<table>
<tr><td>Imię i nazwisko
Meg Paskowski</td><td rowspan="2">Temat laboratorium
Rozwiązywanie równań nieliniowych – metoda bisekcji, metoda Newtona-Raphsona</td><td>Data oddania
21.04.2024r.</td><td>Data ćwiczeń
19.04.2024r.</td></tr>
<tr><td>Prowadzący
dr hab. inż. Marcin Hojny</td><td colspan="2">Grupa laboratoryjna
4</td></tr>
</table>

1. Cel ćwiczenia

Celem laboratorium nr. 7 było zapoznanie się z pojęciami rozwiązań równań nieliniowych – metody bisekcji oraz Newtona-Raphsona oraz implementacja tego zagadania w wybranym przez siebie języku programowania.

2. Wstęp teoretyczny

Metoda bisekcji (metoda płowienia przedziału) – jest to jedna z najprostszych metod znajdowania przybliżonego rozwiązania równań nieliniowych. Działa ona na zasadzie iteracyjnego dzielenia przedziału poszukiwań na pół i wybierania tego podprzedziału, który zawiera rozwiązanie.

Aby można było skorzystać z tej metody muszą zastać spełnione następujące warunki

- Wewnątrz przedziału [a,b] znajduję się dokładnie jeden pierwiastek.
- Funkcja f oraz jej pierwsza i druga pochodna są ciągłe w analizowanym przedziale [a,b]
- Wartości funkcji w punktach a i b mają przeciwny znak,

$$f(a)\cdot f(b) < 0 \tag{1}$$

Ponieważ metoda bisekcji polega na iteracyjnym dzieleniu przedziału [a,b] na pół – znajdowaniu punktu środkowego

$$x_i=\frac{a+b}{2} \tag{2}$$

i sprawdzaniu, w którym z podprzedziałów wartość funkcji zmienia znak. Następnie proces ten powtarzany jest w wybranym podprzedziale, aż osiągniemy wystarczająco mały przedział, w którym można uznać, że znajduje się rozwiązanie.
Wybierany jest ten przedział, dla którego spełnione jest założenie

$$f(x_i)f(a) <0 \text{ albo } f(x_i)f(b) < 0 \tag{3}$$

Metoda Newtona-Raphsona (metoda statycznych) – jest to jedna z najbardziej popularnych technik znajdowania pierwiastków równań nieliniowych. Polega ona na wykorzystaniu stycznych do wykresu funkcji w celu iteracyjnego znajdowania przybliżonego rozwiązania.

Aby można było skorzystać z tej metody muszą zastać spełnione następujące warunki

- Wewnątrz przedziału [a,b] znajduję się dokładnie jeden pierwiastek.
- Funkcja f oraz jej pierwsza i druga pochodna mają stały znak analizowanym przedziale [a,b]
- Wartości funkcji w punktach a i b mają przeciwny znak (1).

Metoda Newtona-Raphsona rozpoczyna się od przybliżonej wartości pierwiastka x_0. Następnie buduje się styczną do wykresu funkcji w punkcie x_0, i znajduje się przecięcie tej stycznej z osią OX. Nowa wartość x_1 to miejsce przecięcia tej stycznej z osią OX. Proces ten jest powtarzany iteracyjnie, z wykorzystaniem kolejnych przybliżeń x_n za pomocą wzoru (4)

$$x_{n+1}=x_n - \frac{f(x_n)}{f'(x_n)} \tag{4}$$

do momentu spełnienia określonego warunku zbieżności.

- Wartość funkcji w wyznaczonym punkcie jest bliska 0

$$|f(x_n)| <= zadana\_dokładność \quad (5)$$

- Odległość pomiędzy kolejnymi przybliżeniami jest wystarczająca mała

$$|x_{n+1}-x_n| <= zadana\_kodładność \quad (6)$$

3. Implementacja

W ramach ćwiczeń zaimplementowano dwie metody rozwiązywania równań nieliniowych – bisekcji, Newtona Raphsona. Na samym początku zadeklarowana została zmienna globalna „*tolerance*” do której przypisana została wartość żądanej dokładności przybliżenia oraz funkcja „*function()*” w której określona jest funkcja.

```
// Zmienna globalna dla tolerancji
const double tolerance = 1e-6;

// Definicja funkcji, dla której szukamy pierwiastka
double function(double x) {
	return x * x - 4; // Przykładowa funkcja
}
```

Pierwsza metoda została zaimplementowana w funkcji „*bisection()*”, która przyjmuje przedział [a,b]. Zadeklarowane są następnie zmienne „*x*” przechowująca wynik oraz „*i*” jako zmienna pomocnicza do przejścia iteracji. W dalszej części metody sprawdzany jest warunek (1), gdy zostanie spełniony w pętli za pomocą wzoru (2) liczony jest następny przedział i sprawdzane czy jest on miejscem zerowym – jeżeli nie to wybierany jest odpowiedni przedział dla którego zostały spełnione warunki (3). Działanie algorytmu kończy się po znalezieniu dokładnego pierwiastka lub po osiągnięciu żądanej dokładności przybliżenia pierwiastka. Pętla również została ograniczona do ilości iteracji dzięki zmiennej „*i*” aby uniknąć nieskończonej ilość i wykonywania pętli - "utknięciu" programu w nieskończonym cyklu. Umożliwia to przerywanie obliczeń w przypadku, gdyby zbieżność nie została osiągnięta w rozsądnym czasie. Na sam koniec wypisywany jest otrzymany wynik.

```
//Metoda bisekcji
void bisection(double start, double end) {
	double x = 0.0;//Zmiana przechowujaca start+end/2
	int i = 0; //Zmienna pomocnicza do iteracji w pętli

	//Sprawdzenie warunku
	if (function(start) * function(end) > 0) {
		cout << "No sign change at the ends of the interval - bisection." << endl;
	}
	else {
		do {
			x = (start + end) / 2;
			if (function(x) == 0.0) {
				break;
			} else if (function(x) * function(start) < 0){
				end = x;
			} else {
				start = x;
			}
			i++;
		} while (fabs(function(x)) > tolerance &&  (i < 1000));
	}
	cout << "Found root - bisection: " << x << endl;
}
```

W metodzie Newtona Raphsona o nazwie „*newtonRaphson()*” przyjmowane jest przewidywane miejsce zerowe „x_0”. Następnie deklarowana jest zmienna pomocnicza do iteracji w funkcji „i”, zmienna „xi” przyjmująca początkowe przybliżenie pierwiastka oraz „f_xn” która przechowuję obliczoną wartość funkcji dla przyjętego przybliżenia. W kolejnym etapie wykonywana jest pętla dopóki wartość bezwzględna funkcji dla xi jest większa od tolerancji oraz liczba iteracji nie przekroczy określonego limitu. Zmienna „f_prime_xi" służy do przechowywania wyniku wartości pierwszej pochodnej funkcji w punkcie xi. Poniżej sprawdzany jest warunek (5), czy pierwsza pochodna jest równa zero, co uniemożliwia kontynuację metody. Później następuje obliczenie następnego przybliżenia x_i+1 (4) oraz sprawdzenie kolejnego warunku - zbieżności (6) (gdy różnica między kolejnymi przybliżeniami jest mniejsza niż przyjęta tolerancja). Następnie aktualizowany jest wartość „xi" dla kolejnej iteracji, obliczana jest wartość funkcji dla nowego przybliżenia i wykonywana jest inkrementacja zmiennej iteracyjnej. Na sam koniec wykonywane jest sprawdzenie, czy wartość bezwzględna funkcji dla xi jest mniejsza lub równa tolerancji i zwracany jest odpowiedni wynik.

```
double newtonRaphson(double x_0) {
	int i = 0; //Zmienna pomocnicza do iteracji w pętli
	double xi = x_0;
	double f_xn = function(x_0);

	while (fabs(f_xn) > tolerance && i < 10000) {
		double f_prime_xi = derivative(xi);

		if (f_prime_xi == 0) {
			return 0;
		}

		double xn_next = xi - f_xn / f_prime_xi;

		if (fabs(xn_next - xi) <= tolerance) {
			return xn_next;
		}
		xi = xn_next;
		f_xn = function(xi);
		i++;
	}

	if (fabs(f_xn) <= tolerance) {
		return xi;
	}
	else {
		return 0;
	}
}
```

Program główny, w którym wywoływane są funkcje metod szukania miejsc zerowych funkcji. Na początku deklarowane są zmienne „*a*" i „*b*" określające przedział [a,b] w metodzie bisekcji. Następnie przedział ten jest określany przez użytkownika i wywoływana jest funkcja „*bisection()*". Poniżej deklarowana jest początkowa wartość przybliżenia pierwiastka oraz uruchamiana funkcja „*newtonRaphson()*" oraz zwracany jest odpowiedni komunikat.

```
	//Metody szukania pierwiastków dla podanej funkcji
	//Metoda bisekcji
	double a, b;   //Deklaracka zmiennych a,b służących do określenia przedziału

	cout << "Enter the interval [a, b]: " << endl;
	cin >> a >> b;

	bistection(a, b);


	//Metoda Newtona-Raphsona
	double x_0 = 1.5;      //Ustalenie początkowego przybliżenia pierwiastka na podstawie
	wykresu funkcji
	double result = newtonRaphson(x_0);

	if (result)
		cout << "Found root - Newton-Raphson: " << result << endl;
	else
		cout << "Root not found - Newton-Raphson." << endl;
```

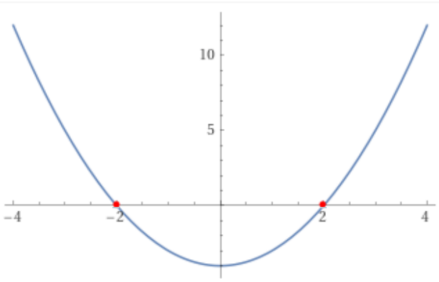
4. Testy jednostkowe

Testy zostały przeprowadzone dla pięciu funkcji oraz porównane z wynikami uzyskanymi na stronie „*WolfranAlpha.com*".

I. Test 1 dla funkcji $f(x) = x^2 - 4$
   - Wynik metody bisekcji dla przedziału [0;5]: 2
   - Wynik metody Newtona-Raphsona dla przewidywanego pierwiastka 1,5: 2
   - Oczekiwany wynik: 2

```
Enter the interval [a, b]:
0
5
Found root - bisection: 2
Found root - Newton-Raphson: 2
Press any key to continue . . .
```

Rysunek 1 Wynik programu dla testu 1

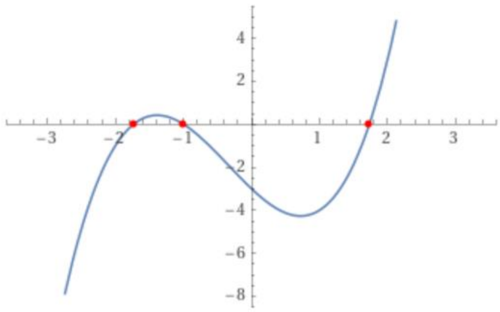


Rysunek 2 Przedstawienie graficzne ze strony "WolfranAlpha.com"

II. Test 2 dla funkcji $f(x) = x^3+x^2-3x-3$
   - Wynik metody bisekcji dla przedziału [0;2]: 1.73205
   - Wynik metody Newtona Raphsona dla przewidywanego pierwiastka 1,5: 1.73205
   - Oczekiwany wynik: $\sqrt{3} \approx 1.73205$

```
Enter the interval [a, b]:
0 2
Found root - bisection: 1.73205
Found root - Newton-Raphson: 1.73205
Press any key to continue . . .
```

Rysunek 3 Wynik programu dla testu 2

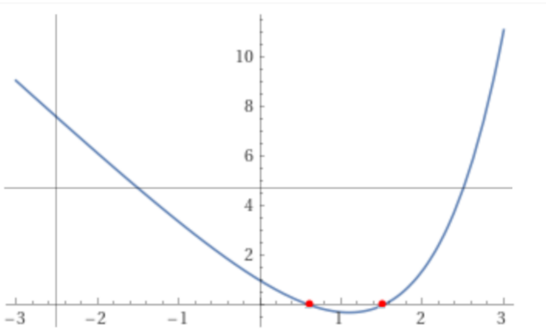


Rysunek 4 Przedstawienie graficzne ze strony „WolframAlpha.com"

III. Test 3 dla funkcji $f(x) = e^x-3x$
   - Wynik metody bisekcji dla przedziału [0; 3]: 1.51213
   - Wynik metody Newtona Raphsona dla przewidywanego pierwiastka 2.0: 1.51213
   - Oczekiwany wynik: 1.51213

```
Enter the interval [a, b]:
1 3
Found root - bisection: 1.51213
Found root - Newton-Raphson: 1.51213
Press any key to continue . . .
```

Rysunek 5 Wynik programu dla testu 3

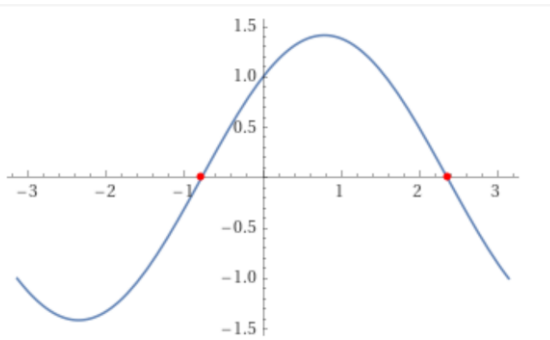


Rysunek 6 Przedstawienie graficzne ze strony „WolframAlpha.com"

IV. Test 4 dla funkcji f(x) = sinx + cosx
- Wynik metody bisekcji dla przedziału [1.5; 3]: 2.35619
- Wynik metody Newtona Raphsona dla przewidywanego pierwiastka 2.0: 2.35619
- Oczekiwany wynik: 2.35619

```
Enter the interval [a, b]:
1.5
3
Found root - bisection: 2.35619
Found root - Newton-Raphson: 2.35619
Press any key to continue . . .
```

Rysunek 7 Wyniki programu dla testu 4

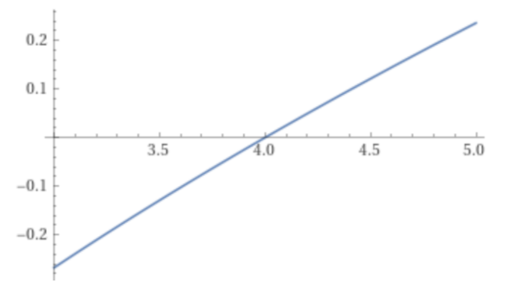


Rysunek 8 Przedstawienie graficzne ze strony „WolframAlpha.com"

V. Test 5 dla funkcji f(x) = $\sqrt{x} - 2$
- Wynik metody bisekcji dla przedziału [3.0; 5.0]: 4
- Wynik metody Newtona Raphsona dla przewidywanego pierwiastka 3.5: 4
- Oczekiwany wynik: 4

```
Enter the interval [a, b]:
3
5
Found root - bisection: 4
Found root - Newton-Raphson: 4
Press any key to continue . . .
```

Rysunek 9 Wyniki programu dla testu 5



Rysunek 10 Przedstawienie graficzne ze strony „WolframAlpha.com"

5. Opracowanie wyników

Zestawienie zawarte w poniższej tabeli 1 przedstawia

Tabela 1. Opracowanie uzyskanych wyników testów

Numer Testu	Metoda bisekcji	Metoda Newtona-Raphsona	Oczekiwany wynik
1	2	2	2
2	1.73205	1.73205	1.73205
3	1.51213	1.51213	1.51213
4	2.35619	2.35619	2.35619
5	4	4	4

Analiza wyników testów wykazała, że wszystkie testy przeprowadzone na funkcjach kwadratowej, sześciennej, wykładniczej, trygonometrycznej i pierwiastkowej zakończyły się sukcesem. Dla funkcji o złożonej charakterystyce lub wąskich przedziałów, metoda Newtona-Raphsona może być bardziej skuteczna. Warto zauważyć, że wybór początkowego przybliżenia dla metody Newtona-Raphsona ma kluczowe znaczenie dla szybkości zbieżności i uniknięcia błędów numerycznych.

6. Wnioski

Metoda bisekcji skutecznie znajduje pierwiastki równań nieliniowych w ustalonym przedziale [a, b]. Należy dokładnie analizować funkcję w celu właściwego dobrania przedziału [a, b], aby uzyskać oczekiwane wyniki. Wadą tej metody jest to, że wymaga większej liczby iteracji do uzyskania odpowiedniej dokładności, szczególnie w przypadku szerokich przedziałów lub funkcji o złożonej charakterystyce. Natomiast metoda Newtona-Raphsona jest wydajniejsza niż metoda bisekcji, ale wymaga początkowego przybliżenia pierwiastka. W przypadku funkcji o gładkiej charakterystyce i dobrego początkowego przybliżenia, metoda Newtona-Raphsona osiąga szybszą zbieżność. Jednakże, w niektórych przypadkach może dojść do niestabilności numerycznej, szczególnie gdy pierwsza pochodna funkcji jest bliska zeru.

7. Źródła

- Prezentacja autorstwa dr hab. inż. Marcina Hojnego „Rozwiązywanie równań nieliniowych – metoda bisekcji, metoda Newtona-Raphsona".
- https://pracownik.kul.pl/files/10382/public/r_nieliniowe3.pdf