Laboratorium **Metod Obliczeniowych** Wydział Elektrotechniki Automatyki i Informatyki Politechnika Świętokrzyska	
Studia: **Stacjonarne I stopnia**	Kierunek: **Informatyka**
Data wykonania: **04.12.2017**	Grupa: **3ID13B**
Imię i nazwisko: **Bartłomiej Osak**	
Numer ćwiczenia: **8**	Temat ćwiczenia: **Układy równań nieliniowych**

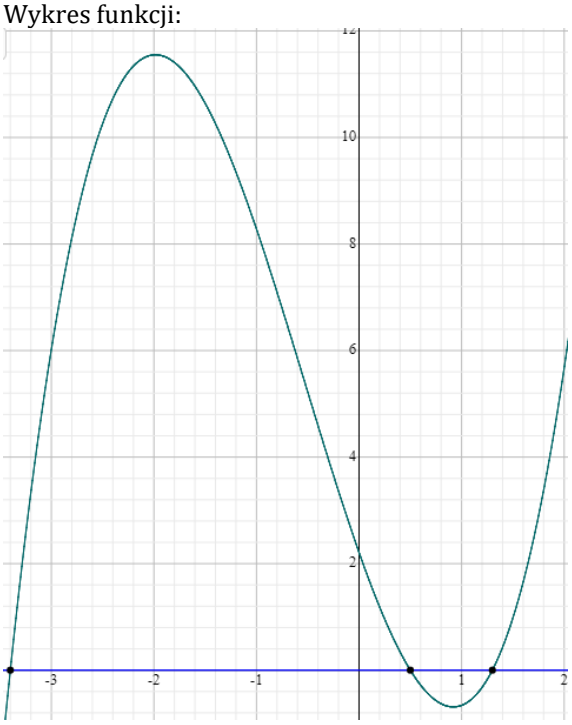
1. Układy równań nieliniowych – metoda cięciw (siecznych) w środowisku MATLAB.

Polecenie: Wykorzystując metodę cięciw rozwiąż równanie: $x^3 + 1.6x^2 - 5.47x = -2.21$.

Wyznaczenie przedziału funkcji:
Zgodnie z wykresem funkcji $x^3 + 1.6x^2 - 5.47x + 2.21 = 0$ miejsca zerowe znajdują się w trzech punktach:

$$x_{0_{(1)}} = -\frac{17}{5}, x_{0_{(2)}} = \frac{1}{2}, x_{0_{(3)}} = \frac{13}{10}$$

Wykres funkcji:



Kod źródłowy:

```
function[c] = metoda_siecznych(a,b,e)
    format long g
    f = inline(input('Podaj rownanie funkcji f(x): ','s'));
    f1 = inline(input('Podaj rownanie pierwszej pochodnej: ','s'));
    f2 = inline(input('Podaj rownanie drugiej pochodnej: ','s'));
    x = (a+b)/2;
    if (f1(x) * f2(x)) < 0
        x(1) = a;
        x(2) = b;
    else
        error('f1(x) * f2(x) musi byc < 0 !');
    end
    for i=3:1000
        x(i) = x(i-1) - (f(x(i-1)))*((x(1) - x(i-1))/(f(x(1)) - f(x(i-1))));
        if abs(f(x(i))) < e
            c = x(i);
            break;
        end
    end
end
```

Wywołanie:

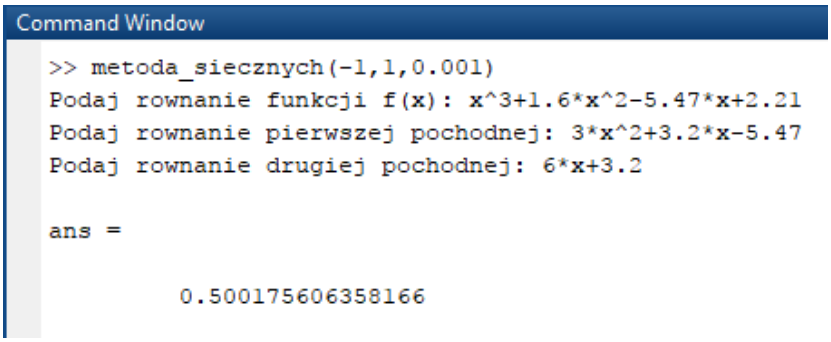
- **Uruchomienie funkcji metoda_siecznych z parametrami: dolna i górna granica przedziału, dokładność:**

```
metoda_siecznych(-1,1,0.001)
```

- **Wynik zadziałania:**

```
Podaj rownanie funkcji f(x): x^3+1.6*x^2-5.47*x+2.21
Podaj rownanie pierwszej pochodnej: 3*x^2+3.2*x-5.47
Podaj rownanie drugiej pochodnej: 6*x+3.2
ans =
        0.500175606358166
```

Zrzut ekranu z działania programu:



Opis programu:

Program metoda_siecznych przyjmuje trzy argumenty wejściowe:

- a – dolna granica przedziału, w którym poszukujemy miejsca zerowego funkcji
- b – górna granica przedziału, w którym poszukujemy miejsca zerowego funkcji
- e – wartość epsilon, czyli wartość dokładności, z jaką będziemy obliczać wystąpienie miejsca zerowego

Ponadto zwraca on wynik zapisany w zmiennej c, czyli miejsce zerowe funkcji zadanej. Na początku do zmiennych f,f1 oraz f2 zapisywane są kolejno wprowadzane przez użytkownika: funkcja, pierwsza pochodna funkcji oraz druga pochodna funkcji. Następnie obliczany jest pkt x, który równy jest sumie dolnej i górnej granicy przedziału podzielonej przez 2. Następnie obliczany jest iloczyn wartości pierwszej i drugiej pochodnej w uprzednio wyznaczonym punkcie x. Jeśli iloczyn będzie ujemny następuje przypisanie do pierwszego indeksu wektora x wartości a, czyli dolnej granicy przedziału, a do kolejnego indeksu przypisanie wartości b, czyli górnej granicy przedziału. W przeciwnym wypadku zostaje wypisany dla użytkownika stosowny komunikat i program kończy działanie (należy dobrać poprawne wartości a oraz b). Całość opiera się na zapisie matematycznym:

$$z = \frac{a+b}{2}$$

$$F'(z) * F''(z) < 0 \rightarrow x_k = a = 1, x_1 = b = 2$$

Następnie inicjowana jest pętla, która iteruje od kolejnego niezajętego indeksu w wektorze x, czyli od wartości 3. W pętli następuje zastosowanie ogólnego wzoru metody siecznych – cięciw:

$$x_i = x_{i-1} - F(x_{i-1}) * \frac{x_k - x_1}{F(x_k) - F(x_1)}$$

Iteracja trwa do momentu określonego nierównością:

$$|F(x)| > \varepsilon$$

Po spełnieniu powyższej pętli została zastosowana instrukcja break, pętla jest przerywana, program kończy działanie i wypisuje oczekiwany wynik.

2. Układy równań nieliniowych – metoda cięciw (siecznych) w języku Java.

Polecenie: Wykorzystując metodę cięciw rozwiąż równanie: $x^3 + 1.6x^2 - 5.47x = -2.21$.

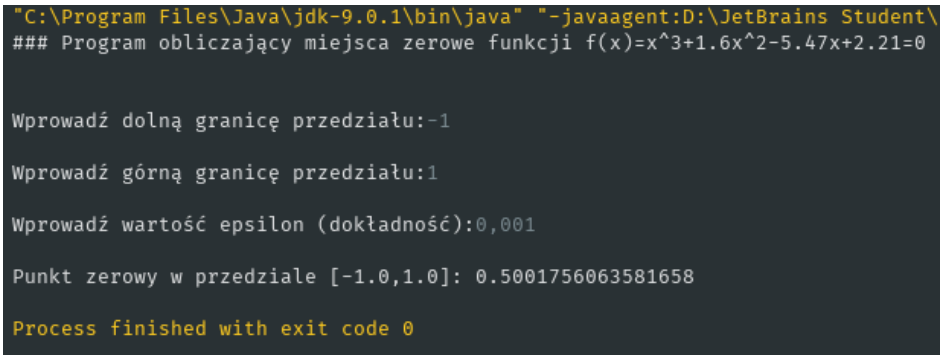
Wyznaczenie punktów zerowych – sposób opisany w pkt. 1 sprawozdania.

Kod źródłowy:

```
import java.util.Scanner;

public class SecantMethod {

    private static double[] x = new double[1000];
    private static double a;
    private static double b;
    private static double e;

    public static double baseFunction(double x) {
        return Math.pow(x, 3) + 1.6 * Math.pow(x, 2) - 5.47 * x + 2.21;
    }

    public static double firstDerivative(double x) {
        return 3 * Math.pow(x, 2) + 3.2 * x - 5.47;
    }

    public static double secondDerivative(double x) {
        return 6 * x + 3.2;
    }

    public static void enterData() {
        boolean end = false;
        while (!end) {
            try {
                System.out.println("### Program obliczający miejsca zerowe funkcji
                f(x)=x^3+1.6x^2-5.47x+2.21=0");
                System.out.println("\n");
                System.out.print("Wprowadź dolną granicę przedziału:");
                a = new Scanner(System.in).nextDouble();
                System.out.print("\nWprowadź górną granicę przedziału:");
                b = new Scanner(System.in).nextDouble();
                System.out.print("\nWprowadź wartość epsilon (dokładność):");
                e = new Scanner(System.in).nextDouble();
                end = true;
            } catch (Exception e) {
                System.out.println("\nWystapił błąd wpisywania danych!");
                for (int i = 0; i < 500; i++) {
                    System.out.println();
                }
            }
        }
    }
    public static void secantAlgorithm() {
        double z = (a + b) / 2;
        double f1 = firstDerivative(z);
        double f2 = secondDerivative(z);
        if (f1 * f2 < 0) {
            x[0] = a;
            x[1] = b;
        } else {
            System.out.println("\nIloczyn wartości pierwszej oraz drugiej pochodnej w przedziale
            [a,b] musi być ujemny!\n");
            enterData();
        }
```

```
        for (int i = 2; i < 1000; i++) {
              x[i] = x[i - 1] - baseFunction(x[i - 1]) * ((x[0] - x[i - 1]) / (baseFunction(x[0])
              - baseFunction(x[i - 1])));
            if (Math.abs(baseFunction(x[i])) < e) {
                double result = x[i];
               System.out.println("\nPunkt zerowy w przedziale [" + a + "," + b + "]: " + result);
                break;
            }
        }
    }

    public static void main(String[] args) {
        SecantMethod.enterData();
        SecantMethod.secantAlgorithm();
    }
}
```

Zrzut ekranu z działania programu:



Opis programu:

Program składa się z jednej klasy publicznej o nazwie SecantMethod. Klasa ta posiada 4 pola statyczne. Są to:

- double[ ] x – tablica przechowująca górną i dolną granicę przedziału oraz wartości powstałe w działaniu algorytmu.
- double a – zmienna przechowująca dolną granicę przedziału wprowadzoną przez użytkownika.
- double b – zmienna przechowująca górną granicę przedziału wprowadzoną przez użytkownika.
- double e – zmienna przechowująca wartość epsilon wprowadzoną przez użytkownika.

Ponadto w programie znajduje się 5 metod statycznych nie wliczając metody inicjalizacyjnej main. Są to:

- basefunction – funkcja przyjmująca parametr i zwracająca wartość funkcji dla tego argumentu.
- firstDerivative – funkcja przyjmująca parametr i zwracająca wartość pierwszej pochodnej funkcji dla tego argumentu.
- secondDerivative– funkcja przyjmująca parametr i zwracająca wartość drugiej pochodnej funkcji dla tego argumentu.
- enterData – metoda odpowiedzialna za wprowadzanie do pól statycznych a,b,e wartości przez użytkownika. W przypadku wystąpienia wyjątku – błędne wpisanie – użytkownik zostanie o tym poinformowany i będzie mógł ponownie wpisać wymagane dane.
- secantAlgorithm – główna metoda odpowiedzialna za obliczenie miejsca zerowego funkcji w podanym przedziale. Cały algorytm opiera się na zasadzie cięciw, która została opisana w pkt 1. sprawozdania w opisie programu w środowisku MATLAB.

Po poprawnym wpisaniu danych program zwraca natychmiastowo oczekiwany wynik.

3. Metoda Newtona (stycznych).

Wstęp ogólny:

Metoda Newtona jest połączeniem idei iteracji i lokalnej aproksymacji liniowej. W tej metodzie iteracyjnej x_{n+1} jest określone jako odcięta punktu przecięcia z osią x stycznej do krzywej y=f(x) w punkcie $(x_n,f(x_n))$.

Przybliżanie krzywej y=f(x) styczną do niej w punkcie $(x_0,f(x_0))$ jest równoważne zastąpieniu funkcji dwoma początkowymi składnikami jej szeregu Taylora w punkcie $x=x_0$. Odpowiednie przybliżanie dla funkcji wielu zmiennych ma również ważne zastosowania.

Twierdzenie:

Niech f będzie funkcją posiadającą pochodne rzędu pierwszego i rzędu drugiego w przedziale [a,b]. Wówczas, jeśli:

- $f(a) * f(b) < 0$
- funkcja f' ma stały znak na $[a, b]$
- funkcja f'' ma stały znak na $[a, b]$

to równanie $f(x) = 0$ ma w przedziale $[a, b]$ dokładnie jedno rozwiązanie c i ponadto dla dowolnego punktu $x_0 \in [a, b]$ takiego, że $f(x_0) * f''(x_0) > 0$ wszystkie wyrazy ciągu x_n określonego wzorem rekurencyjnym:

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

należą do przedziału $[a, b]$ i ciąg ten jest zbieżny do wartości c.

Ponadto, jeśli istnieją stałe $m, M \in R$, takie, że:

$$|f'(x)| \geq m > 0 \; dla \; każdego \; x \in [a, b]$$
$$|f''(x)| \leq M \; dla \; każdego \; x \in [a, b]$$

to otrzymujemy:

$$|c - x_n| \leq \frac{M}{2m} |x_n - x_{n-1}|^2$$

oraz:

$$|c - x_n| \leq \frac{|f(x_n)}{m}$$

Zapis na wykresie: