

Wydział WIMiIP	Imię i nazwisko Mateusz Witkowski	Rok II	Grupa 4
Temat: Optymalizacja – metoda złotego podziału.			Prowadzący dr hab. inż. Hojny Marcin, prof. AGH
Data ćwiczenia 07.05.2020	Data oddania 13.05.2020	Data zaliczenia	OCENA

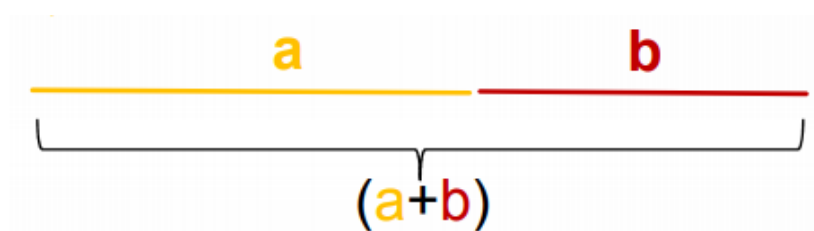
1. Cel ćwiczenia

Celem ćwiczenia było zapoznanie się oraz implementacja metody optymalizacji na podstawie metody złotego podziału.

2. Wprowadzenie teoretyczne

Złoty podział (złota proporcja, boska proporcja, podział harmoniczny) – jest to sposób dzielenia odcinka na dwie części, w wyniku którego stosunek dłuższej z nich do krótszej jest dokładnie taki sam, jak stosunek całego odcinka do dłuższej części. Podział wyraża następujący wzór:

$$\frac{a+b}{a} = \frac{a}{b} \cong \varphi$$



Rysunek 1. Złoty podział odcinka

Złota liczba – φ , przyjmuje następującą wartość, wynikającą ze wzoru:

$$\varphi = \frac{1 + \sqrt{5}}{2} = 1,61803398$$

Tym samym terminem określa się liczbę odwrotną, czyli stosunek krótszej długości do dłuższej, zastosowaną w tym ćwiczeniu:

$$\frac{1}{\varphi} = \frac{2}{1 + \sqrt{5}} = \frac{\sqrt{5} - 1}{2} = \varphi - 1 = 0,61803398$$

Złoty podział znany był już od starożytności gdzie przypisywano mu wyjątkowe walory estetyczne, wykorzystuje się go w architekturze, malarstwie, a nawet fotografii. Matematycy od dawnych czasów badali jego wyjątkowe właściwości. W naszym przypadku posłuży jako metoda odszukania minimum w zadanym przedziale, jednak by algorytm mógł zadziałać muszą zostać spełnione odpowiednie warunki:

- Funkcja jest ciągła w przedziale $[a, b]$.
- Funkcja posiada co najwyżej jedno ekstremum w przedziale $[a, b]$.

Celem algorytmu jest znalezienie najmniejszego możliwego przedziału w zadanej dokładności, w którym ma znajdować się minimum. W tym celu obliczamy dwa dodatkowe punkty znajdujące się w naszym przedziale, spełniające warunek $a < x_l < x_p < b$ przy pomocy wzorów:

$$x_l = b - \varphi(b - a)$$

$$x_p = a + \varphi(b - a)$$

x_l – punkt znajdujący się bliżej lewej granicy przedziału.

x_p – punkt znajdujący się bliżej prawej granicy przedziału.

Wykorzystując wyliczone w ten sposób punkty sprawdzamy, w którym z przedziałów znalazło się nasze minimum:

- Jeżeli $f(x_l) > f(x_p)$ to szukane minimum znajduje się w przedziale $[x_l, b]$
- Jeżeli $f(x_l) < f(x_p)$ to szukane minimum znajduje się w przedziale $[a, x_p]$

Następnie zawężamy przedział, obieramy nowe a lub b , obliczymy kolejne x_l i x_p . Proces powtarzamy aż do spełnienia warunku, w którym odległość między granicami nowo utworzonego przedziału jest mniejsza od zadanej dokładności ($(b - a) < \varepsilon$).

3. Kod programu

Zdefiniowano globalne zmienne odpowiedzialne za dokładność przybliżenia wyniku, wartość złotej liczby i ilość wyświetlanych miejsc po przecinku oraz funkcję pomocniczą mającą na celu obliczanie wartości zadanej funkcji w przekazanym punkcie.

```
using namespace std;
double Dokladnosc = 0.00000001; //Dokladnosc obliczen
double phi = 0.61803398; //Wartosc odwrotnosci złotej liczby
int prec = 11; //Liczba wyswietlanych miejsc po przecinku

double f(double x) { //Pomocnicza funkcja obliczajaca wartosc w punkcie x
    return x*x - 2 * x + 3;
}
```

Rysunek 2. Globalne zmienne, funkcje pomocnicze.

W funkcji main zadeklarowano odpowiednie zmienne i pobrano wartości granic przedziałów oraz wywołano funkcję odpowiedzialną za znalezienie minimum. Wyświetlono wynik.

```
int main() {

    double a, b; //deklaracja zmiennych
    cout << "Obliczanie minimum" << endl;
    cout << endl;
    cout << "-----" << endl;
    cout << "Metoda zlotego podzialu" << endl;
    cout << "Podaj przedzial poszukiwania minimum: " << endl;
    cout << "a = "; cin >> a; //Pobranie granic przedzialu
    cout << "b = "; cin >> b;
    cout << "Wynik: " << setprecision(prec) << zlotyPodzial(a, b, phi, Dokladnosc) << endl;
    cout << endl; //Wyswitlenie wyniku

    getchar(); getchar();
    return 0;
}
```

Rysunek 3. Funkcja main

Definicję algorytmu złotego podziału rozpoczęto od zapisu granic w zmiennych lokalnych. Rozpoczęto pętlę while kończącą się w przypadku gdy odległość między granicami będzie mniejsza od zadanej dokładności. W pętli, w każdym obiegu obliczamy nowe punkty pomocnicze x_l i x_p , na podstawie których sprawdzamy warunki w jakim przedziale mieści się nasze minimum. Po wyznaczeniu właściwej części zawężamy nasz przedział i powtarzamy wszystkie czynności aż do osiągnięcia przekazanej dokładności. Następnie wyświetlamy lewą i prawą granicę oraz zwracamy punkt pomiędzy nimi jako rozwiązanie. Funkcja przyjmuje jako argumenty granice przedziału, wartość odwrotności złotej liczby oraz wybraną dokładność obliczeń.

```
double zlotyPodzial(double a, double b, double Phi, double precision) {
    double l_granica = a;
    double p_granica = b;

    while (fabs(p_granica - l_granica) > precision)
    {
        double Xl = p_granica - Phi*(p_granica - l_granica);    //obliczanie punktow pomocniczych
        double Xp = l_granica + Phi*(p_granica - l_granica);    //w kazdym obiegu petli

        if (f(Xl) > f(Xp)) {    //jesli warunek spelniony - minimum w przedziale [Xl;p_granica]
            l_granica = Xl;    //zawezenie przedzialu
        }
        else{    //jesli warunek nie spelniony - minimum w przedziale [l_granica;Xp]
            p_granica = Xp;    //zawezenie przedzialu
        }
    }

    cout << "Lewa granica: " << setprecision(prec) << l_granica << endl;
    cout << "Prawa granica: " << setprecision(prec) << p_granica << endl;
    return (p_granica + l_granica) / 2; //wybieramy punkt dokladnie posrodku otrzymanych granic
}
```

Rysunek 4. Funkcja realizująca algorytm złotego podziału w celu znalezienia minimum.

Cały kod:

```
using namespace std;
double Dokladnosc = 0.00000001; //Dokladnosc obliczen
double phi = 0.61803398; //Wartosc odwrotnosci zlotej liczby
int prec = 11; //Liczba wyswietlanych miejsc po przecinku

double f(double x) { //Pomocnicza fukncja obliczajaca wartosc w punkcie x
    return x*x - 2 * x + 3;
}

double zlotyPodzial(double a, double b, double Phi, double precision) {
    double l_granica = a;
    double p_granica = b;

    while (fabs(p_granica - l_granica) > precision)
    {
        double Xl = p_granica - Phi*(p_granica - l_granica); //obliczanie punktow pomocniczych
        double Xp = l_granica + Phi*(p_granica - l_granica); //w kazdym obiegu petli

        if (f(Xl) > f(Xp)) { //jesli warunek spelniony - minimum w przedziale [Xl;p_granica]
            l_granica = Xl; //zawezenie przedzialu
        }
        else{ //jesli warunek nie spelniony - minimum w przedziale [l_granica;Xp]
            p_granica = Xp; //zawezenie przedzialu
        }
    }

    cout << "Lewa granica: " << setprecision(prec) << l_granica << endl;
    cout << "Prawa granica: " << setprecision(prec) << p_granica << endl;
    return (p_granica + l_granica) / 2; //wybieramy punkt dokladnie posrodku otrzymanych granic
}

int main() {

    double a, b; //deklaracja zmiennych
    cout << "Obliczanie minimum" << endl;
    cout << endl;
    cout << "-----" << endl;
    cout << "Metoda zlotego podzialu" << endl;
    cout << "Podaj przedzial poszukiwania minimum: " << endl;
    cout << "a = "; cin >> a; //Pobranie granic przedzialu
    cout << "b = "; cin >> b;
    cout << "Wynik: " << setprecision(prec) << zlotyPodzial(a, b, phi, Dokladnosc) << endl;
    cout << endl; //Wyswitlenie wyniku

    getchar(); getchar();
    return 0;
}
```

Rysunek 5. Cały kod programu.

4. Testy

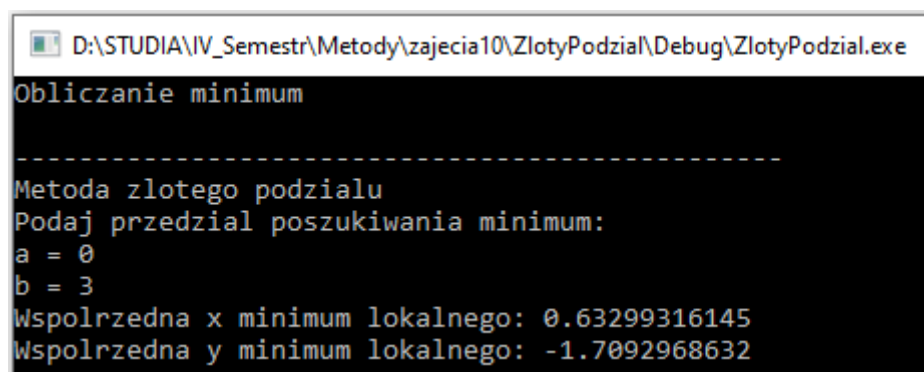
W celu zweryfikowania wyników programu dokonano trzech testów z wykorzystaniem kalkulatora znajdującego się na stronie <https://www.wolframalpha.com/>. W każdym przypadku obliczono błąd względny. Wszystkie obliczenia w programie wykonywano z dokładnością do 0.00000001.

Test 1.

Tabela 1. Dane dla testu 1

Wzór funkcji	Lewa granica przedziału	Prawa granica przedziału
$x^3 + 3x^2 - 5x$	0	3

Wynik programu:



```
D:\STUDIA\IV_Semestr\Metody\zajecia10\ZlotyPodzial\Debug\ZlotyPodzial.exe
Obliczanie minimum
-----
Metoda zlotego podzialu
Podaj przedzial poszukiwania minimum:
a = 0
b = 3
Wspolrzedna x minimum lokalnego: 0.63299316145
Wspolrzedna y minimum lokalnego: -1.7092968632
```

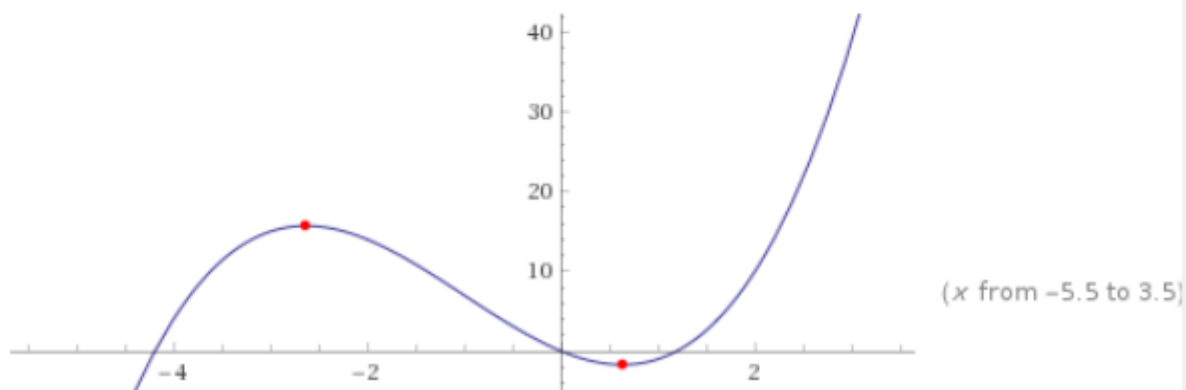
Rysunek 6. Wynik działania programu.

Wynik ze strony:

$$\min\{x^3 + 3x^2 - 5x\} = 7 - \frac{32\sqrt{\frac{2}{3}}}{3} \text{ at } x = 2\sqrt{\frac{2}{3}} - 1$$

[Need a step by step solution for this problem? >>](#)

Plot:



Rysunek 7. Wynik ze strony.

Wartość minimum lokalnego wyniosła 0.6329931619

Błąd względny:

$$\delta = \left| \frac{0.6329931619 - 0.63299316145}{0.6329931619} \right| \cdot 100\% = 7.10908 \cdot 10^{-8} \%$$

Test 2.

Tabela 2. Dane dla testu 2

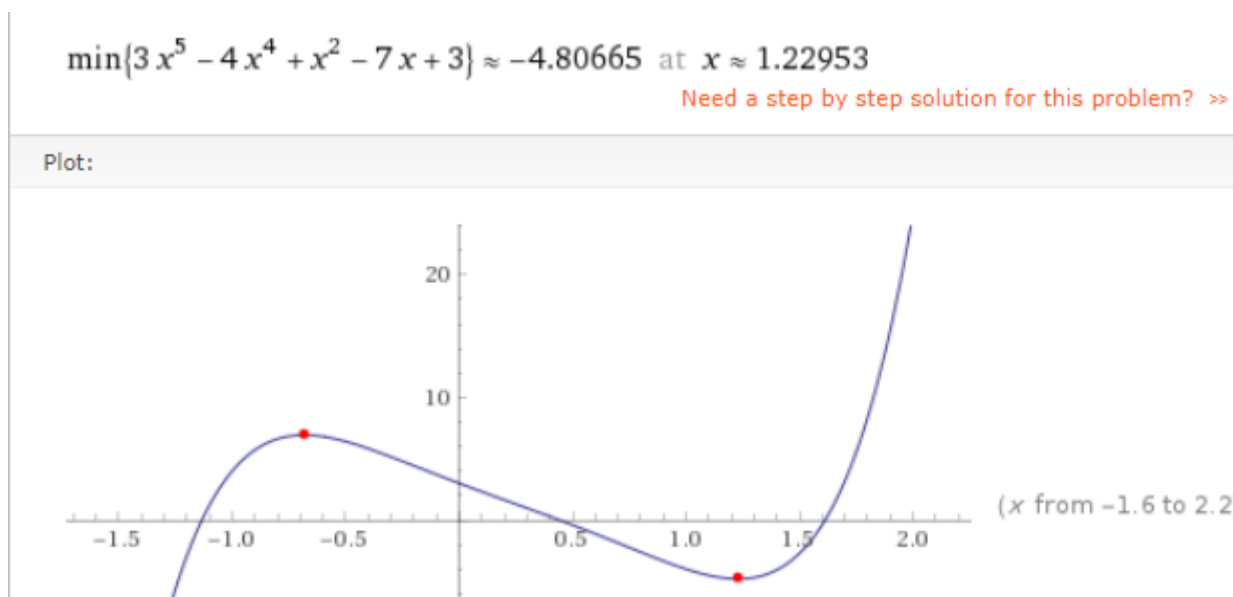
Wzór funkcji	Lewa granica przedziału	Prawa granica przedziału
$3x^5 - 4x^4 + x^2 - 7x + 3$	0	5

Wynik programu:

```
D:\STUDIA\IV_Semestr\Metody\zajecia10\ZlotyPodzial\Debug\ZlotyPodzial.exe
Obliczanie minimum
-----
Metoda zlotego podzialu
Podaj przedzial poszukiwania minimum:
a = 0
b = 5
Prawa granica: 1.2295335255
Wspolrzedna x minimum lokalnego: 1.2295335213
Prawa granica: 1.2295335255
Wspolrzedna y minimum lokalnego: -4.8066530463
```

Rysunek 8. Wynik działania programu.

Wynik ze strony:



Rysunek 9. Wynik ze strony.

Błąd względny:

$$\delta = \left| \frac{1.22953 - 1.22953}{1.22953} \right| \cdot 100\% = 0\%$$

Test 3.

Tabela 3. Dane dla testu 3

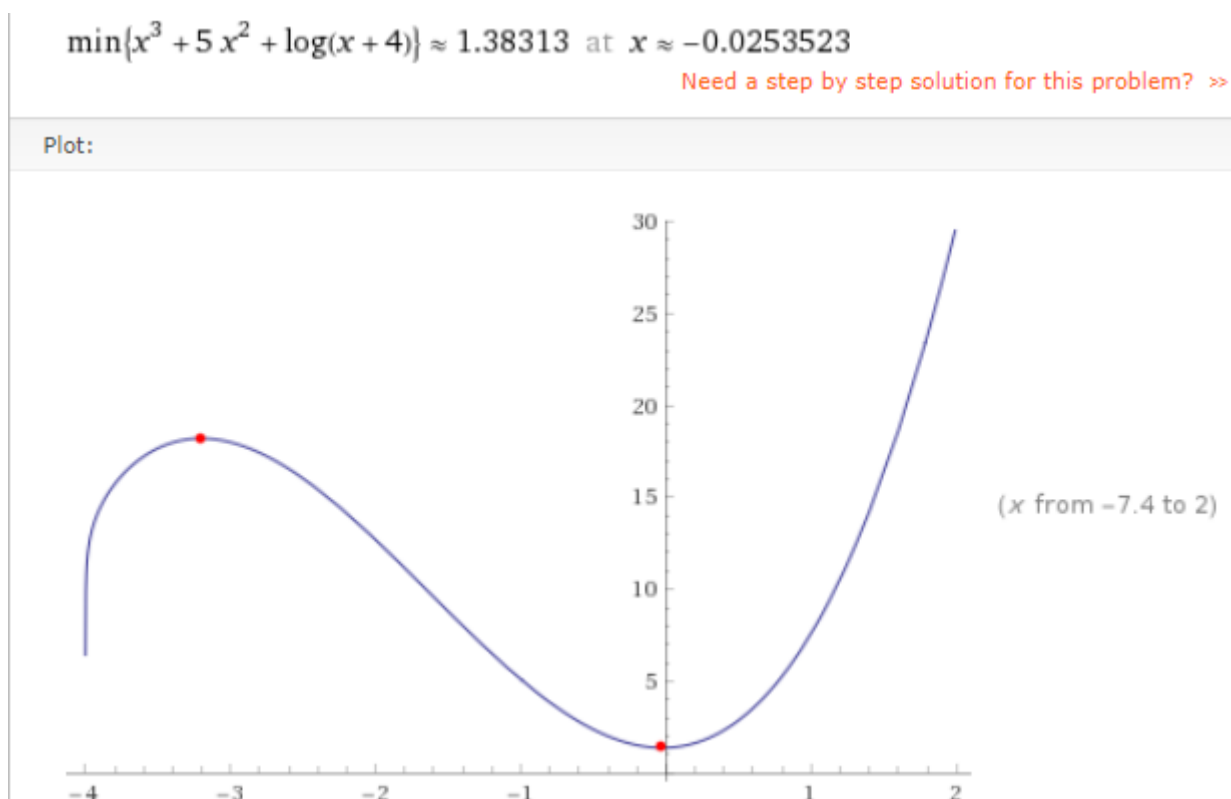
Wzór funkcji	Lewa granica przedziału	Prawa granica przedziału
$x^3 + 5x^2 + \ln(x + 4)$	-3	2

Wynik programu:

```
D:\STUDIA\IV_Semestr\Metody\zajecia10\ZlotyPodzial\Debug\ZlotyPodzial.exe
Obliczanie minimum
-----
Metoda zlotego podzialu
Podaj przedzial poszukiwania minimum:
a = -3
b = 2
Prawa granica: -0.025352280498
Wspolrzedna x minimum lokalnego: -0.025352284671
Prawa granica: -0.025352280498
Wspolrzedna y minimum lokalnego: 1.3831335159
```

Rysunek 10. Wynik działania programu.

Wynik ze strony:



Rysunek 11. Wynik ze strony.

Błąd względny:

$$\delta = \left| \frac{0.0253523 - 0.025352284671}{0.0253523} \right| \cdot 100\% = 6.04639 \cdot 10^{-5} \%$$

5. Wnioski

Wykorzystana w ćwiczeniu metoda złotego podziału jest niezwykle popularny i użyteczną metodą. Znajduje zastosowanie w wielu dziedzinach takich jak architektura, a także w przypadku niektórych matematycznych obliczeń. Jak udowodniły testy metoda ta sprawdza się również w przypadku algorytmu poszukiwania minimum w z zadany przedziale. Wyniki otrzymane tym sposobem, mimo iż zależą od obranej dokładności i zastosowanych przybliżeń są niemalże bezbłędne. Bardzo istotną rzeczą jest dobranie odpowiednie przedziału, tak by znajdowało się tam co najwyżej jedno minimum gdyż w przeciwnym wypadku możemy otrzymać błędne wyniki. Wyliczone błędy względne pokazały jak bardzo precyzyjna jest ta metoda.