

Akademia Górniczo-Hutnicza im. Stanisława Staszica w Krakowie

Wydział Inżynierii Metali i Informatyki Przemysłowej

# **Sprawozdanie z Laboratorium: Rozwiązywanie Równań Nieliniowych**

Przedmiot: Metody Numeryczne

Kierunek: Inżynieria Obliczeniowa

Autor: Filip Rak

Prowadzący przedmiot: dr hab. inż. Marcin Hojny

Data: 19 kwietnia 2024

Numer lekcji: 7

Grupa laboratoryjna: 4

## Wstęp teoretyczny

### Metoda bisekcji

Jednym ze sposobów na numeryczne rozwiązywanie równań nieliniowych jest metoda równego podziału, zwana również metodą bisekcji. Metoda opiera się na twierdzeniu Darboux:

Jeżeli funkcja  $f(x)$  ma na końcach przedziału domkniętego wartości różnych znaków, to wewnątrz tego przedziału, istnieje co najmniej jeden pierwiastek równania  $f(x) = 0$

*Twierdzenie 1*

Aby można było zastosować metodę bisekcji, muszą być spełnione założenia:

- Funkcja  $f(x)$  jest ciągła w przedziale domkniętym  $[a; b]$
- Funkcja przyjmuje różne znaki na końcach przedziału:  $f(a)f(b) < 0$

Algorytm sprawdza, czy środek przedziału  $[a, b]$  wyrażony jako  $x_1$  jest pierwiastkiem równania poprzez ocenę  $f(x_1) = 0$ . Jeśli  $f(x_1)$  nie równa się zero, przedział jest dzielony na dwa mniejsze przedziały, a następnie, w zależności od znaku wartości funkcji na końcach przedziałów, jedna z granic przedziału  $a$  lub  $b$  jest zastępowana przez  $x_1$ , proces ten jest powtarzany aż do osiągnięcia zadanej dokładności.

Środek przedziału obliczamy w następujący sposób:

$$x_1 = \frac{a + b}{2} \tag{1}$$

Iteracje są kontynuowane do momentu, aż różnica między kolejnymi przybliżeniami będzie mniejsza niż zadana dokładność:

$$|a - b| > \epsilon \tag{2}$$

Gdzie  $\epsilon$  jest przyjętą dokładnością.

## Metoda Newtona-Raphsona

Metoda Newtona-Raphsona, znana również jako metoda stycznych, jest popularną techniką numeryczną służącą do znajdowania przybliżonych rozwiązań równań nieliniowych. Opiera się na wykorzystaniu pierwszej pochodnej funkcji do szybkiego lokalizowania pierwiastków.

Aby metoda Newtona-Raphsona była skuteczna, muszą być spełnione następujące warunki:

- Funkcja  $f(x)$  musi być ciągła i różniczkowalna w rozważanym przedziale.
- Pochodna  $f'(x)$  nie powinna być równa zero w punkcie startowym, aby uniknąć dzielenia przez zero.

Algorytm rozpoczyna się od wybrania początkowego przybliżenia  $x_0$ , które jest punktem startowym dla iteracji.

W każdej iteracji obliczane są wartości  $f(x_0)$  i  $f'(x_0)$  dla aktualnego przybliżenia  $x_0$ .

Nowe przybliżenie  $x_1$  obliczane jest według wzoru:

$$x_1 = x_0 - \frac{f(x_0)}{f'(x_0)} \quad (3)$$

To równanie opiera się na założeniu, że linia styczna do krzywej w punkcie  $x_0$  przecina oś  $X$  w punkcie, który jest lepszym przybliżeniem pierwiastka.

Iteracje są kontynuowane do momentu, aż różnica między kolejnymi przybliżeniami będzie mniejsza niż zadana dokładność  $\epsilon$ , czyli:

$$|x_1 - x_0| < \epsilon \quad (4)$$

Gdzie  $\epsilon$  jest przyjętą dokładnością.

Przybliżoną wartość pochodnej w punkcie  $x$ , jesteśmy w stanie policzyć wzorem różnicowym centralnym

$$f'(x) \approx \frac{f(x+h) - f(x-h)}{2h} \quad (5)$$

Gdzie  $h$  to bardzo mała wartość, zwana krokiem różniczkowania, która pozwala na obliczenie różnicy między wartościami funkcji po obu stronach punktu  $x$ .

## Implementacja

Implementacja metody została przeprowadzona w języku C++ w postaci niezależnych od siebie funkcji `bisection` oraz `newtonRaphson`.

Cechami wspólnymi funkcji są:

- zwracany typ `double`.
- parametr będący wskaźnikiem do zadanej funkcji `double (*f)(double)`.
- parametr będący przyjętą dokładnością rozwiązania `double threshold`.

Definicja obliczanej funkcji matematycznej wygląda następująco i będzie ona zmieniana według potrzeb:

```
double function(double x)
{
    return pow(x, 2) - 9;
}
```

## Pełna definicja funkcji `bisection`

```
double bisection(double (*f)(double), double a, double b, double threshold)
{
    while (abs(a - b) > threshold)
    {
        double x1 = (a + b) / 2;

        if (f(x1) == 0) return x1;

        if (f(a) * f(x1) < 0) b = x1;
        else a = x1;
    }

    return (a + b) / 2;
}
```

## Omówienie elementów funkcji `bisection`

Jako swoje argumenty funkcja przyjmuje wskaźnik do rozwiązywanej funkcji matematycznej `*f` początek przedziału `a`, koniec przedziału `b`, oraz stopień dokładności `threshold`. Zwracanym typem jest liczba zmiennoprzecinkowa podwójnej precyzji.

```
double bisection(double (*f)(double), double a, double b, double threshold)
```

Działanie funkcji jest zamknięte w pętli, której warunkiem jest osiągnięcie odpowiedniej dokładności (2).

```
while (abs(a - b) > threshold)
```

Wewnątrz pętli najpierw obliczymy  $x_1$ , który stanowi środek naszego przedziału. Wykorzystujemy wzór (1).

```
double x1 = (a + b) / 2;
```

Następnie sprawdzamy czy wyliczony środek jest naszym szukanym pierwiastkiem. Wywołujemy funkcję z  $x_1$  będącym jej argumentem. Wynik porównujemy z zerem, jeżeli warunek jest spełniony, udało nam się odnaleźć pierwiastek i go zwracamy.

```
if (f(x1) == 0) return x1;
```

W przeciwnym wypadku nie udało nam się odnaleźć pierwiastka. Z  $x_1$  dzieląc nasz przedział na dwie części, wykorzystując twierdzenie 1 sprawdzamy w którym z tych przedziałów znajduje się pierwiastek. I dostosowujemy granice w taki sposób, aby go przeszukiwać w następnej iteracji.

```
if (f(a) * f(x1) < 0) b = x1;  
else a = x1;
```

Jeżeli działanie pętli `while` zakończy się przed zwróceniem wyniku, oznacza to, że nie udało nam się obliczyć dokładnej wartości pierwiastka. W takim wypadku obliczamy  $x_1$  wzorem (1) i zwracamy jako przybliżenie.

```
return (a + b) / 2;
```

## Pełna definicja funkcji newtonRaphson

```
double newtonRaphson(double (*f)(double), double x0, double threshold)  
{  
    double x1;  
    while (true)  
    {  
        x1 = x0 - f(x0) / numericDerivative(f, x0);  
  
        if (fabs(x1 - x0) < threshold)  
            return x1;  
  
        x0 = x1;  
    }  
}
```

Funkcja `newtonRaphson` wykorzystuje osobną funkcję `numericDerivative` do obliczania wartości pochodnych. Zwraca ona przybliżoną wartość pochodnej funkcji w punkcie  $x$ , wykorzystując wzór (5). Jej definicja jest następująca:

```
double numericDerivative(double (*f)(double), double x, double h = 1e-5)
{
    return (f(x + h) - f(x - h)) / (2 * h);
}
```

## Omówienie elementów funkcji `newtonRaphson`

Funkcja w ramach swoich argumentów przyjmuje wskaźnik `*f` do funkcji dla której szukamy miejsca zerowego. `x0` będące wartością startową używaną jako pierwsze przybliżenie miejsca zerowego. Stopień przybliżenia `threshold`. Zwracaną wartość jest typ zmiennoprzecinkowy podwójnej precyzji.

```
double newtonRaphson(double (*f)(double), double x0, double threshold)
```

Algorytm najpierw deklaruje zmienną `x1`, służącą do przechowywania nowego przybliżenia.

```
double x1;
```

Reszta pracy algorytmu dzieje się wewnątrz pętli `while`.

```
while (true)
```

Najpierw obliczana jest nowa wartość przybliżenia z użyciem wzoru (3)

```
x1 = x0 - f(x0) / numericDerivative(f, x0);
```

Jeżeli przybliżenie tej wartości jest satysfakcjonujące, wynik jest zwracany. Wzór (4).

```
if (fabs(x1 - x0) < threshold)
    return x1;
```

W przypadku przeciwnym, do wartości początkowego przybliżenia `x0`, przypisane jest nowo obliczone przybliżenie `x1` i iteracja jest powtórzona.

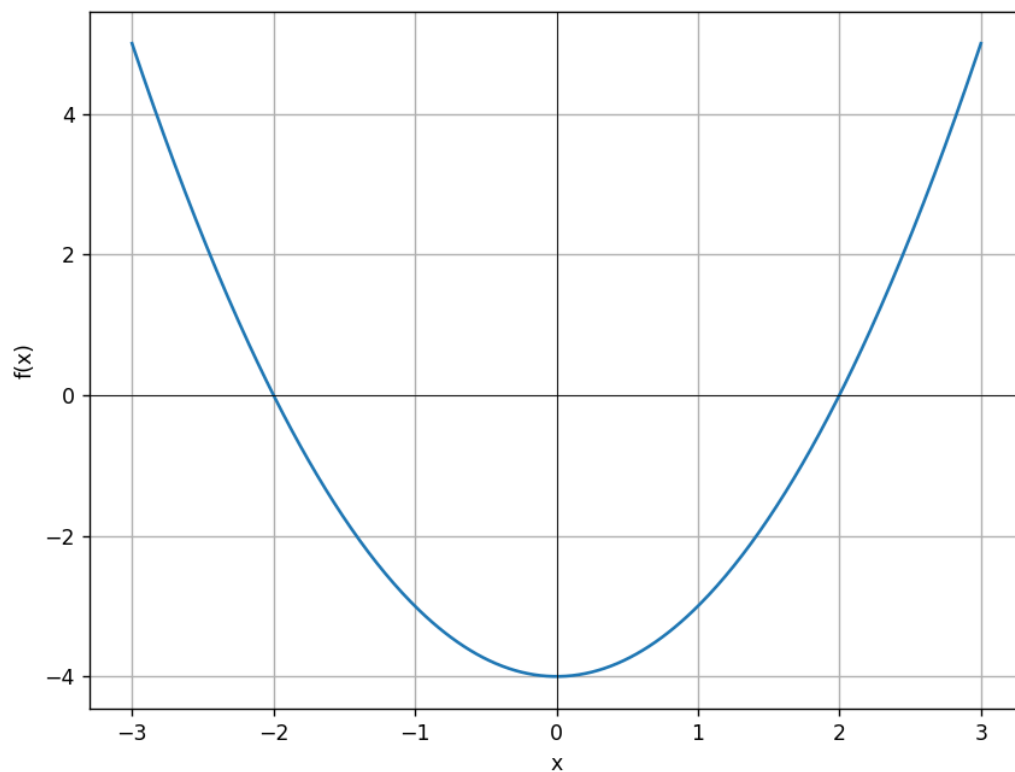
```
x0 = x1;
```

## Testy na wybranych przykładach

Skuteczność zaimplementowanej metody została przetestowana na tle starannie wyselekcjonowanych testów. Zaimplementowane metody zostały przetestowane z różnymi funkcjami i różnymi wartościami przybliżenia.

### Funkcja kwadratowa

$$f(x) = x^2 - 4$$



Wykres 1

**Przedział:**  $[0, 5]$

$X_0 = 2.5$

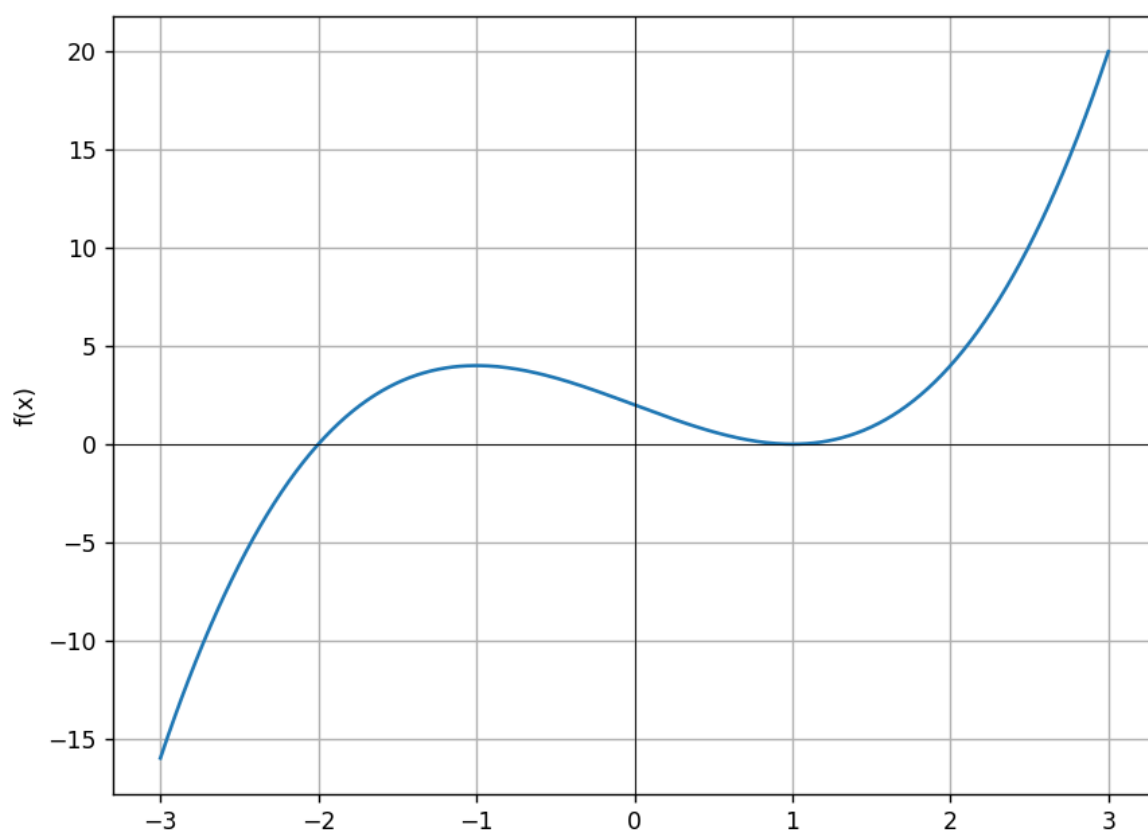
**Oczekiwane pierwiastki:**  $-2, 2$

Wyniki				Funkcja
$\epsilon = 1e-8$	$\epsilon = 1e-6$	$\epsilon = 1e-4$	$\epsilon = 1e-2$	
2	2	2.00001	1.99707	
2	2	2	2	newtonRaphson

Tabela 1

## Funkcja kubiczna

$$f(x) = x^3 - 3x + 2$$



Wykres 2

**Przedział:**  $[-3, 3]$

$X_0 = 0$

**Oczekiwane pierwiastki:**  $1, -2$

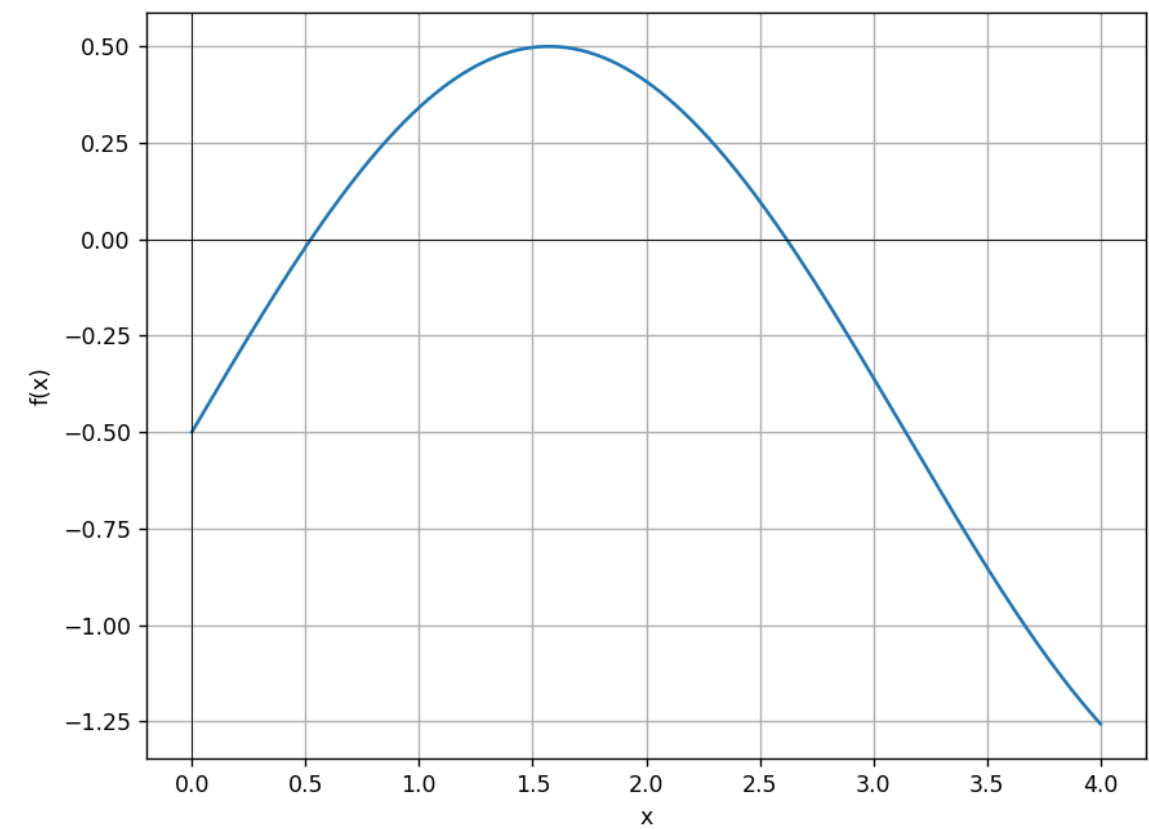
Wyniki				
$\epsilon = 1e-8$	$\epsilon = 1e-6$	$\epsilon = 1e-4$	$\epsilon = 1e-2$	Funkcja
-2	-2	-2.00002	-2.00098	bisection
1	0.999999	0.999928	0.990763	newtonRaphson

Tabela 2



Funkcja trygonometryczna

$f(x) = \sin(x) - 0.5$



Wykres 3

Przedział:[0, 4]

$X_0 = 2$

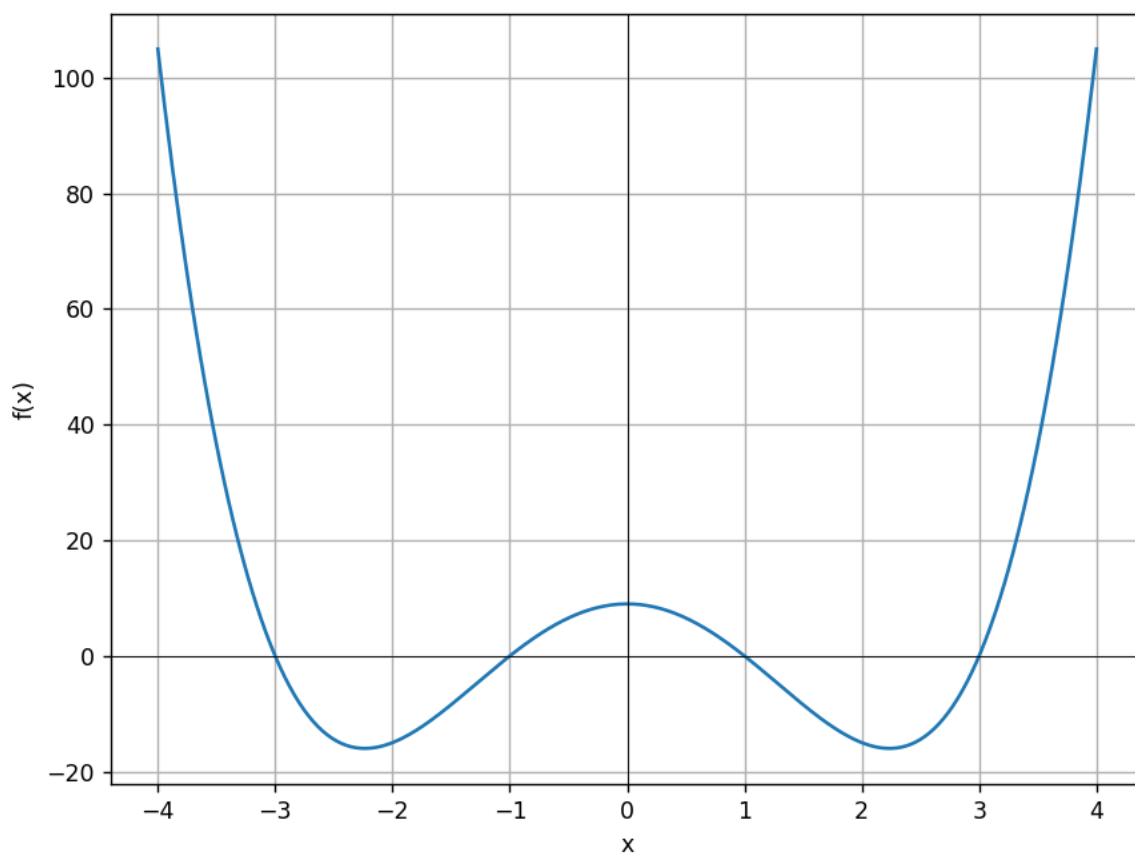
Oczekiwane pierwiastki: 0.5236,3.6652

Wyniki				Funkcja
$\epsilon = 1e-8$	$\epsilon = 1e-6$	$\epsilon = 1e-4$	$\epsilon = 1e-2$	
0.523599	0.523599	0.52359	0.527344	
2.61799	2.61799	2.61799	2.61799	newtonRaphson

Tabela 3

## Funkcja wielomianowa

$$f(x) = x^4 - 10x^2 + 9$$



Wykres 4

**Przedział:**  $[-4, 4]$

$X_0 = 0.5$

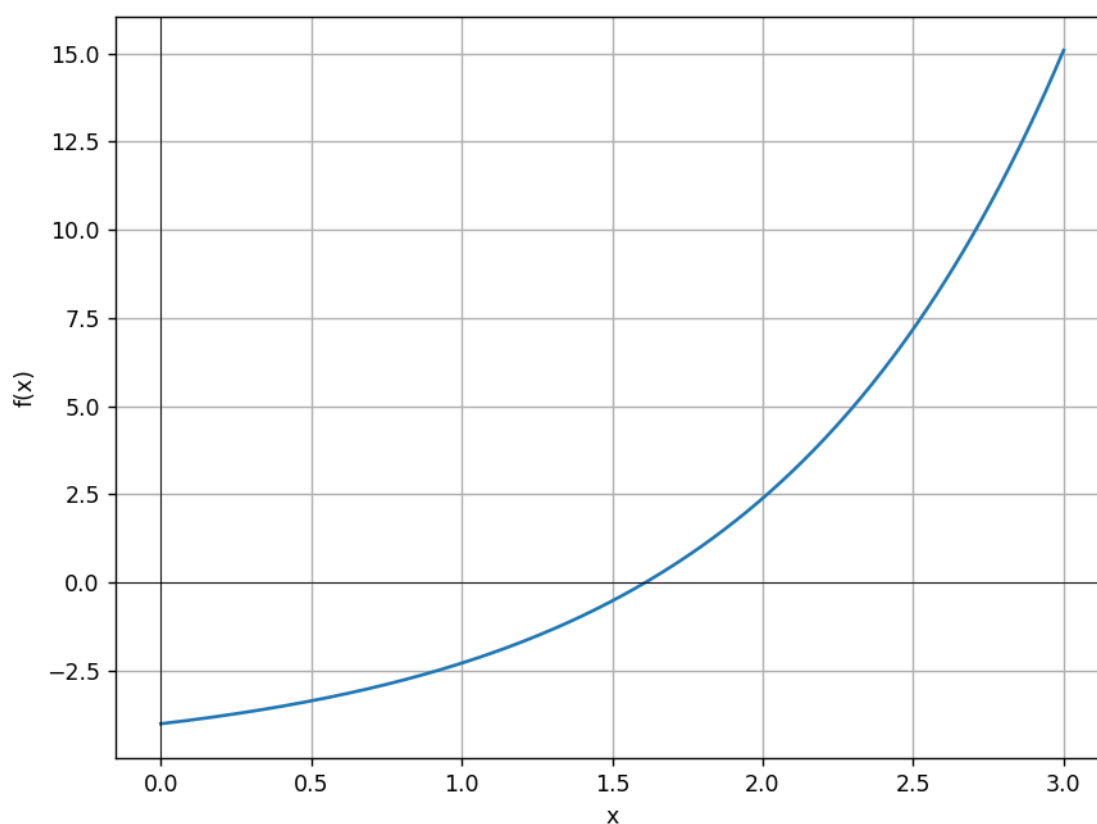
**Oczekiwane pierwiastki:**  $-3, -1, 1, 3$

Wyniki				Funkcja
$\epsilon = 1e-8$	$\epsilon = 1e-6$	$\epsilon = 1e-4$	$\epsilon = 1e-2$	
1	1	1	1	bisection
1	1	1	1.00001	newtonRaphson

Tabela 4

## Funkcja eksponencyjna

$$f(x) = e^x - 5$$



Wykres 5

**Przedział:**  $[0, 3]$

**$X_0 = 2$**

**Oczekiwane pierwiastki:**  $\ln(5) \approx 1.6094$

Wyniki				Funkcja
$\epsilon = 1e-8$	$\epsilon = 1e-6$	$\epsilon = 1e-4$	$\epsilon = 1e-2$	
1.60944	1.60944	1.60945	1.6084	
1.60944	1.60944	1.60944	1.60944	newtonRaphson

Tabela 5

## Opracowanie wyników

**Funkcja Kwadratowa:**  $f(x) = x^2 - 4$

- **Metoda Bisekcji** zbiega do prawidłowego pierwiastka  $x = 2$  dla wszystkich wartości progowych, z niewielkimi odchyleniami przy większym  $\varepsilon$ .
- **Metoda Newtona-Raphsona** szybko zbiega do dokładnego rozwiązania  $x = 2$ , wykazując wysoką efektywność i stabilność w tym przypadku.

**Funkcja Kubiczna:**  $f(x) = x^3 - 3x + 2$

- **Metoda Bisekcji** z powodzeniem znajduje pierwiastek  $x = -2$  niezależnie od  $\varepsilon$ , z niewielkimi odchyleniami dla większych wartości  $\varepsilon$ .
- **Metoda Newtona-Raphsona** wykazuje problem z zbieganiem do różnych pierwiastków w zależności od wartości  $\varepsilon$ , co wskazuje na wpływ punktu startowego oraz lokalnej geometrii funkcji na wynik.

**Funkcja Trygonometryczna:**  $f(x) = \sin(x) - 0.5$

- **Metoda bisekcji:** Daje dokładne wyniki dla pierwiastka w okolicach  $x = 0.5236$  z niewielkimi odchyleniami przy  $\varepsilon = 1e-2$
- **Metoda Newtona-Raphsona:** Wybór  $x_0 = 2$  skutkuje znalezieniem drugiego pierwiastka *około*  $x = 2.61799$ , co jest stabilne dla wszystkich wartości progowych.

**Funkcja Wielomianowa:**  $f(x) = x^4 - 10x^2 + 9$

- Obie metody skutecznie zbiegają do pierwiastka  $x = 1$  dla wszystkich wartości  $\varepsilon$ . Wartości te są dokładne i pokazują, że zarówno bisekcja jak i Newton-Raphson mogą być efektywne dla wielomianowych równań wyższego stopnia z odpowiednimi warunkami startowymi.

**Funkcja Eksponencyjna:**  $f(x) = e^x - 5$

- Obie metody zbiegają do bardzo dokładnego przybliżenia pierwiastka  $\ln(5) \approx 1.6094$ , z nieznacznymi różnicami dla większych  $\varepsilon$  w metodzie bisekcji.

## Podsumowanie

- **Metoda Bisekcji** wykazuje stabilność, ale czasami z mniejszą dokładnością przy większym  $\varepsilon$ . Jest to metoda bezpieczna, ponieważ zawsze zbiega do pierwiastka, jeśli tylko istnieje zmiana znaku na końcach przedziału.
- **Metoda Newtona-Raphsona** jest szybsza i bardziej efektywna, gdy  $x_0$  jest dobrze wybrane, ale może prowadzić do problemów z zbieżnością w przypadku niestandardowych funkcji lub nieodpowiedniego punktu startowego.