DOI: https://doi.org/10.36910/6775-2524-0560-2026-63-11
УДК 519.6
Брагінець Оксана Вікторівна, к.ф.-м.н., доцент
https://orcid.org/0000-0003-2635-7203
Чорноморський національний університет ім. Петра Могили, м. Миколаїв, Україна

ЗАСТОСУВАННЯ MAPLE ДО РОЗВ'ЯЗУВАННЯ НЕЛІНІЙНИХ РІВНЯНЬ ЧИСЕЛЬНИМИ МЕТОДАМИ

Брагінець О.В. Застосування Maple до розв'язування нелінійних рівнянь чисельними методами. В роботі досліджено чисельні методи розв'язування нелінійних рівнянь, що мають широке застосування у математиці, фізиці та інженерії. Розглянуто класичні алгоритми – метод половинного поділу, метод хорд, метод Ньютона та метод простої ітерації. Для кожного методу наведено теоретичні основи, умови збіжності, переваги й недоліки. Особливу увагу приділено реалізації алгоритмів у системі Maple, що дозволяє поєднувати теоретичний аналіз із практичною демонстрацією. Показано приклади знаходження двох коренів конкретного рівняння як за допомогою власних програмних циклів, так і через вбудовані команди Maple (**fsolve, Bisection, Secant, Newton**). Здійснено детальний аналіз збіжності розглянутих чисельних методів та досліджено вплив вибору початкових наближень на ефективність обчислень. Встановлено, що метод Ньютона забезпечує найшвидшу збіжність за умови коректного вибору стартової точки, тоді як метод хорд демонструє збалансоване поєднання простоти реалізації та прийнятної швидкості. Метод дихотомії гарантує збіжність для будь-якої неперервної функції зі зміною знаку на кінцях інтервалу, що робить його універсальним, хоча й повільнішим. Метод простої ітерації потребує спеціального вибору функції, для правого та лівого коренів запропоновано різні варіанти, що забезпечують збіжність. Отримані результати підтверджують ефективність використання Maple як інструменту для навчання та наукових досліджень. У висновках узагальнено сильні та слабкі сторони методів, а також окреслено перспективи подальших досліджень: розширення аналізу на системи нелінійних рівнянь, дослідження умов збіжності для різних класів функцій та використання Maple для візуалізації процесів.
Ключові слова: нелінійні рівняння, метод дихотомії, метод хорд, метод Ньютона, метод простої ітерації, Maple, чисельні методи.

Brahinets O. Application of Maple to solving nonlinear equations by numerical methods. The paper investigates numerical methods for solving nonlinear equations that are widely used in mathematics, physics, and engineering. Classical algorithms are considered – the half-division method, the chord method, Newton's method, and the simple iteration method. For each method, theoretical foundations, convergence conditions, advantages, and disadvantages are given. Particular attention is paid to the implementation of algorithms in the Maple system, which allows combining theoretical analysis with practical demonstration. Examples of finding two roots of a specific equation are shown both using their own program cycles and using Maple's built-in commands (**fsolve, Bisection, Secant, Newton**). A detailed analysis of the convergence of the considered numerical methods is carried out and the influence of the choice of initial approximations on the efficiency of calculations is investigated. It is established that Newton's method provides the fastest convergence provided that the starting point is correctly chosen, while the chord method demonstrates a balanced combination of simplicity of implementation and acceptable speed. The dichotomy method guarantees convergence for any continuous function with a change of sign at the ends of the interval, which makes it universal, although slower. The simple iteration method requires a special choice of the function, and various options that ensure convergence are proposed for the right and left roots. The results obtained confirm the effectiveness of using Maple as a tool for teaching and scientific research. The conclusions summarize the strengths and weaknesses of the methods, and outline the prospects for further research: extending the analysis to systems of nonlinear equations, studying the convergence conditions for different classes of functions, and using Maple for process visualization.
Keywords: nonlinear equations, dichotomy method, chord method, Newton's method, simple iteration method, Maple, numerical methods.

Постановка проблеми.

Нелінійні рівняння відіграють ключову роль у сучасному математичному моделюванні, оскільки саме вони описують більшість реальних фізичних, технічних, біологічних та економічних процесів. На відміну від лінійних моделей, нелінійні залежності дозволяють відтворювати складні явища – від коливальних систем і хімічної кінетики до задач оптимізації та динаміки популяцій. Проте їхня аналітична розв'язність є винятком, а не правилом: у загальному випадку точний розв'язок або не існує, або має надто складний вигляд для практичного використання. Саме тому чисельні методи стають основним інструментом для знаходження коренів нелінійних рівнянь.

Основні труднощі розв'язування нелінійних рівнянь чисельними методами:

– відсутність універсального методу, який би гарантував розв'язок для всіх типів нелінійних рівнянь;
– залежність збіжності методів від вибору початкового наближення;
– необхідність контролю похибки та оцінки точності результатів;
– потреба у візуалізації процесу збіжності та аналізу поведінки функції.

Сучасні комп'ютерні математичні системи значно спрощують застосування чисельних алгоритмів, забезпечуючи високу точність, автоматизацію обчислень та можливість візуального аналізу. Одним із найпотужніших середовищ такого типу є Maple, який поєднує символьні та чисельні методи, підтримує широкий спектр алгоритмів для розв'язування нелінійних рівнянь і надає інструменти для аналізу збіжності, графічної інтерпретації та програмної реалізації власних методів. Такий підхід дозволяє показати практичну цінність Maple для ефективного навчання, досліджень та практичного застосування у різних галузях.

Аналіз досліджень.

Проблемою чисельного розв'язування нелінійних рівнянь є однією з ключових у чисельному аналізі, тому її розвиток простежується у великій кількості фундаментальних та сучасних досліджень. Класичні підходи до побудови ітераційних методів, аналізу їх збіжності та оцінки ефективності детально викладено у працях [1-3]. Значний внесок у розвиток теорії ітераційних методів подано у роботі [4], а концепцію оптимальних багатоточкових методів докладно розглянуто у дослідженні [5]. Сучасні тенденції розвитку високопорядкових ітераційних схем, їх оптимізації та розширення області збіжності представлені у спеціальному випуску журналу Mathematics (MDPI), присвяченому чисельним методам для нелінійних рівнянь [6].

Окремий напрям досліджень стосується застосування комп'ютерних математичних систем, зокрема Maple, для реалізації чисельних методів. У роботі [7] продемонстровано використання Maple для реалізації методів бісекції, Ньютона та січних, а також можливості графічного аналізу та дослідження збіжності. У статті [8] Maple застосовано для реалізації методу Адамяна при розв'язуванні нелінійних диференціальних рівнянь, що підтверджує ефективність поєднання символьних і чисельних засобів. Узагальнений огляд застосування Maple для розв'язування нелінійних диференціальних рівнянь подано у роботі [9]. Практичні аспекти використання Maple для розв'язування систем нелінійних рівнянь розглянуто у дослідженні [10].

В [11] розглянуто застосування математичного пакету Maple до розв'язування систем лінійних алгебраїчних рівнянь прямими чисельними методами. На прикладі програмного пакету Maple в [12] показано не тільки отримання чисельного розв'язку, а і його візуалізація у вигляді графіків, поверхонь, що допомагає в розумінні поведінки розв'язку та впливу різних крайових умов. Додаткові можливості середовища описано в офіційній документації Maplesoft [13, 14].

Попри наявність значної кількості робіт, у яких Maple використовується для демонстрації окремих методів або розв'язування специфічних задач, питання систематизованого порівняння класичних чисельних методів саме в середовищі , з аналізом збіжності, впливу початкових наближень та зручності реалізації, висвітлено недостатньо. Це визначає актуальність роботи.

Метою роботи є дослідження можливостей Maple у контексті чисельного розв'язання нелінійних рівнянь, порівняння ефективності різних методів та демонстрація практичних прикладів їх застосування.

Виклад основного матеріалу й обгрунтування отриманих результатів дослідження.

Розглядається рівняння вигляду:

$$f(x) = 0, \tag{1}$$

де $f(x)$ – нелінійна неперервна функція однієї змінної, тобто $f\colon R^1 ^\wedge R^1$.

У загальному випадку рівняння (1) може мати багато коренів. Чисельні методи розв'язання нелінійних рівнянь, які розглянуто далі, дозволяють знаходити один корінь на заданому відрізку $[a, b]$. При цьому на інтервалі повинен існувати тільки один корінь. Знайти відрізок, що задовольняє цю умову можна різними способами:

а) з фізичних міркувань, тобто на основі фізичних знань про задачу;

б) на основі досвіду розв'язання аналогічних задач;

в) за допомогою графічних методів;

г) шляхом відокремлення коренів.

Якщо функція $f(x)$ заздалегідь відома, то найбільш ефективним є графічний спосіб пошуку відрізка $[a, b]$. В інших випадках, коли відрізок $[a, b]$ треба найти автоматично (не візуально), то застосовують алгоритм відокремлення коренів.

В чисельних методах рівняння вважається розв'язаним, якщо всі корені знайдені з заданою точністю, тобто з наперед заданою похибкою. Знаходження наближених коренів рівняння звичайно складається з двох етапів:

1) відокремлення коренів, тобто знаходження для кожного з них відрізку ізоляції;

2) обчислення кореня у відрізку ізоляції з наперед заданою точністю.

Графічний метод. Графічний метод відокремлення кореня полягає у попередньому аналізі поведінки функції $f(x)$ на певному інтервалі з метою визначення ділянок, на яких функція змінює знак, а отже – гарантовано має корінь. Цей метод не дає точного значення кореня, але забезпечує початкову інформацію, необхідну для подальшого застосування чисельних методів (методу Ньютона, січних, простої ітерації тощо).

Ідея методу полягає в тому, що якщо функція неперервна на відрізку $[a, b]$ і виконується умова

$$f(a) \cdot f(b) < 0,$$

то, згідно з теоремою Больцано, на цьому відрізку існує принаймні один корінь рівняння (1). Графічний метод дозволяє візуально знайти такі відрізки, тобто побачити, де крива перетинає вісь Ox.

Переваги графічного способу:

- дозволяє гарантовано знайти відрізок, де є корінь;
- допомагає уникнути помилок при виборі початкових наближень;
- дає змогу виявити усі корені на заданому проміжку;
- забезпечує наочність і інтуїтивне розуміння поведінки функції;
- є універсальним і застосовується до будь-яких неперервних функцій.

Розглянемо реалізацію цього методу у Maple на прикладі рівняння

$$x^2 - 2 + e^{-x} = 0. \quad (2)$$

Задаємо функцію та будуємо її графік:

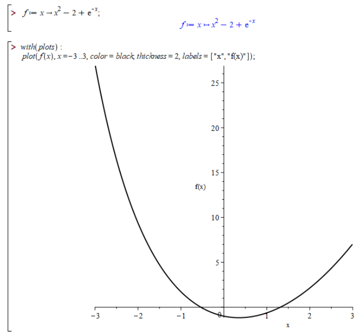


Рис. 1. Графічне відокремлення коренів рівняння (2) у Maple.

Графік показує два перетини з віссю Ox на проміжках $[-1{,}0]$ та $[1{,}2]$, тобто рівняння має два корені. У Maple команда **fsolve** призначена для чисельного розв’язування рівнянь та систем рівнянь. На відміну від **solve**, яка шукає аналітичний розв’язок, **fsolve** повертає наближене числове значення кореня. Ця команда може знаходити один корінь за один виклик, використовуючи внутрішні алгоритми (модифікований Ньютона, комбіновані методи), тому важливо задавати інтервал, на якому треба відшукати розв’язок, якщо рівняння має не один корінь.

Оскільки рівняння (2) має два корені, тому використовуємо два виклики:

```
> x1 := fsolve(f(x), x = -1 ..0);
  x2 := fsolve(f(x), x = 1 ..2);
                                  x1 := -0.5372744492
                                  x2 := 1.315973778
```

Використання інтервалів, отриманих графічним методом, забезпечує стабільність та точність чисельного розв’язання. Далі продемонструємо застосування чотирьох класичних методів для знаходження коренів на цих інтервалах.

Метод дихотомії (бісекції). Метод дихотомії ще називають методом половинного поділу. При розв'язанні нелінійного рівняння методом половинного поділу задаються відрізок $[a, b]$, на якому існує лише один розв'язок, і бажана точність $\varepsilon > 0$ розв'язання задачі. Він ґрунтується на послідовному діленні відрізка навпіл. Якщо на кінцях відрізка функція має різні знаки, то корінь міститься в одній із половин. Повторюємо, доки довжина інтервалу не стане меншою за задану похибку.

Метод дихотомії в Maple можна реалізувати двома способами:

- власна реалізація, тобто написання процедури, яка крок за кроком виконує алгоритм ітерацій, що корисно для навчання та демонстрації принципу роботи методу;

– використання готової функції **Bisection** з пакета **Student[NumericalAnalysis]**, що дозволяє швидко отримати результат із заданою точністю.

Покажемо обидва варіанти. Власна реалізація виглядає так:

```
> f := x → x^2 − 2 + exp(−x);

  a := −1.0;
  b := 0.0;
  tol := 10^(−5);
  iter := 0 :

  while (b − a) > tol do
    iter := iter + 1 :
    c := (a + b)/2 :
    printf("Ітерація %d: a = %.6f, b = %.6f, c = %.6f, f(c) = %.6f\n",
        iter, a, b, c, f(c));
    if f(a) * f(c) < 0 then
      b := c :
    else
      a := c :
    end if;
  end do:

  printf("Наближений корінь 1: x1 = %.5f\n", c);
```

$$f := x \mapsto x^2 - 2 + e^{-x}$$
$$a := -1.0$$
$$b := 0.$$
$$tol := \frac{1}{100000}$$

```
Ітерація 1: a = -1.000000, b = 0.000000, c = -0.500000, f(c) = -0.101279
Ітерація 2: a = -1.000000, b = -0.500000, c = -0.750000, f(c) = 0.679500
Ітерація 3: a = -0.750000, b = -0.500000, c = -0.625000, f(c) = 0.258871
Ітерація 4: a = -0.625000, b = -0.500000, c = -0.562500, f(c) = 0.071461
Ітерація 5: a = -0.562500, b = -0.500000, c = -0.531250, f(c) = -0.016716
Ітерація 6: a = -0.562500, b = -0.531250, c = -0.546875, f(c) = 0.026917
Ітерація 7: a = -0.546875, b = -0.531250, c = -0.539062, f(c) = 0.004987
Ітерація 8: a = -0.539062, b = -0.531250, c = -0.535156, f(c) = -0.005893
Ітерація 9: a = -0.539062, b = -0.535156, c = -0.537109, f(c) = -0.000460
Ітерація 10: a = -0.539062, b = -0.537109, c = -0.538086, f(c) = 0.002262
Ітерація 11: a = -0.538086, b = -0.537109, c = -0.537598, f(c) = 0.000901
Ітерація 12: a = -0.537598, b = -0.537109, c = -0.537354, f(c) = 0.000220
Ітерація 13: a = -0.537354, b = -0.537109, c = -0.537231, f(c) = -0.000120
Ітерація 14: a = -0.537354, b = -0.537231, c = -0.537292, f(c) = 0.000050
Ітерація 15: a = -0.537292, b = -0.537231, c = -0.537262, f(c) = -0.000035
Ітерація 16: a = -0.537292, b = -0.537262, c = -0.537277, f(c) = 0.000008
Ітерація 17: a = -0.537277, b = -0.537262, c = -0.537270, f(c) = -0.000014
Наближений корінь 1: x1 = -0.53727
```

```
> f := x → x^2 − 2 + exp(−x);
  a := 1.0;
  b := 2.0;
  tol := 10^(−5);
  iter := 0 :

  while (b − a) > tol do
    iter := iter + 1 :
    c := (a + b)/2 :
    printf("Ітерація %d: a = %.6f, b = %.6f, c = %.6f, f(c) = %.6f\n",
        iter, a, b, c, f(c));
    if f(a) * f(c) < 0 then
      b := c :
    else
      a := c :
    end if;
  end do:

  printf("Наближений корінь 2: x2 = %.5f\n", c);
```

$$f := x \mapsto x^2 - 2 + e^{-x}$$
$$a := 1.0$$
$$b := 2.0$$
$$tol := \frac{1}{100000}$$

```
Ітерація 1: a = 1.000000, b = 2.000000, c = 1.500000, f(c) = 0.473130
Ітерація 2: a = 1.000000, b = 1.500000, c = 1.250000, f(c) = -0.150995
Ітерація 3: a = 1.250000, b = 1.500000, c = 1.375000, f(c) = 0.143465
Ітерація 4: a = 1.250000, b = 1.375000, c = 1.312500, f(c) = -0.008197
Ітерація 5: a = 1.312500, b = 1.375000, c = 1.343750, f(c) = 0.066530
Ітерація 6: a = 1.312500, b = 1.343750, c = 1.328125, f(c) = 0.028890
Ітерація 7: a = 1.312500, b = 1.328125, c = 1.320312, f(c) = 0.010277
Ітерація 8: a = 1.312500, b = 1.320312, c = 1.316406, f(c) = 0.001022
Ітерація 9: a = 1.312500, b = 1.316406, c = 1.314453, f(c) = -0.003592
Ітерація 10: a = 1.314453, b = 1.316406, c = 1.315430, f(c) = -0.001286
Ітерація 11: a = 1.315430, b = 1.316406, c = 1.315918, f(c) = -0.000132
Ітерація 12: a = 1.315918, b = 1.316406, c = 1.316162, f(c) = 0.000445
Ітерація 13: a = 1.315918, b = 1.316162, c = 1.316040, f(c) = 0.000157
Ітерація 14: a = 1.315918, b = 1.316040, c = 1.315979, f(c) = 0.000012
Ітерація 15: a = 1.315918, b = 1.315979, c = 1.315948, f(c) = -0.000060
Ітерація 16: a = 1.315948, b = 1.315979, c = 1.315964, f(c) = -0.000024
Ітерація 17: a = 1.315964, b = 1.315979, c = 1.315971, f(c) = -0.000006
Наближений корінь 2: x2 = 1.31597
```

Рис. 2. Програмна реалізація методу дихотомії у Maple.

На кожному кроці надруковано номер ітерації, ліва межа a, права межа b, середина інтервалу c, значення $f(c)$. Наприкінці виводить наближене значення кореня. Такий варіант гарно ілюструє збіжність методу бісекції. Ми бачимо як інтервал звужується, а середина поступово сходиться до кореня.

Використовуючи вбудовану функцію **Bisection,** отримаємо:

```
> with(Student[NumericalAnalysis]) :
> x_1 := Bisection(f(x), x = [−1, 0], tolerance = 10^(−5));
```

$$x\_1 := -0.5372734069$$

```
> x_1 := Bisection(f(x), x = [−1, 0], tolerance = 10^(−5), output = information, stoppingcriterion = absolute, maxiterations = 20);
```

$x\_1 :=$

n	a_n	b_n	p_n	$f(p_n)$	absolute error
1	−1.	0.	−0.5000000000	−0.101278729	0.5000000000
2	−1.	−0.5000000000	−0.7500000000	0.679500017	0.2500000000
3	−0.7500000000	−0.5000000000	−0.6250000000	0.258870957	0.1250000000
4	−0.6250000000	−0.5000000000	−0.5625000000	0.071460907	0.0625000000
5	−0.5625000000	−0.5000000000	−0.5312500000	−0.016716136	0.0312500000
6	−0.5625000000	−0.5312500000	−0.5468750000	0.026917323	0.0156250000
7	−0.5468750000	−0.5312500000	−0.5390625000	0.004987239	0.0078125000
8	−0.5390625000	−0.5312500000	−0.5351562500	−0.005892736	0.0039062500
9	−0.5390625000	−0.5351562500	−0.5371093750	−0.000459827	0.0019531250
10	−0.5390625000	−0.5371093750	−0.5380859375	0.002261935	0.0009765625
11	−0.5380859375	−0.5371093750	−0.5375976563	0.000900612	0.0004882812
12	−0.5375976563	−0.5371093750	−0.5373535157	0.000220282	0.0002441406
13	−0.5373535157	−0.5371093750	−0.5372314453	−0.000119800	0.0001220704
14	−0.5373535157	−0.5372314453	−0.5372924804	0.000050234	0.0000610353
15	−0.5372924804	−0.5372314453	−0.5372619628	−0.000034785	0.0000305176
16	−0.5372924804	−0.5372619628	−0.5372772216	7.724×10^{-6}	0.0000152588
17	−0.5372772216	−0.5372619628	−0.5372695922	−0.000013531	7.6294×10^{-6}

```
> x_2 := Bisection(f(x), x = [1, 2], tolerance = 10^(−5));
```

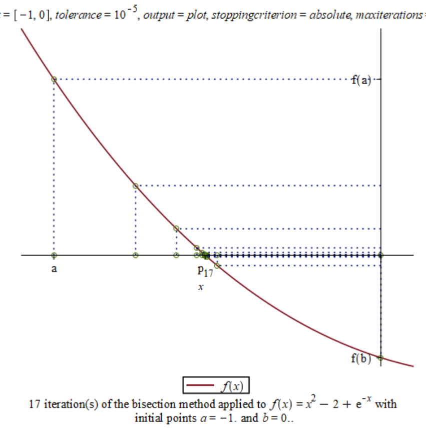
$$x\_2 := 1.315971374$$

```
> x_2 := Bisection(f(x), x = [1, 2], tolerance = 10^(−5), output = information, stoppingcriterion = absolute, maxiterations = 20);
```

$x\_2 :=$

n	a_n	b_n	p_n	$f(p_n)$	absolute error
1	1.	2.	1.500000000	0.4731301601	0.500000000
2	1.	1.500000000	1.250000000	−0.1509952031	0.250000000
3	1.250000000	1.500000000	1.375000000	0.1434645958	0.125000000
4	1.250000000	1.375000000	1.312500000	−0.0081974013	0.062500000
5	1.312500000	1.375000000	1.343750000	0.0665296481	0.031250000
6	1.312500000	1.343750000	1.328125000	0.0288896374	0.015625000
7	1.312500000	1.328125000	1.320312500	0.0102769332	0.007812500
8	1.312500000	1.320312500	1.316406250	0.0010224616	0.003906250
9	1.312500000	1.316406250	1.314453125	−0.0035917967	0.001953125
10	1.314453125	1.316406250	1.315429688	−0.0012857482	0.000976563
11	1.315429688	1.316406250	1.315917969	−0.0001319138	0.000488281
12	1.315917969	1.316406250	1.316162110	0.0004452078	0.000244141
13	1.315917969	1.316162110	1.316040040	0.0001566314	0.000122071
14	1.315917969	1.316040040	1.315979004	0.0000123534	0.000061035
15	1.315917969	1.315979004	1.315948486	−0.0000597822	0.000030517
16	1.315948486	1.315979004	1.315963745	−0.0000237149	0.000015259
17	1.315963745	1.315979004	1.315971374	-5.6821×10^{-6}	7.629×10^{-6}

```
> x_1 := Bisection(f(x), x = [−1, 0], tolerance = 10^(−5), output = plot, stoppingcriterion = absolute, maxiterations = 20);
```




```
> x_2 := Bisection(f(x), x = [1, 2], tolerance = 10^(−5), output = plot, stoppingcriterion = absolute, maxiterations = 20);
```

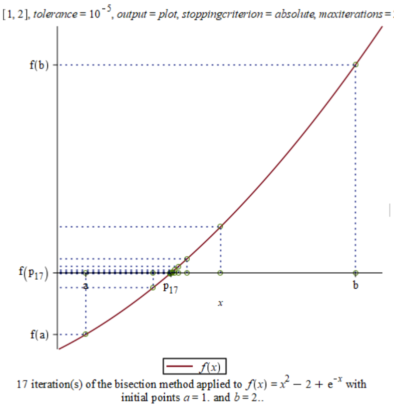


Рис. 3. Візуалізація методу дихотомії через вбудовану команду **Bisection** у Maple.

Переваги вбудованої команди в тому, що не потрібно писати власний код, а достатньо викликати готову функцію, можна легко змінювати параметри точності та інтервал, Maple автоматично контролює похибку та кількість ітерацій. Результат може бути виведений в різному вигляді: тільки корінь або повна таблиця збіжності, або графічна візуалізація, що продемонстровано на рис.3.

Порівнюючи наведені способи реалізації методу половинного поділу у Maple, видно, що бажана точність досягається за однакове число ітерацій 17.

Метод хорд. Метод хорд ще називають методом січних. Він базується на побудові хорди між двома точками $(a, f(a))$ та $(b, f(b))$ і знаходженні точки перетину цієї хорди з віссю Ox. Формула для нового наближення:

$$x_{k+1} = x_k - \frac{f(x_k)}{f(x_k) - f(c)}(x_k - c), \qquad k = 0, 1, 2, \ldots, \tag{3}$$

де $c \in [a; b]$.

Як і з методом бісекції продемонструємо реалізацію методу хорд двома способами у Maple. Власна реалізація виглядає так:

```
> f := x -> x^2 - 2 + exp(-x);
  a := -1.0;
  b := 0.0;
  iter := 0 :
  epsilon := 10^(-5);

  while (b - a) > epsilon do
    iter := iter + 1;
    c := b - f(b)*(b - a)/(f(b) - f(a));
     if f(c)*f(a) < 0 then
      b := c;
    else
      a := c;
    end if:
  end do:

  printf("Наближений корінь 1: x1 = %.7f (за %d ітерацій)\n", c, iter);
```

$$f := x \mapsto x^2 - 2 + e^{-x}$$
$$a := -1.0$$
$$b := 0.$$
$$\epsilon := \frac{1}{100000}$$

```
Наближений корінь 1: x1 = -0.5372744 (за 17 ітерацій)
```

```
> f := x -> x^2 - 2 + exp(-x);
  a := 1.0;
  b := 2.0;
  iter := 0 :
  epsilon := 10^(-5);

  while (b - a) > epsilon do
    iter := iter + 1;
    c := b - f(b)*(b - a)/(f(b) - f(a));
     if f(c)*f(a) < 0 then
      b := c;
    else
      a := c;
    end if:
  end do:

  printf("Наближений корінь 2: x2 = %.7f (за %d ітерацій)\n", c, iter);
```

$$f := x \mapsto x^2 - 2 + e^{-x}$$
$$a := 1.0$$
$$b := 2.0$$
$$\epsilon := \frac{1}{100000}$$

```
Наближений корінь 2: x2 = 1.3159738 (за 15 ітерацій)
```

Рис. 4. Програмна реалізація методу хорд у Maple.

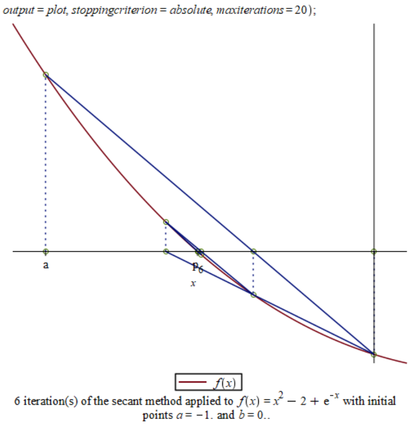
На кожному кроці перевіряється умова $f(c) \cdot f(a) < 0$, яка дозволяє звузити інтервал так, щоб він містив корінь. Завжди залишаємо той кінець інтервалу, де знак функції відрізняється від знаку в точці c. Це гарантує, що корінь залишиться всередині нового відрізка. Наприкінці виводить наближене значення кореня.

Використовуючи вбудовану команду **Secant** з пакета **Student[NumericalAnalysis]**, отримаємо:

```
> with(Student[NumericalAnalysis]) :
  x_1 := Secant(f(x), x = [-1, 0], tolerance = 10^(-5));
```

$$x\_1 := -0.5372744468$$

```
> Secant(f(x), x = [-1, 0], output = plot, stoppingcriterion = absolute, maxiterations = 20);
```




```
> x_2 := Secant(f(x), x = [1, 2], tolerance = 10^(-5));
```

$$x\_2 := 1.315973773$$

```
> x_2 := Secant(f(x), x = [1, 2], tolerance = 10^(-5), output = plot, stoppingcriterion = absolute, maxiterations = 20);
```

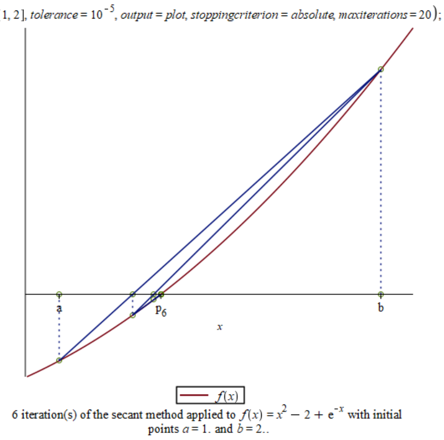


Рис. 5. Візуалізація методу хорд через вбудовану команду **Secant** у Maple.

Власна реалізація дає змогу зрозуміти алгоритм методу покроково, а вбудована команда **Secant** зручна у простоті використання та швидкого отримання результатів у практичних задачах.

Метод Ньютона. Ідея методу полягає в послідовній заміні ділянки кривої $f(x)$ дотичною в точці c, що належить відрізку $[a;b]$ і перетинає вісь Ox в точці x_k. Наступне наближення розв'язку x_{k+1} визначається таким чином. У точці $\left(x_k, f(x_k)\right)$ проводиться дотична до графіка $f(x)$ і визначається точка x_{k+1} як точка перетину дотичної з віссю абсцис. Пошук розв'язку припиняється при досягненні заданої точності $|x_{k+1} - x_k| < \varepsilon$ або $|f(x_k)| < \varepsilon$.

Для визначення точки перетину $(k+1)$-ї дотичної з віссю абсцис користуються формулою:

$$x_{k+1} = x_k - \frac{f(x_k)}{f'(x_k)}, \quad (k = 0,1,2,\dots). \tag{4}$$

За початкове наближення краще брати точку x_0, у якій виконується умова $f(x_0) \cdot f''(x_0) > 0$. Це означає, що функція і її кривизна «дивляться» в один бік, і дотична веде до кореня. Зазвичай за x_0 беруть один із кінців відрізку ізоляції, міркуючи наступним чином:

– якщо $f(a) \cdot f''(a) > 0$, то початкове наближення беруть $x_0 = a$;
– якщо $f(b) \cdot f''(b) > 0$, то початкове наближення беруть $x_0 = b$.

```
> f := x -> x^2 - 2 + exp(-x);
  df := x -> 2*x - exp(-x); # перша похідна
  d2f := x -> 2 + exp(-x);  # друга похідна

  a := -1.0;
  b := 0.0;
  epsilon := 1e-5;
  iter := 0 :
  maxiter := 50 :

  # Вибір початкового наближення
  if f(a) * d2f(a) > 0 then
    x := a;
  elif f(b) * d2f(b) > 0 then
    x := b;
  end if:

  printf("Початкове наближення x0 = %.5f\n", x);

  # Метод Ньютона
  while iter < maxiter do
    iter := iter + 1 :
    xnew := x - f(x) / df(x) :
    printf("Ітерація %d: x = %.7f, f(x) = %.7f\n", iter, xnew, f(xnew));
    if abs(xnew - x) < epsilon or abs(f(xnew)) < epsilon then
      break;
    end if;
    x := xnew;
  end do:

  printf("Наближений корінь 1: x1 = %.5f (за %d ітерацій)\n", xnew, iter);

                    f := x ↦ x^2 - 2 + e^(-x)
                    df := x ↦ 2·x - e^(-x)
                    d2f := x ↦ 2 + e^(-x)
                    a := -1.0
                    b := 0.
                    ϵ := 0.00001
Початкове наближення x0 = -1.00000
Ітерація 1: x = -0.6358247, f(x) = 0.2928520
Ітерація 2: x = -0.5431567, f(x) = 0.0164515
Ітерація 3: x = -0.5372974, f(x) = 0.0000638
Ітерація 4: x = -0.5372744, f(x) = 0.0000000
Наближений корінь 1: x1 = -0.53727 (за 4 ітерацій)
```

```
> f := x -> x^2 - 2 + exp(-x);
  df := x -> 2*x - exp(-x); # перша похідна
  d2f := x -> 2 + exp(-x);  # друга похідна

  a := 1.0;
  b := 2.0;
  epsilon := 1e-5;
  iter := 0 :
  maxiter := 50 :

  # Вибір початкового наближення
  if f(a) * d2f(a) > 0 then
    x := a;
  elif f(b) * d2f(b) > 0 then
    x := b;
  end if:

  printf("Початкове наближення x0 = %.5f\n", x);

  # Метод Ньютона
  while iter < maxiter do
    iter := iter + 1 :
    xnew := x - f(x) / df(x) :
    printf("Ітерація %d: x = %.7f, f(x) = %.7f\n", iter, xnew, f(xnew));
    if abs(xnew - x) < epsilon or abs(f(xnew)) < epsilon then
      break;
    end if;
    x := xnew;
  end do:

  printf("Наближений корінь 2: x2 = %.5f (за %d ітерацій)\n", xnew, iter);

                    f := x ↦ x^2 - 2 + e^(-x)
                    df := x ↦ 2·x - e^(-x)
                    d2f := x ↦ 2 + e^(-x)
                    a := 1.0
                    b := 2.0
                    ϵ := 0.00001
Початкове наближення x0 = 2.00000
Ітерація 1: x = 1.4474720, f(x) = 0.3303394
Ітерація 2: x = 1.3232741, f(x) = 0.0173163
Ітерація 3: x = 1.3159992, f(x) = 0.0000600
Ітерація 4: x = 1.3159738, f(x) = 0.0000000
Наближений корінь 2: x2 = 1.31597 (за 4 ітерацій)
```

Рис. 6. Програмна реалізація методу Ньютона у Maple.

Використовуючи вбудовану команду **Newton** з пакета **Student[NumericalAnalysis]**, отримаємо:

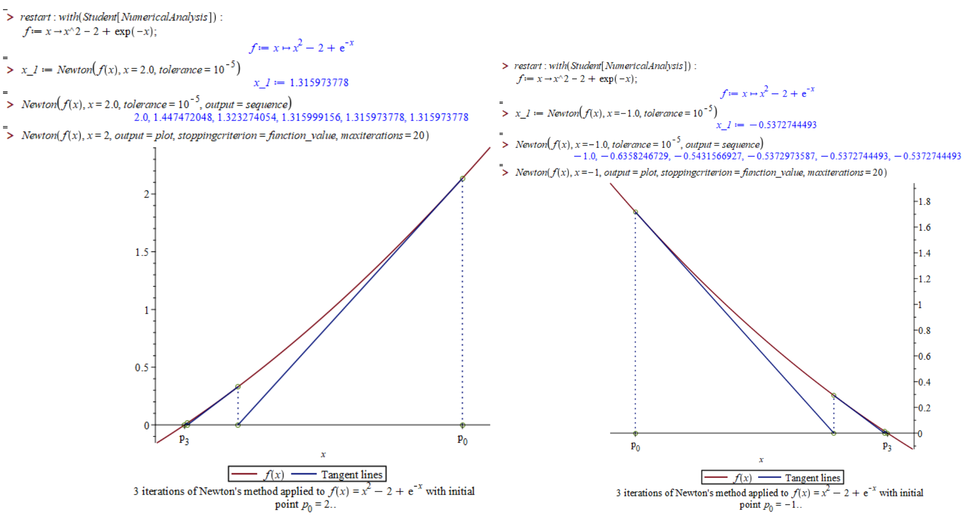


Рис. 7. Візуалізація методу Ньютона через вбудовану команду **Newton** у Maple.

Використовуючи команду **Newton,** треба самостійно задати початкове наближення, не відрізок ізоляції, як у попередніх методах. Щоб правильно вибрати один із кінців інтервалу треба в точках a і b перевірити виконання умови $f(x_0)\cdot f''(x_0) > 0$. Саме ця умова забезпечує швидку збіжність методу.

Отже, у власній реалізації початкове наближення може бути вибране програмою автоматично за умовою збіжності, а у вбудованій команді **Newton** користувач повинен сам обрати цю точку, перевіривши необхідну умову, що не зручно. Але можна написати код для вибору точки x_0, а розв’язати рівняння через **Newton**.

Метод простої ітерації. Метод простої ітерації ґрунтується на перетворенні рівняння $f(x) = 0$ у вигляд $x = g(x)$ та побудові послідовності $x_{n+1} = g(x_n), n = 0,1,2, \ldots$. Збіжність методу залежить від вибору функції $g(x)$ та початкового наближення. Метод збігається, якщо на інтервалі, де шукається корінь, виконується умова $|g'(x)| < 1$. Це гарантує, що послідовність $\{x_n\}$ є збіжною.

Для рівняння (2) існує два корені різних знаків, тому для відшукання їх методом простої ітерації однозначне представлення рівнянням $x = g(x)$ неможливе. Для від’ємного кореня, тобто кореня з відрізка $[-1{,}0]$ маємо:

$$x = -\ln(2 - x^2),$$

оскільки логарифм працює з додатними аргументами. А для кореня з відрізка $[1{,}2]$ доцільно вибрати наступну функцію:

$$x = \sqrt{2 - e^{-x}}.$$

Як і в попередніх методах пошук розв'язку припиняється при досягненні заданої точності $|x_{k+1} - x_k| < \varepsilon$ або $|f(x_k)| < \varepsilon$.

```
> f := x → x^2 − 2 + exp(−x);

 # Перетворення для простої ітерації
 g := x → −ln(2 − x^2);

 x := −1.0;    # початкове наближення в інтервалі [-1,0]
 epsilon := 1e−5;
 iter := 0 :
 maxiter := 50 :

 while iter < maxiter do
   iter := iter + 1 :
   xnew := g(x) :
  # printf("Ітерація %d: x = %.7f\n", iter, xnew) :
   if abs(xnew − x) < epsilon then
     break
   end if:
   x := xnew :
 end do:

 printf("Наближений корінь 1: x1 = %.5f (за %d ітерацій)\n", xnew, iter);

                    f := x ↦ x^2 − 2 + e^-x
                    g := x ↦ −ln(2 − x^2)
                         x := −1.0
                       ε := 0.00001
Наближений корінь 1: x1 = -0.53727 (за 25 ітерацій)
```

```
> f := x → x^2 − 2 + exp(−x);

 # Перетворюємо рівняння f(x)=0 у вигляд x=g(x)
 g := x → sqrt(2 − exp(−x));

 x := 1.0;    # початкове наближення
 epsilon := 1e−5;
 iter := 0 :
 maxiter := 50 :

 while iter < maxiter do
   iter := iter + 1;
   xnew := g(x);
 #   printf("Ітерація %d: x = %.7f\n", iter, xnew);
   if abs(xnew − x) < epsilon then
     break;
   end if;
   x := xnew;
 end do:

 printf("Наближений корінь 2: x2 = %.5f (за %d ітерацій)\n", xnew, iter);

                    f := x ↦ x^2 − 2 + e^-x
                    g := x ↦ sqrt(2 − e^-x)
                         x := 1.0
                       ε := 0.00001
Наближений корінь 2: x2 = 1.31597 (за 6 ітерацій)
```

Рис. 8. Програмна реалізація методу простої ітерації у Maple.

Висновки та перспективи подальшого дослідження. Розв’язування нелінійних рівнянь є однією з ключових задач прикладної математики, і жоден метод не є універсальним. В роботі було проведено повний аналіз рівняння (2) із застосуванням графічних та чисельних методів, а також реалізовано всі алгоритми в Maple. Було реалізовано та досліджено чотири класичні методи.

Метод дихотомії (бісекції) гарантує збіжність за умови зміни знаку на кінцях інтервалу, показав стабільну роботу на обох інтервалах, є найповільнішим, але найнадійнішим методом. Метод хорд не потребує похідної, проте залежить від вибору інтервалу. Він збігається швидше за дихотомію і коректно працює на обох інтервалах при правильному виборі початкових точок. Метод Ньютона забезпечує найшвидшу збіжність, проте потребує похідної та вдалого початкового наближення. Метод простої ітерації є найпростішим для реалізації, проте його збіжність залежить від правильного вибору функції $g(x)$. Він добре ілюструє у навчальних цілях ідею послідовних наближень, але для практичних задач частіше застосовують метод Ньютона чи хорд. Система Maple дозволяє як реалізувати алгоритми вручну, так і використовувати вбудовані команди, що робить її ефективним інструментом для навчання та досліджень.

Надалі планується розширення аналізу на системи нелінійних рівнянь, де методи ітерації застосовуються у багатовимірному просторі, дослідження умов збіжності для різних класів функцій та побудова оптимальних перетворень $g(x)$. Показати використання Maple для візуалізації процесу збіжності та аналізу поведінки функцій. Планується розробка навчальних модулів з курсу «Чисельні методи» у Maple, які демонструють покрокову роботу кожного методу для студентів.

Список бібліографічного опису

1. Burden R.L., Faires J.D. (2015) Numerical Analysis. 10th ed. Boston: Cengage Learning, 888 p. DOI: 10.13140/2.1.4830.2406
2. Ralston A., Rabinowitz P. (2001) A First Course in Numerical Analysis. 2nd ed. New York: Dover Publications, 572 p.
3. Atkinson K. E. (1989) An Introduction to Numerical Analysis. 2nd ed. New York: John Wiley & Sons, 712 p.
4. Traub J. F. (1964) Iterative Methods for the Solution of Equations. Englewood Cliffs: Prentice-Hall, 144 p.
5. Kung H. T., Traub J. F. (1974) Optimal order of one-point and multipoint iteration. Journal of the ACM, vol. 21, no. 4, 643–651. DOI: 10.1145/321850.321860
6. Special Issue “Numerical Methods for Nonlinear Equations”. Mathematics (MDPI), 2023. https://www.mdpi.com/journal/mathematics/special_issues/Nonlinear_Equations
7. Yalda Q. (2014) Numerical Solution of Nonlinear Equations in Maple. International Journal of Engineering Research and Applications, vol. 4, no. 6, 100–104. https://doi.org/10.31033/ijrasb.8.4.6
8. Maturi D. A., Malaikah H. M. (2015) The Adomian Decomposition Method for Solving Nonlinear Partial Differential Equation Using Maple. American Journal of Computational and Applied Mathematics, vol. 5, no. 2, 37–43. DOI: 10.4236/apm.2021.116038
9. Abbasbandy S. (2006) A review on the application of Maple in solving nonlinear differential equations. Applied Mathematics and Computation, vol. 176, no. 2, 431–438. DOI: 10.1007/978-3-7091-0517-7
10. Wang P., Zeng X. (2013) Solving Systems of Nonlinear Equations Using Maple. Applied Mechanics and Materials, vol. 427, 52–56. DOI: 10.1007/978-3-319-07671-3_7
11. Брагінець О.В., Воробйова А.І. (2022) Використання математичного пакету Maple до розв’язання СЛАР прямими чисельними методами // Комп'ютерно-інтегровані технології: освіта, наука, виробництво, № 47. С.55-63. DOI: https://doi.org/10.36910/6775-2524-0560-2022-47-08
12. Брагінець О.В. (2025) Використання математичного пакету Maple до розв’язування задач математичної фізики чисельними методами // Комп'ютерно-інтегровані технології: освіта, наука, виробництво, № 59. С.54-60. DOI: https://doi.org/10.36910/6775-2524-0560-2025-59-07
13. Maplesoft. Maple User Manual. Waterloo: Maplesoft, 2023. https://www.maplesoft.com/documentation/
14. Maplesoft. RootFinding Package — Maple Documentation. https://www.maplesoft.com/support/help/Maple/view.aspx?path=RootFinding

References

1.Burden R.L., Faires J.D. (2015) Numerical Analysis. 10th ed. Boston: Cengage Learning, 888 p. DOI: 10.13140/2.1.4830.2406
2.Ralston A., Rabinowitz P. (2001) A First Course in Numerical Analysis. 2nd ed. New York: Dover Publications, 572 p.
3.Atkinson K. E. (1989) An Introduction to Numerical Analysis. 2nd ed. New York: John Wiley & Sons, 712 p.
4.Traub J. F. (1964) Iterative Methods for the Solution of Equations. Englewood Cliffs: Prentice-Hall, 144 p.
5.Kung H. T., Traub J. F. (1974) Optimal order of one-point and multipoint iteration. Journal of the ACM, vol. 21, no. 4, 643–651. DOI: 10.1145/321850.321860
6.Special Issue “Numerical Methods for Nonlinear Equations”. Mathematics (MDPI), 2023. https://www.mdpi.com/journal/mathematics/special_issues/Nonlinear_Equations

7.Yalda Q. (2014) Numerical Solution of Nonlinear Equations in Maple. International Journal of Engineering Research and Applications, vol. 4, no. 6, 100–104. https://doi.org/10.31033/ijrasb.8.4.6
8.Maturi D. A., Malaikah H. M. (2015) The Adomian Decomposition Method for Solving Nonlinear Partial Differential Equation Using Maple. American Journal of Computational and Applied Mathematics, vol. 5, no. 2, 37–43. DOI: 10.4236/apm.2021.116038
9.Abbasbandy S. (2006) A review on the application of Maple in solving nonlinear differential equations. Applied Mathematics and Computation, vol. 176, no. 2, 431–438. DOI: 10.1007/978-3-7091-0517-7
10. Wang P., Zeng X. (2013) Solving Systems of Nonlinear Equations Using Maple. Applied Mechanics and Materials, vol. 427, 52–56. DOI: 10.1007/978-3-319-07671-3_7
11. Brahinets O., Vorobyova A. (2022). Using the mathematical program Maple to solve systems of linear algebraic equations by direct numerical methods. Computer-integrated technologies: education, science, production, (47), 55-63. DOI: https://doi.org/10.36910/6775-2524-0560-2022-47-08
12. Brahinets O. (2025). Using the mathematical program Maple to solve mathematical physics problems by numerical methods. Computer-integrated technologies: education, science, production, (59), 54-60. DOI: https://doi.org/10.36910/6775-2524-0560-2025-59-07
13. Maplesoft. Maple User Manual. Waterloo: Maplesoft, 2023. https://www.maplesoft.com/documentation/
14. Maplesoft. RootFinding Package — Maple Documentation.
https://www.maplesoft.com/support/help/Maple/view.aspx?path=RootFinding

Історія статті:
Отримано: 28.03.2026 Доопрацьовано: 1.04.2026 Прийнято до друку: 23.05.2026 Опубліковано: 29.05.2026