Звіт

Шевченко Максим, ШІ

Лабораторна робота 1

Варіант 1

Бібліотека для розв’язку нелінійних рівнянь

Реалізовані чисельні методи:

* Дотичних <https://en.wikipedia.org/wiki/Newton%27s_method>
* Дихотомії <https://en.wikipedia.org/wiki/Bisection_method>
* Хорд <https://en.wikipedia.org/wiki/Secant_method>

Реалізовані шаблони нелінійних рівнянь:

* a\*x + b\*cos(x) + c
* a\*x2 + b\*x + c
* a\*x \* eb\*x + c

Реалізація

Увесь програмний код можна знайти за посиланням <https://github.com/Gurdel/Magistracy/tree/main/Cross-platform%20and%20multiplatform%20techs/HW1%20Component%20dll>

Для створення бібліотеки була використана мова C#. Щоб створити бібліотеку, необхідно створити проект Class Library, написати код та скомпілювати проект. Реалізація можна переглянути в папці ClassLibrary, клас Solver. Скомпільована бібліотека знаходиться за шляхом ClassLibrary\ClassLibrary1\bin\Debug\netcoreapp3.1

Використання

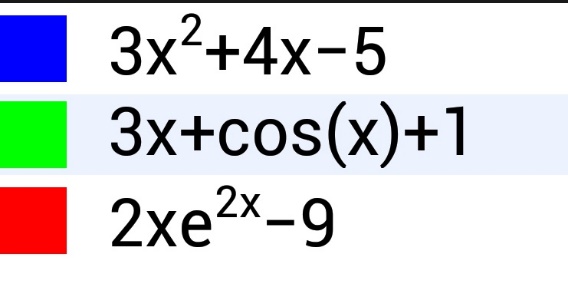
Щоб додати бібліотеку до проекту C# або Visual Basic, використовуючи IDE Visual Studio, необхідно перейти в Project -> Add Project Reference -> Browse, у діалоговому вікні перейти до розташування бібліотеки, обрати файл та натиснути Add. Приклад використання можна переглянути в папках Csharp і VisualBasic.

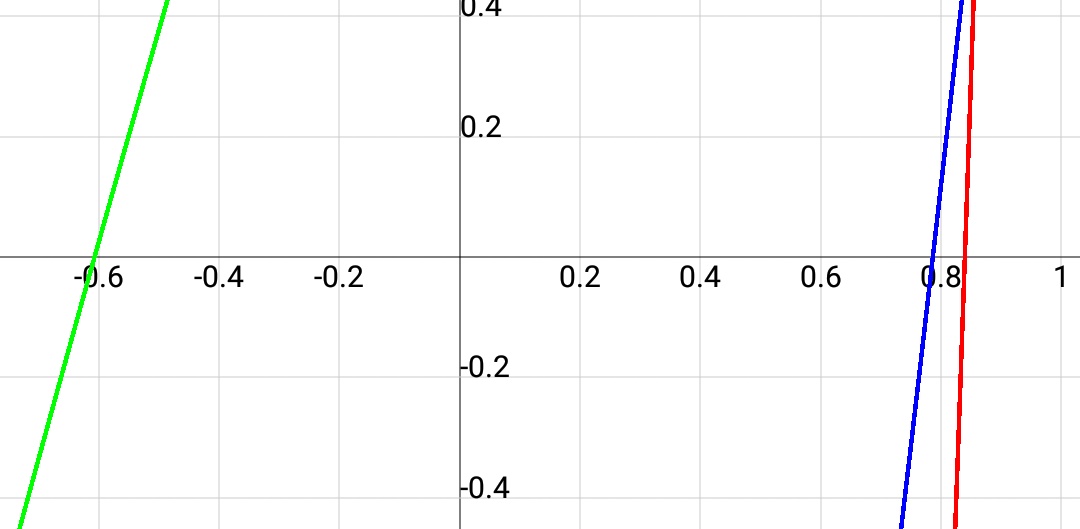
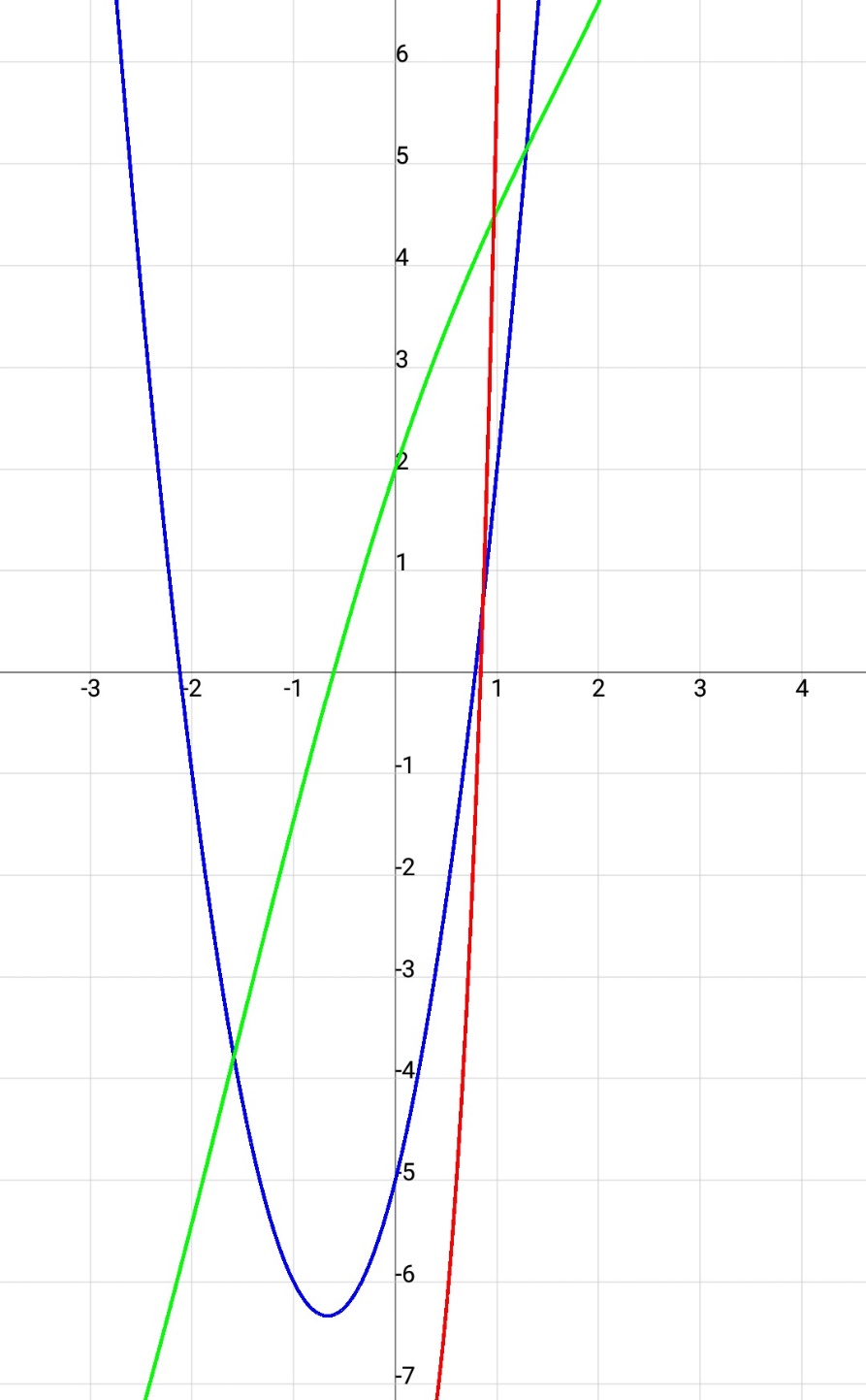
Щоб додати бібліотеку до проекту Python, необхідно використати бібліотеку pythonnet, модуль clr. Код знаходиться в папці Python.

Приклад роботи

Демонстрація роботи: силь на ютуб

Було розв’язано наступні рівняння на інтервалі [-2, 2] з точністю е=0.0001:





Виклик методів (C#, інші мови мають схожий синтаксис):

Newton(0, 0.0001, 1, new double[] { 3.0, 1.0, 1.0 });

Dichotomy(-1, 1, 0.0001, 1, new double[] { 3.0, 1.0, 1.0 });

Secant(-2, -1, 0.0001, 1, new double[] { 3.0, 1.0, 1.0 });

Newton(0, 0.0001, 2, new double[] { 3.0, 4.0, -5.0 });

Dichotomy(-2, 2, 0.0001, 2, new double[] { 3.0, 4.0, -5.0 });

Secant(0, 0.5, 0.0001, 2, new double[] { 3.0, 4.0, -5.0 });

Newton(0, 0.0001, 3, new double[] { 2.0, 2.0, -9.0 });

Dichotomy(-2, 2, 0.0001, 3, new double[] { 2.0, 2.0, -9.0 });

Secant(0, 0.5, 0.0001, 3, new double[] { 2.0, 2.0, -9.0 });

Вивід програми:

Newton (tangent) method

3x + 1cos(x) + 1

0) x = 0,0000000000 f = 2,0000000000

---

3) x = -0,6071016657 f = -0,0000000628

Dichotomy method

3x + 1cos(x) + 1

0) a = -1,0000000000 b = 1,0000000000 x = 0,0000000000 f = 2,0000000000

---

14) a = -0,6071777344 b = -0,6070556641 x = -0,6071166992 f = -0,0000537399

Secant (chord) method

3x + 1cos(x) + 1

0) x-2 = -2,0000000000 x-1 = -1,0000000000 x = -1,0000000000 f = -1,4596976941

---

3) x-2 = -0,6310586484 x-1 = -0,6080260581 x = -0,6071041732 f = -0,0000090159

Newton (tangent) method

3x^2 + 4x + -5

0) x = 0,0000000000 f = -5,0000000000

---

4) x = 0,7863000215 f = 0,0000032578

Dichotomy method

3x^2 + 4x + -5

0) a = -2,0000000000 b = 2,0000000000 x = 0,0000000000 f = -5,0000000000

---

17) a = 0,7862854004 b = 0,7863159180 x = 0,7863006592 f = 0,0000088166

Secant (chord) method

3x^2 + 4x + -5

0) x-2 = 0,0000000000 x-1 = 0,5000000000 x = 0,5000000000 f = -2,2500000000

---

4) x-2 = 0,7734806630 x-1 = 0,7857777284 x = 0,7863019608 f = 0,0000201640

Newton (tangent) method

2x \* exp(2x) + -9

0) x = 0,0000000000 f = -9,0000000000

--

13) x = 0,8395087419 f = 0,0000152809

Dichotomy method

2x \* exp(2x) + -9

0) a = -2,0000000000 b = 2,0000000000 x = 0,0000000000 f = -9,0000000000

---

16) a = 0,8394775391 b = 0,8395385742 x = 0,8395080566 f = -0,0000044015

Secant (chord) method

2x \* exp(2x) + -9

0) x-2 = 0,0000000000 x-1 = 0,5000000000 x = 0,5000000000 f = -6,2817181715

---

8) x-2 = 0,8367012979 x-1 = 0,8396026043 x = 0,8395078455 f = -0,0000104651

Вхідні дані можна вказати у файлі. Для цього у файлі необхідно вказати назву методу та параметри через пробіл і викликати функцію ProcessFiles, вказавши розташування вхідного та вихідного файлів. Приклад файлів: input.txt і output.txt.

Також можна отримати довідку про бібліотеку та реалізовані методи, викликавши метод Help:

Library for solving equations by numerical methods

Equations templates:

1. a\*x + b\*cos(x) + c

2. a\*x^2 + b\*x + c

3. a\*x \* exp(b\*x) + c

Methods:

1. Newton (tangent) method

2. Dichotomy method

3. Secant (chord) method

Common function parameters:

e: accuracy

template: number of equation template

args: list of equation parameters [a, b, c]

Newton(double x, double e, int template, double[] args)

x: init value

Dichotomy(double a, double b, double e, int template, double[] args)

a: left value of the interval in which the solution is located

b: right value of the interval in which the solution is located

Secant(double x\_2, double x\_1, double e, int template, double[] args)

x\_1: previous value of x (Xn-1)

x\_2: previous value of x\_1 (Xn-2)