похідної . На кожній ітерації метод січних використовує два поточних наближення ( та ) для знаходження нового наближення за формулою:

.

Однак метод січних може бути більш стійким до шуму в даних, ніж метод Ньютона-Рафсона, і може бути корисним, коли складно або неможливо обчислити похідну f'(x) [4].

1. **Комбінований метод:**

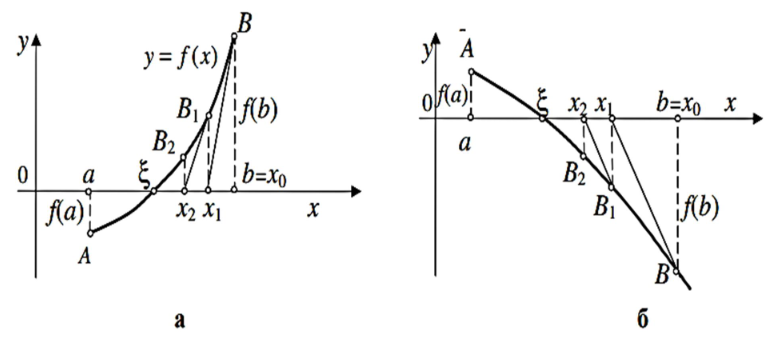
Поєднання методу Ньютона-Рафсона та Хорд для покращення збіжності та швидкості розв'язання. На кожній ітерації метод зменшує проміжок знаходження кореня на проміжку [а, b], одна границя проміжку обраховується за формулою (1.1) методу Ньютона-Рафсона, а інша границя проміжку за формулою методу Хорд:

Цей метод може бути корисним для задач, де метод Ньютона-Рафсона може не збігатися або коли швидкість збіжності методу Ньютона-Рафсона стає повільною.

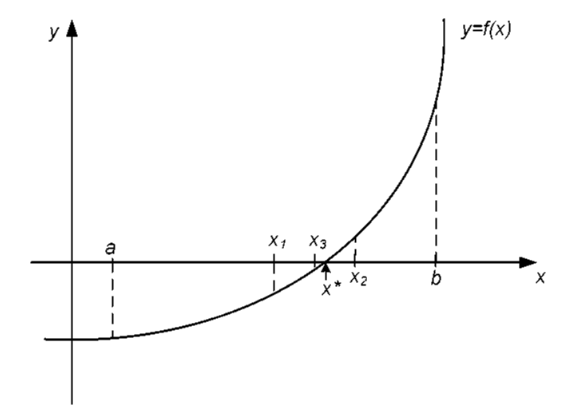
# 1.3 Порівняння з іншими методами пошуку коренів та їх графічним представленням

Розуміння графічного представлення методів є важливим етапом у проєктуванніпрограмного застосунку пошуку коренів нелінійних рівнянь чисельними методами.

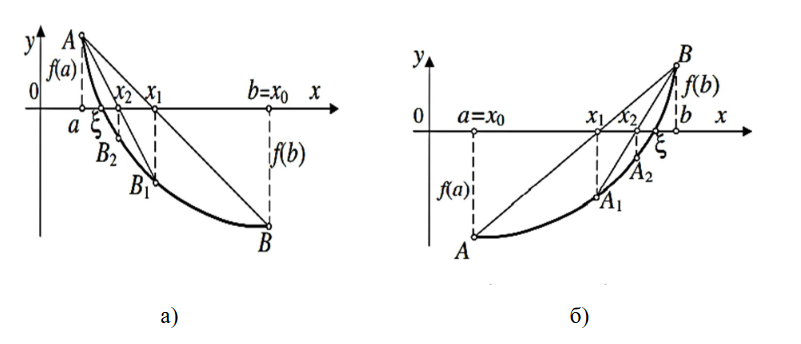
1. **Метод Ньютона-Рафсона:** можна візуалізувати, побудувавши графік функції та ітеративно проводячи дотичні до графіка в точках початкових наближень.

Рис. 1.1 Геометрична інтерпретація пошуку розв’язку нелінійного рівняння методом Ньютона: а) для увігнутої функції; б) для опуклої [4]

1. **Метод дихотомії (бісекція, половинне ділення):** Цей метод ділить проміжок, що містить корінь, навпіл на кожному кроці, зменшуючи проміжок, поки не буде знайдено значення змінної із заданою точністю.

Рис 1.2. Графічна інтерпретація розв'язання нелінійних рівнянь методом [4]

1. **Метод хорд:** Цей метод схожий на метод дихотомії, але замість того, щоб ділити інтервал навпіл, він з’єднує прямою кінці проміжку, що проходить через дві точки на кривій функції, аби знайти нове наближення кореня

Рис. 1.3 Графічна інтерпретація розв’язання нелінійних рівнянь методом хорд; а) у випадку, коли f(a)>0; б) f(а)<0 [4]

1. **Метод ітерацій:** Цей метод використовує ітераційний процес, де на кожному кроці значення змінної оновлюється за допомогою заданої формули.

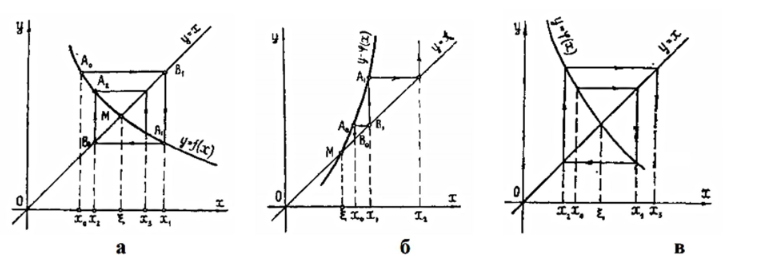


Рис. 1.4 Варіанти пошуку розв’язків нелінійного рівняння методом ітерацій: а) -1<Ф’(x)<0; б)-Ф’(x) >1; в) -Ф‘(x) <-1 [4]

[7] Вибір методу пошуку коренів залежить від конкретної задачі, її характеристик та необхідної точності. Важливо враховувати такі фактори:

Таблиця 1.1 — Переваги та недоліки чисельних методів

| Чисельний метод | Переваги | Недоліки |
| --- | --- | --- |
| Метод Ньютона-Рафсона | * Швидка збіжність (квадратична) * Може використовуватися для знаходження коренів багатовимірних функцій * Не потребує додаткових параметрів (наприклад, похідних вищих порядків) | * Може не збігатися для деяких початкових наближень * Чутливий до шуму в даних * Потрібна перша та друга похідні функції |
| Метод дихотомії | * Гарантовано збігається для будь-якої неперервної функції * Не потребує похідних функції * Простий у реалізації | * Лінійна швидкість збіжності * Може потребувати багато ітерацій для досягнення заданої точності |
| Метод хорд | * Швидша збіжність, ніж метод дихотомії (квадратична) * Простий у реалізації * Не потребує похідних вищих порядків | * Може не збігатися для деяких рівнянь * Чутливий до вибору початкових наближень |
| Метод ітерацій | * Може використовуватися для широкого класу рівнянь * Гнучкий * Не потребує похідних вищих порядків | * Може потребувати глибоких знань про задачу для вибору відповідної ітераційної схеми * Може мати повільну збіжність * Може не збігатися для деяких рівнянь |

[5] При виборі наближеного чисельного методу важливо враховувати такі фактори:

1. **Швидкість збіжності:** Яка швидкість збіжності необхідна для задачі?
2. **Стійкість:** Чи є задача чутливою до шуму в даних?
3. **Складність:** Наскільки складно реалізувати метод?
4. **Наявність похідних:** Чи доступні перша та/або друга похідні функції?
5. **Тип рівняння:** Чи є рівняння неперервним, диференційованим, з множинними коренями?

# 1.4 Висновки до розділу 1

У цьому розділі було розглянуто наближені методи пошуку коренів рівнянь. Кожен метод має свої переваги та недоліки, тому вибір методу залежить від конкретної задачі та необхідної точності. Проведений аналіз наближених чисельних методів, зокрема методу Ньютона-Рафсона, дозволяє зробити важливий висновок: розробка програмного забезпечення для пошуку коренів нелінійних рівнянь методом Ньютона-Рафсона та візуалізації роботи алгоритму дозволяє краще зрозуміти його принцип дії та проводити порівняльні дослідження.

## РОЗДІЛ 2. ПРОЄКТУВАННЯ ПРОГРАМНОГО ЗАСТОСУНКУ ПОШУКУ КОРЕНІВ НЕЛІНІЙНИХ РІВНЯНЬ МЕТОДОМ НЬЮТОНА

# 2.1 Постановка задачі проєктування

Розробити програмне забезпечення (ПЗ) для пошуку коренів нелінійних рівнянь методом Ньютона-Рафсона та візуалізації роботи алгоритму «відповідно до 1.2.1».

Задача ПЗ: на основі вхідних даних обрахувати та згенерувати візуалізацію методу пошуку коренів, та відобразити його у вікні програми. Отже, можемо розділити задачу на такі **цілі**:

1. Отримання вхідних даних які наведені «у розділі 1.2» від користувача через графічний інтерфейс
2. Генерація математичної анімації методу пошуку на основі графічної інтерпретації методу «відповідно до 1.3»;
3. Виведення результатів роботи програми.

**Функціональні вимоги ПЗ:**

1. Введення початкових даних які наведені «у розділі 1.2» користувачем:
   1. Текст нелінійного рівняння.
   2. Число початкового наближення кореня.
   3. Число допустимої похибки.
   4. Числа проміжок на якому знаходиться корінь
   5. Число кількість ітерацій
2. Перевірка початкових даних на помилку.
3. Пошук коренів нелінійного рівняння методом Ньютона-Рафсона.
4. Генерації візуалізації пошуку коренів нелінійного рівняння методом Ньютона-Рафсона.
5. Виведення результату пошуку коренів нелінійного рівняння.
6. Відтворення відео згенерованої візуалізації чисельного методу.

**Нефункціональні вимоги ПЗ:**

1. Швидкість роботи
2. Стабільність роботи
3. Простота використання інтерфейсу
4. Масштабованість функціональних можливостей ПЗ

Мета проєктування програмного комплексу для пошуку коренів нелінійних рівнянь методом Ньютона полягає у створенні математичного та навчально-дослідницького інструменту, який може бути використаний у межах навчального курсу "Чисельні методи" в університетах України.

# 

# 2.2 Опис архітектури та взаємодія між модулями програмного засобу

Для реалізації відповідних цілей наведених «у розділі 2.1» та виконання функціональних і нефункціональних вимог необхідно спроєктувати три модулі: Графічний Інтерфейс Користувача (“ГІК”), Математична Анімація (“МА”), Чисельний метод (“ЧМ”).

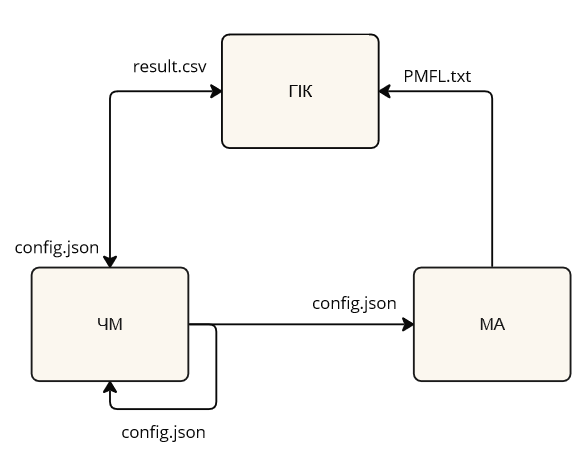
**Архітектура ПЗ:**

1. **Графічний Інтерфейс Користувача (“ГІК”):**Цей модуль буде відповідати за візуалізацію інтерфейсу користувача, отримання вхідних даних які наведені «у розділі 1.2» від користувача та відображення результатів.
   1. Він може використовувати бібліотеки GUI, наприклад, такі як Tkinter або його розширення CustomTkinter; для створення вікна, його віджетів та елементів керування. CustomTkinter дозволяє зробити сучасний та простий у використанні інтерфейс [8], [9].
   2. ГІК буде взаємодіяти з іншими модулями за допомогою інтерфейсів програмування API, наприклад запускаючи паралельний процес генерації анімації. Зв’язок між модулями можна подати за допомогою взаємно доступних файлів формату .json, .txt або .csv.
2. **Математична Анімація (“МА”):**
   1. Цей модуль буде відповідати за візуалізацію процесу ітерацій чисельного методу, зокрема методу Ньютона.
   2. Він буде використовувати бібліотеки візуалізації, наприклад, такі як Matplotlib або Manim, який дозволяє програмно створювати математичну анімацію та візуалізацію. Manim надає більше контролю над математичною анімацією [10], [11].
   3. МА буде отримувати на кожній ітерації дані від модуля ЧМ та генерувати анімацію.
3. **Чисельний метод (“ЧМ”):**
   1. Цей модуль буде відповідати за реалізацію чисельних методів, зокрема методу Ньютона для чисельного розв'язання нелінійних рівнянь.
   2. Він буде використовувати математичні бібліотеки, наприклад, такі як SymPy або NumPy, для виконання обчислень. SymPy є системою комп’ютерної алгебри, яка, дозволяє диференціювати функції [12].
   3. ЧМ буде надавати дані модулю МА та ГІК за допомогою спільних файлів.

**Взаємодія між модулями :**

Модулі будуть взаємодіяти один з одним за допомогою API.

1. **“ГІК”** буде отримувати від користувача вхідні дані та запускати процес роботи модуля “ЧМ”, передаючи їх за допомогою “config.json”.
2. **“ЧМ”** буде виконувати ітерації методом Ньютона та надавати результати модулю “МА” (наприклад, у файлі “config.json”) та ГІК (наприклад, у файлі “result.csv”).
3. **“МА”** буде візуалізувати процес ітерацій методом Ньютона, отримуючи дані від модуля “ЧМ” та повертаючи список адрес згенерованих математичних анімацій (наприклад у файлі **P**artial**M**ovie**F**ile**L**ist.txt).
4. **“ГІК”** буде відображати знайдене значення кореня та візуалізацію від модуля МА користувачеві (наприклад, зчитуючи файлPartialMovieFileList.txt).

Рис. 2.1 Схема взаємодії модулів

**Переваги модульної архітектури:**

1. **Модульність:** Розбиття програмного засобу на модулі робить його більш гнучким та керованим.
2. **Відновлюваність:** Кожен модуль можна тестувати та вдосконалювати незалежно.
3. **Повторне використання:** Модулі можна використовувати повторно в інших проєктах.
4. **Зниження складності:** Розбиття складної системи на менші, більш прості частини полегшує її розуміння та розробку.
5. **Масштабованість:** Збільшенняфункціональних можливостей одного модулю не впливає суттєво на роботу інших.

# 2.3 Математична модель алгоритму пошуку коренів нелінійних рівнянь

Модуль ЧМ має виконувати обчислення «відповідно до 1.2.1» результату нового наближення на кожній ітерації зчитуючи поточні вхідні дані з файлу.

**Алгоритм роботи модуля ЧМ:**

1. Зчитати поточні вхідні дані.
2. Виконати алгоритм обчислення відповідного чисельного методу на даній ітерації.
3. Занести зміни у файл конфігурації.
4. Повторювати дії починаючи з 1 кроку поки не відбудеться критерій зупинки. Критеріями зупинки є задана точність та кількість ітерацій.

На першій ітерації вхідні дані (конфігурації) отримуються з модуля ГІК за допомогою файлу “config.json”, в якому знаходяться усі необхідні початкові дані які наведені «у розділі 1.2», а також кількість ітерацій. Після зчитування даних відбувається ітерація обчислення згідно з відповідним алгоритмом «відповідно до 1.2.1 та формули 1.2.1.1» чисельного методу, результати якого заносяться у файл “config.json”, змінюючи попередні дані. Далі процес зчитує нові поточні дані та виконує ітерацію. Так процес виконується доти, доки не відбудеться критерій зупинки : буде досягнута задана точність «відповідно до формули 1.2.1.2» або не вичерпано кількість ітерацій.

Також модуль ЧМ може виконувати додаткові обчислення перевірки умов збіжності при заданих початкових даних або необхідні обчислення для МА модулю (наприклад, знаходження формули дотичної графіка в заданій точці), що своєю чергою полегшує роботу даного модуля.

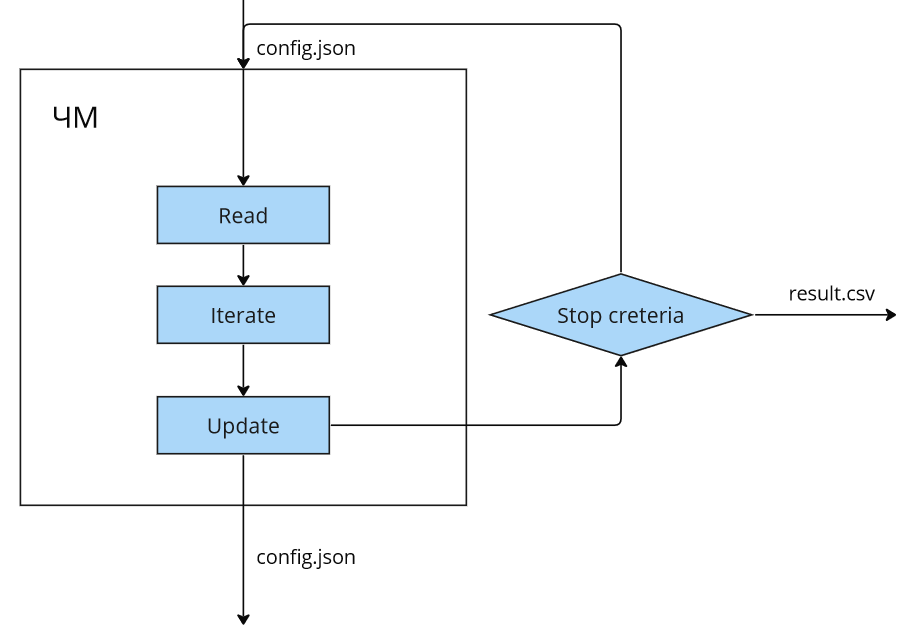


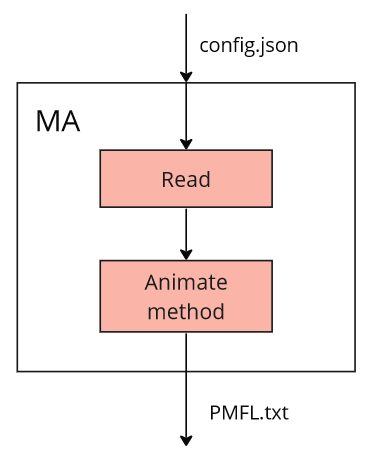
Рис. 2.2 Схема роботи модуля ЧМ

# 2.4 Створення графічного представлення алгоритму чисельного методу

Розробка графічного представлення алгоритму має керуватися графічною інтерпретацією відповідного чисельного методу.

**Алгоритм роботи модуля МА:**

1. Зчитати поточні вхідні дані.
2. Виконати генерацію графічного представлення алгоритму відповідного чисельного методу на даній ітерації.
3. Занести адреси створених відео у файл PMFL.txt
4. Повторювати дії починаючи з 1 кроку поки не відбудеться критерій зупинки. Критеріями зупинки є задана точність та кількість ітерацій.

Рис. 2.3 Схема роботи модуля МА

Кроки анімації «(рисунок 2.4)», методу Ньютона на кожній ітерації «відповідно до розділу 1.3»:

1. Намалювати координатну площину;
2. Побудувати графік функції f(x);
3. Відмітити точку (xi; f(xi)) на функції;
4. Побудувати дотичну до функції в точці xi;
5. Побудувати нову точку (xi+1; f(xi+1)) згідно з формулою;
6. Видалити точку (xi; f(xi)) та дотичну в цій точці;

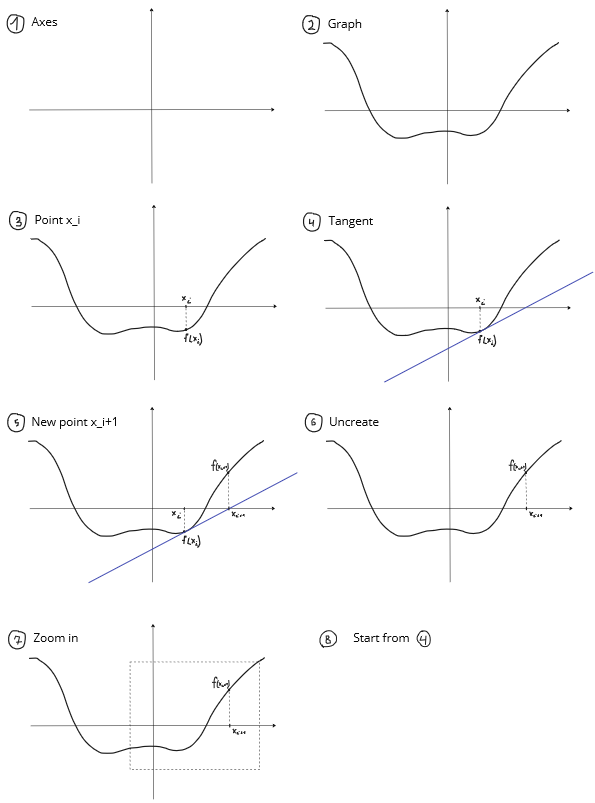


Рис. 2.4 Блок-схема кроків анімації методу Ньютона

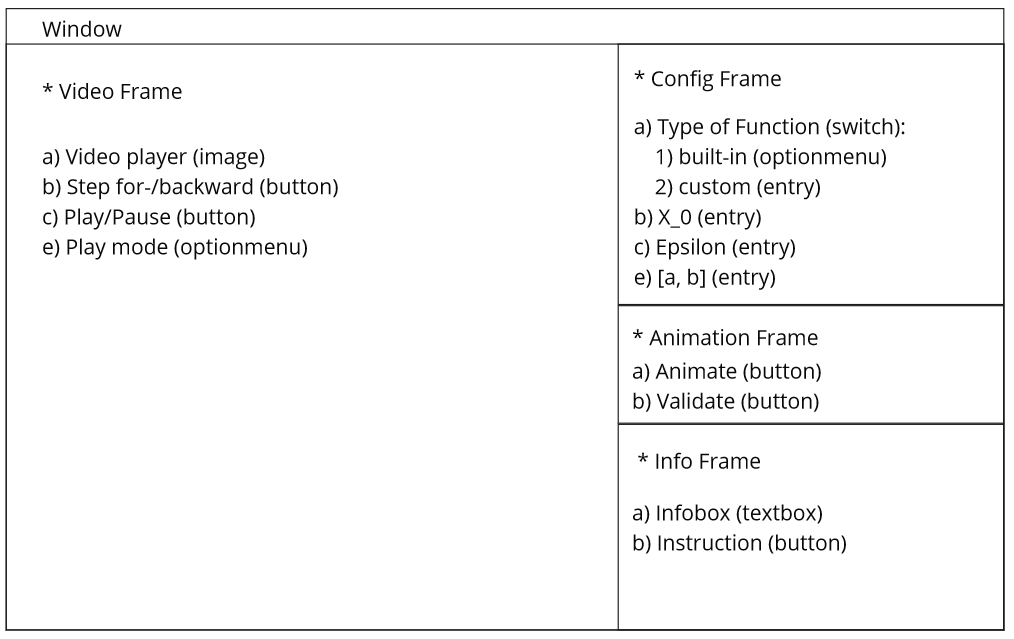
# 2.5 Розробка графічного інтерфейсу для роботи ПЗ

Модуль “ГІК” є сполучною ланкою між користувачем та модулями “МА” і “ЧМ”. З функціональних можливостей даного модуля можна виокремити 4 типи процесів:

1. Створення, введення початкових умов для модуля “ЧМ”.
2. Створення математичної анімації:
   1. Перевірка початкових умов;
   2. Запуск процесу генерації математичної анімації.
3. Відображення відео анімації.
4. Виведення текстових результатів роботи програми.

На основі даних процесів можна спроєктувати 4 фрейми «(рисунок 2.5)»: Vifeo Frame, Config Frame, Animate Frame, Info Frame.

| Назва фрейму | Опис фрейму |
| --- | --- |
| “Vifeo Frame” | Відображення відео у вікні, програвання та керування відео. |
| “Config Frame” | Налаштування початкових даних для чисельного методу. |
| “Animate Frame” | Перевірка та оновлення налаштування. Запуск процесу генерації анімації “GAP” |
| “Info Frame” | Вивід у вікно текстової інформації про роботу програми, а також додаткові відомості. |

****

**Рис. 2.5 Схема вікна програми та його елементів**

Візуальні компоненти фреймів це віджети: поля введення, кнопки дій, спадні меню, перемикачі, текстові поля виведення та відеопрогравач. Відеопрогравач у Video Frame можна організувати з використанням бібліотеки OpenCV [13], [14], для захоплення відео і відображення по кадрах відео у вікні програми графічного інтерфейсу користувача.

**Взаємодія з користувачем**

1. Користувач вводить початкові дані: нелінійне рівняння, початкове наближення та точність у “Config Frame”.
2. Користувач перевіряє введенні початкові дані, за потреби корегує.
3. Запускає процес генерації математичної анімації з відповідними початковими параметрами у “Animate Frame”, та чекає завантаження відео файлів, за потреби припиняє процес генерації.
4. Керує переглядом завантаженими відео в програвачі у “Video Frame”.
5. Переглядає результат роботи чисельного методу та текстову інформацію про стан роботи програми у “Info Frame”.

# 2.6 Висновки до розділу 2

У цьому розділі було описано проєктування програмного засобу для розв'язання нелінійних рівнянь методом Ньютона. Було поставлено задачу та описано мету проєктування, та можливості масштабування ПЗ до програмного застосунку пошуку коренів нелінійних рівнянь чисельними методами.

Визначено архітектуру та взаємодію модулів представлено їх схему. Розроблено кроки математичної анімації для графічного представлення алгоритму. Розроблено структуру графічного інтерфейсу користувача з чотирма фреймами, візуальні компоненти, взаємодію з користувачем.

# Перелік використаних джерел

* + 1. Чисельні методи : навчальний посібник / В. М. Задачин, І. Г. Конюшенко. – Х. : Вид. ХНЕУ ім. С. Кузнеця, 2014. – 180 с. (Укр. мов.)
    2. Youtube-джерело плейліст: Чисельні методи / канал LiudmylaVasylieva: <https://www.youtube.com/playlist?list=PL0lxPY6Csev3_PWZE9O4F4QcHuhPAsCtc>
    3. Ethernet-джерело стаття: ЧИСЕЛЬНІ МЕТОДИ РОЗВ'ЯЗАННЯ НЕЛІНІЙНИХ РІВНЯНЬ ТА СИСТЕМ НЕЛІНІЙНИХ РІВНЯНЬ: <https://web.posibnyky.vntu.edu.ua/fksa/14moskvina__komp_metod_dosl_analiz_danykh/lek4.htm>
    4. Чисельні методи: Навчальний посібник. / Волонтир Л.О, Зелінська О.В., Потапова Н.А., Чіков І.А., Вінницький національний аграрний університет. – Вінниця: ВНАУ, 2020 – 322 с.
    5. Чисельнi методи (для студентiв факультету комп’ютерних наук та кiбернетики, ОП ¾Системний аналiз¿): навчальний посiбник / Голубєва К.М., Кашпур О.Ф., Клюшин Д.А. – Київ: 2022. – 145 с.
    6. Ethernet-джерело стаття: Weisstein, Eric W. "Newton's Method." From *MathWorld*--A Wolfram Web Resource: <https://mathworld.wolfram.com/NewtonsMethod.html>
    7. Ethernet-джерело стаття: Розв'язування нелінійних рівнянь: <http://amc.ptngu.com/rozdil4.html>
    8. Документація бібліотеки CustomTkinter: <https://customtkinter.tomschimansky.com/documentation/>
    9. Youtube-джерело плейліст: Modern Tkinter Design With CustomTkinter/ канал Tkinter․com: <https://www.youtube.com/playlist?list=PLfZw_tZWahjxJl81b1S-vYQwHs_9ZT77f>
    10. Документація бібліотеки Manim: <https://docs.manim.community/en/stable/index.html#>
    11. Youtube-джерело плейліст : ManimCE Tutorials 2021/ канал Brian Amedee: <https://www.youtube.com/playlist?list=PLWOlLjdyZm2NQD1YZmEPB0dwbd0yKINAT>
    12. Документація бібліотеки SymPy: <https://docs.sympy.org/latest/index.html>
    13. Документація бібліотеки OpenCV: <https://opencv.org/>
    14. Youtube-джерело плейліст: OpenCV Python Tutorials/ канал Tech With Tim <https://youtube.com/playlist?list=PLzMcBGfZo4-lUA8uGjeXhBUUzPYc6vZRn&si=7BAG1p-EkXaiNH4o>