|  |  |  |
| --- | --- | --- |
| Katedra Automatyki i Robotyki  **Metody optymalizacji**  Ćwiczenie nr 3 – Metoda złotego podziału | | |
| L.p. | **Imię i nazwisko** | **Data i godzina** |
| 1. | Jakub Szczypek | 04.11.22r.  godz. 15:30 – 17:00 (piątek) |

1. **Charakterystyka metody złotego podziału**

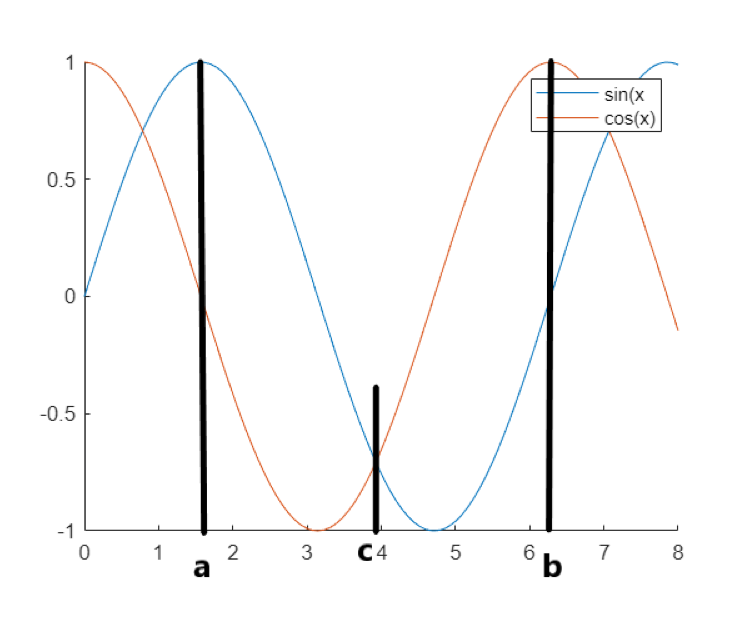
**Metoda złotego podziału** jest to przybliżona metoda poszukiwania lokalizacji minimum funkcji jednej zmiennej na zadanym przedziale. Jest to metoda należąca do grupy metod bezgradientowych. Warunkiem koniecznym stosowania tej metody jest unimodalność badanej funkcji na zadanym przedziale – funkcja posiada dokładnie jedno minimum w przedziale [a, b] i jest ciągła. Lokalizacja minimum pochodzi z arytmetycznie uśrednionych granic przedziału wyjściowego:

dla którego spełniony jest warunek:

Gdzie:

– przyjęta dokładność poszukiwań, przeważnie wynosząca poniżej 0,1.

Aby dokonać redukcji przedziału nieokreśloności, musimy znać wartość funkcji w co najmniej dwóch jego punktach wewnętrznych. Znając wartość funkcji w tylko jednym punkcie wewnętrznym, nie można określić, po której stronie znajduje się poszukiwane minimum. Poniżej przedstawiam przykład dlaczego wymagamy dwóch punktów wewnętrznych.

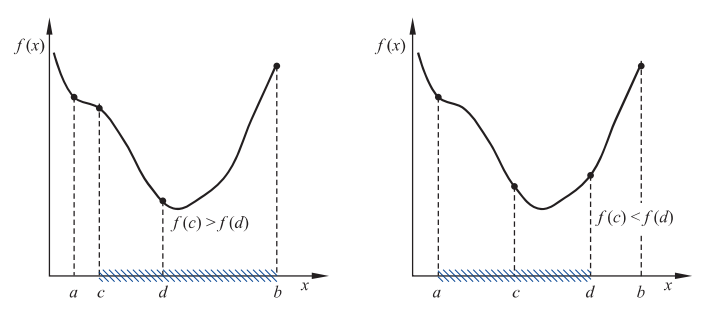


*Rysunek 1. Funkcja sin(x) i cos(x)*

Po pierwsze wybieramy sobie punkt C. Teraz możemy zauważyć, że dla funkcji cosinus minimum funkcji znajduje się w przedziale [a, c] i jest to cos(π), zaś dla funkcji sinus w przedziale [c, b] i jest to sin(). Istnieje w takim razie możliwość, że utracimy unimodalność funkcji na przedziale w kolejnej iteracji metody.

1. **Interpretacja dwóch punktów wewnętrznych**

Jeżeli nasza funkcja f(x) jest unimodalna w przedziale [a, b], to do określenia podprzedziału, w którym leży minimum, wystarczy obliczyć wartości w dwóch punktach wewnętrznych c, d (c < d) tego przedziału. Jeżeli f(c) > f(d), to minimum funkcji znajduje się na prawo od punktu d i przedział poszukiwań zawężamy do [c, b]. Natomiast jeżeli f(c) < f(d) to minimum znajduje się po prawej stronie punktu c i zawężamy przedział poszukiwań do [a, d].



*Rysunek 2. Lokalizowanie minimum w przedziale [a, b]*

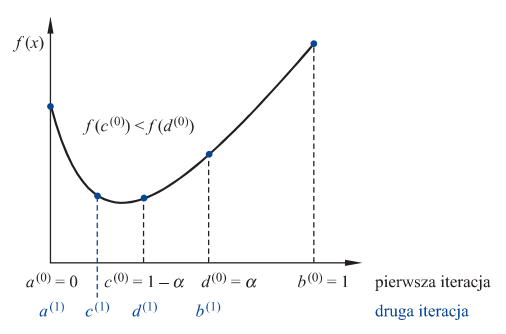
Znając przedział, w którym znajduje się minimum funkcji f, możemy go stopniowo zawężać, aż osiągnie długość równą zadanej dokładności obliczeń . Uzyskamy to, obliczając wartości funkcji w wybranych punktach przedziału, a efektywność tego podejścia zależy od kolejnego ich wyboru. Należy zauważyć, że ze względu na fakt, że jeden z punktów c, d używanych do redukowania przedziału poszukiwań leży zawsze wewnątrz nowego przedziału, może być on z powodzeniem wykorzystywany jako punkt próbny w następnym etapie poszukiwań.

1. **Metoda złotego podziału**

Jest to jedna z najczęściej stosowanych metod służących do zawężania przedziału poszukiwań. Idea metody oparta jest na założeniu, że w każdym kroku obliczeń długość przedziału poszukiwań zmniejszana jest w stałym stosunku . Aby to osiągnąć należy podzielić początkowy przedział poszukiwań na trzy podprzedziały spełniające zależność:

Obraz zawierający tekst

Opis wygenerowany automatycznie



*Rysunek 3. Zilustrowanie pierwszych dwóch iteracji metody*

Przekształcając poprzednią zależność otrzymujemy postaci i-tych wartości c oraz d:

Zauważmy, że zostało spełnione dla i = 0, zatem nowy przedział poszukiwań to , a prawym punktem pośrednim . Można zatem zapisać:

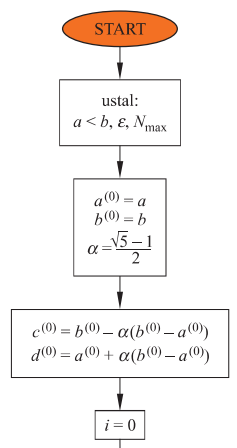
Finalnie, korzystając z otrzymanych zależności, wyznaczamy wartość współczynnika :

Równoważnie, możemy zapisać wynik w postaci:

Którego jedynym dodatnim rozwiązaniem jest , jest to tak zwana złota liczba.

1. **Implementacja metody w Matlabie**

Określamy przedział poszukiwań, dokładność oraz maksymalną liczbę iteracji, by przekazać je do funkcji realizującej metodę złotego przedziału. W utworzonej funkcji określamy punkty początkowe, nadajemy wartość alfa oraz określamy punkty wewnętrzne i zaczynamy iterowanie

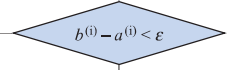


Obraz zawierający tekst

Opis wygenerowany automatycznie

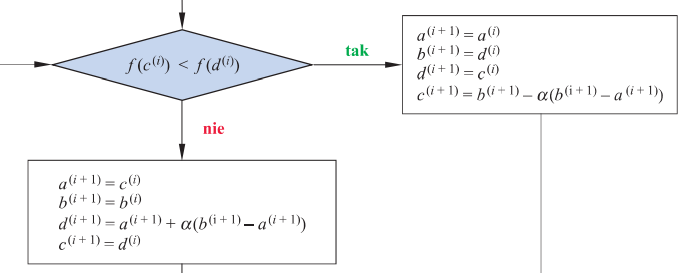
*Rysunek 4. Schemat blokowy algorytmu metody złotego podziału i odpowiadający mu kod w Matlabie*

Następnie tworzymy pętlę while, która będzie wykonywana dopóki szerokość końcowego przedziału jest większa od ustalonej dokładności oraz dopóki wykonaliśmy mniej iteracji niż mamy ustalone dla maksymalnej liczby iteracji



*Rysunek 5. Schemat blokowy algorytmu metody złotego podziału i odpowiadający mu kod w Matlabie*

W zależności od wartości funkcji w punktach wewnętrznych przesuwane są granice przedziałów oraz obliczany jest nowy punkt wewnętrzny, który stał się teraz jedną z granic przedziału.



Obraz zawierający tekst

Opis wygenerowany automatycznie

Ta funkcja wykonywana jest kiedy warunek jest spełniony.

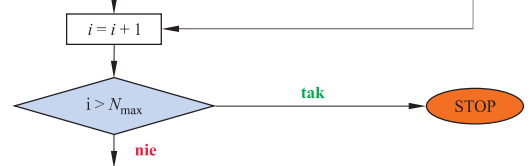
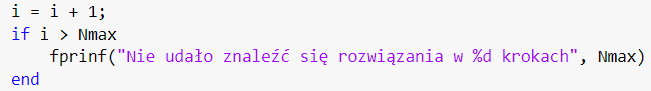
Obraz zawierający tekst

Opis wygenerowany automatycznie

A ta funkcja jest wykonywana kiedy .

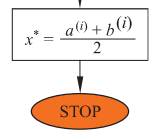
*Rysunek 6. Schemat blokowy algorytmu metody złotego podziału i odpowiadający mu kod w Matlabie*

Następnie zwiększamy licznik iteracji oraz sprawdzamy, czy nie napotkaliśmy warunku stopu – przekroczenie maksymalnej liczby iteracji.



*Rysunek 6. Schemat blokowy algorytmu metody złotego podziału i odpowiadający mu kod w Matlabie*

Po otrzymaniu ostatecznego przedziału wyliczamy przewidywane położenie minimum funkcji.



*Rysunek 7. Schemat blokowy algorytmu metody złotego podziału i odpowiadający mu kod w Matlabie*

1. **Wykorzystanie metody w ramach laboratorium**

W trakcie pracy na laboratorium należało dla dwóch podanych funkcji f(x) i g(x) wyznaczyć lokalizacje minimów z wykorzystaniem zaimplementowanej wcześniej metody złotego podziału. Po wyznaczeniu przedziałów początkowych przy użyciu metody ekspansji, którą napisaliśmy na poprzednich zajęciach, wystarczyło użyć je (nasze przedziały początkowe) w odpowiednio zaimplementowanej funkcji wykonującej algorytm złotego podziału. Poniżej przedstawiam napisaną przeze mnie funkcję pomocniczą do wyświetlania wyników oraz kod który wyświetla poszczególne funkcje.

Obraz zawierający tekst

Opis wygenerowany automatycznie

*Rysunek 8. Funkcja Display służąca do opisu funkcji*

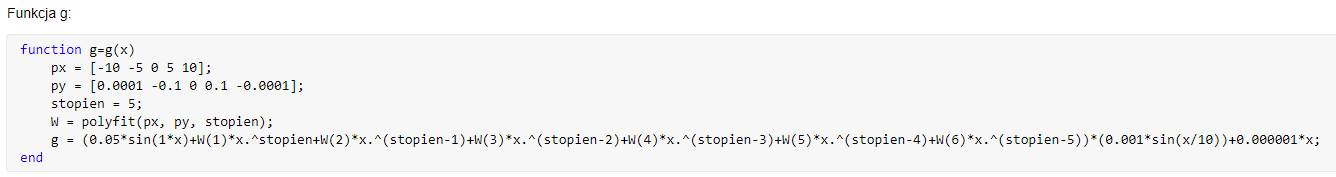
*Obraz zawierający tekst

Opis wygenerowany automatycznie*

*Rysunek 9. Część kodu w funkcji gold\_divide służący do plotowania wykresów.*

*Obraz zawierający tekst

Opis wygenerowany automatycznie*

*Rysunek 10. Funkcja f(x)*

*Rysunek 11. Funkcja g(x)*

*Obraz zawierający tekst

Opis wygenerowany automatycznie*

*Rysunek 12. Wywołanie funkcji f(x)*

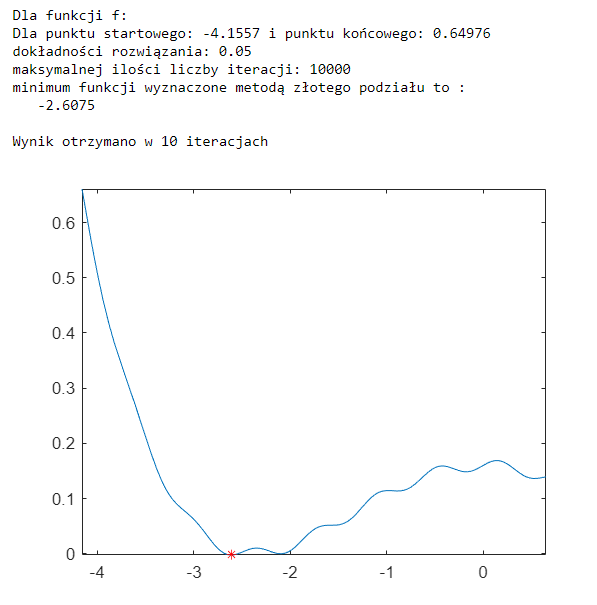
*Obraz zawierający tekst

Opis wygenerowany automatycznie*

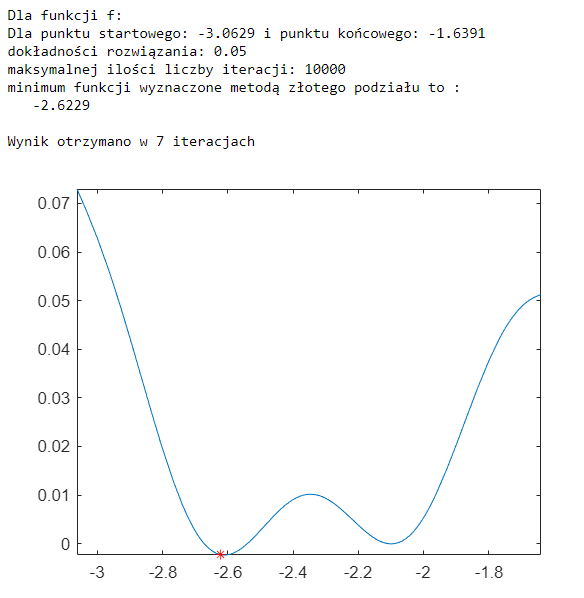
*Rysunek 13. Wywołanie funkcji g(x)*

1. **Wyniki**

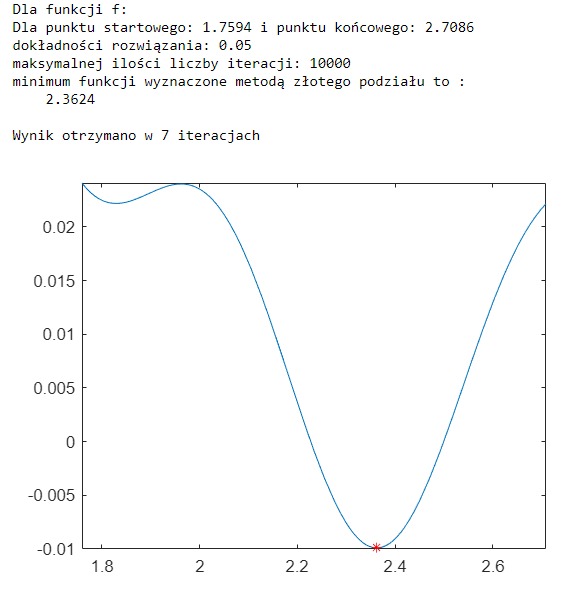
Dla wyżej przedstawionych funkcji oraz kodu wywołującego nasze wyniki dla funkcji f(x) otrzymałem następujące wykresy i informacje:



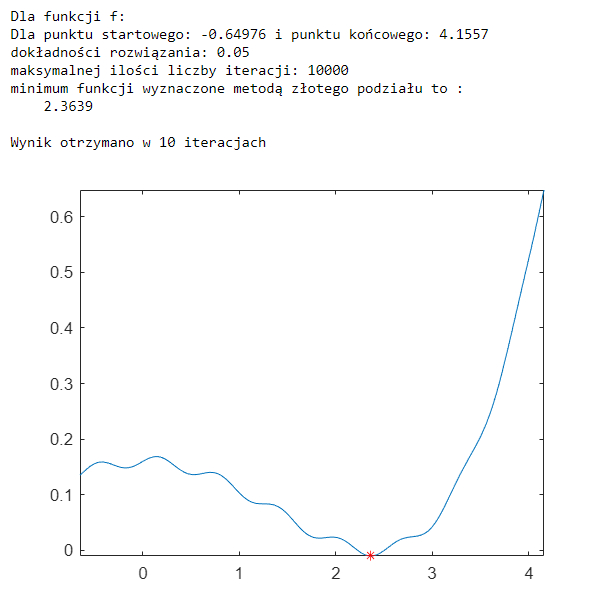
*Rysunek 14. Wynik wywołania wcześniej zaimplementowanych funkcji dla f(x)*



*Rysunek 15. Wynik wywołania wcześniej zaimplementowanych funkcji dla f(x)*

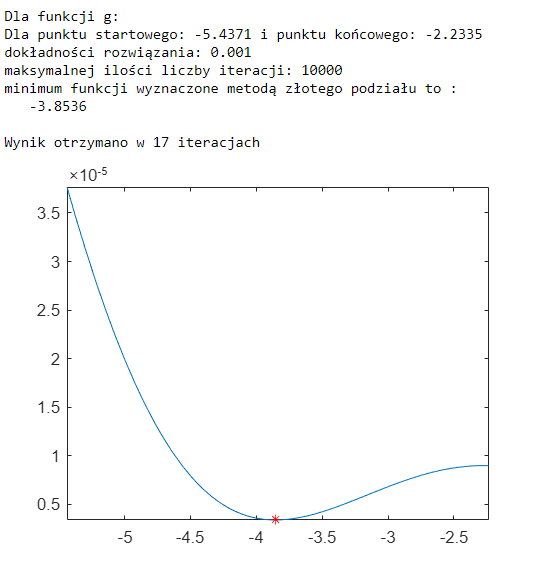
**

*Rysunek 16. Wynik wywołania wcześniej zaimplementowanych funkcji dla f(x)*

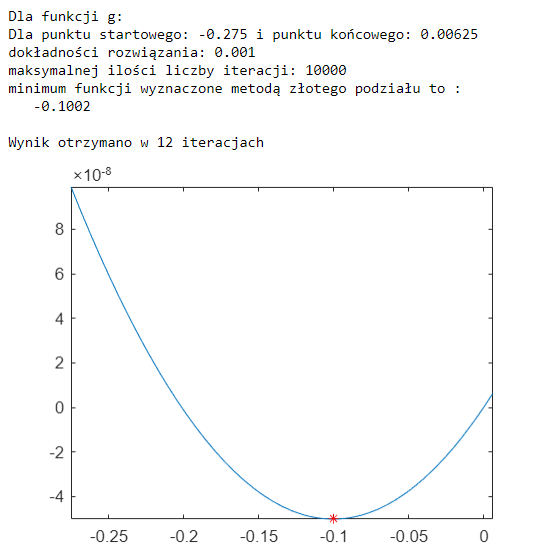
**

*Rysunek 17. Wynik wywołania wcześniej zaimplementowanych funkcji dla f(x)*

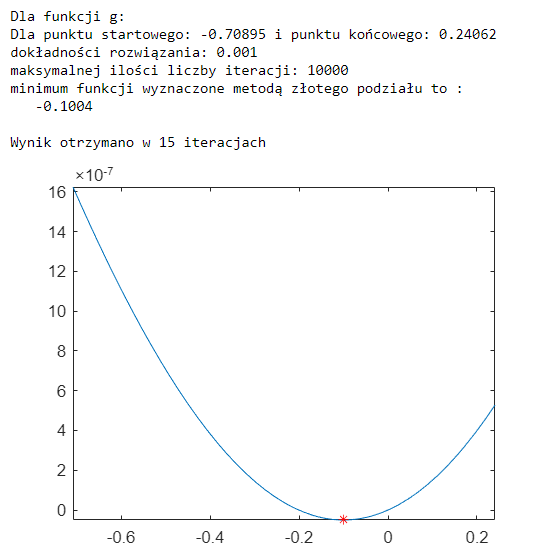
Dla wyżej przedstawionych funkcji oraz kodu wywołującego nasze wyniki dla funkcji g(x) otrzymałem następujące wykresy i informacje:



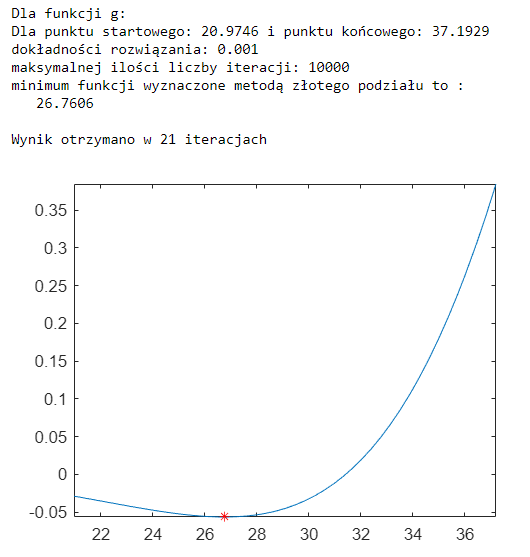
*Rysunek 18. Wynik wywołania wcześniej zaimplementowanych funkcji dla g(x)*

**

*Rysunek 19. Wynik wywołania wcześniej zaimplementowanych funkcji dla g(x)*



*Rysunek 19. Wynik wywołania wcześniej zaimplementowanych funkcji dla g(x)*

**

*Rysunek 20. Wynik wywołania wcześniej zaimplementowanych funkcji dla g(x)*

1. **Wnioski**

Jak możemy zauważyć na powyższych wynikach, pomimo tego że w przedziale znajduje się więcej niż jedno minimum to funkcja złotego podziału potrafi znaleźć to „najlepsze”. Metoda ta jest efektywna w zależności od tego jak dobry został wybrany przedział początkowy. Trzeba pamiętać, że w tej metodzie musimy mieć zawsze dwa punkty wewnętrzne. Metoda ta może się „zepsuć” jeśli w trakcie którejś iteracji wartości funkcji w punktach wewnętrznych będą równe. Trzeba także zauważyć, że metoda złotego podziału nie oblicza dokładnie wartości minimum funkcji lecz podaje jej przybliżenie.