**Informatyka II**

**Iteracyjne metody rozwiązywania równań nieliniowych**

**Krzysztof Haręza 304428**

Prowadzący: Michał Stachura

# Funkcja

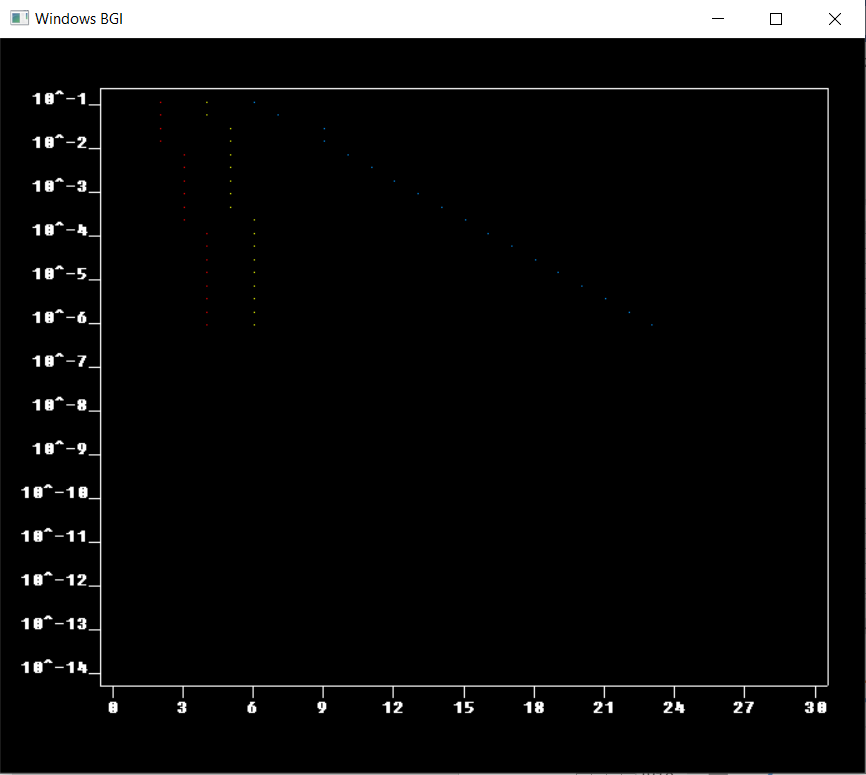
Przyjęte dolne i górne granice przedziału : a=-1 b=5 oraz : a=-5 b=5

# Przedstawienia graficzne

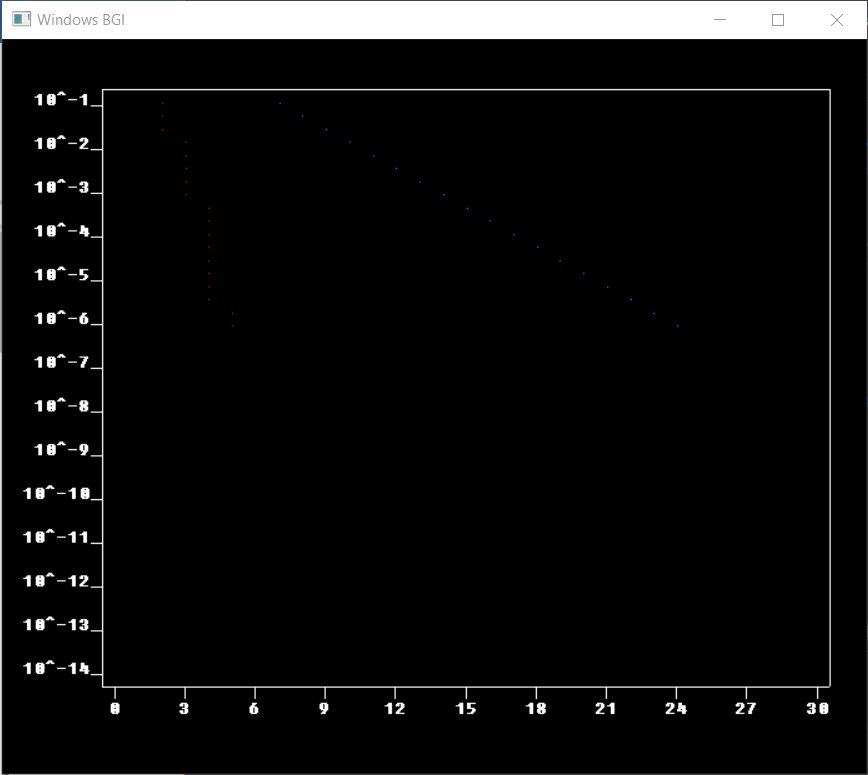
Czerwone kropki: Metoda siecznych

Żółte kropki: Metoda Newtona

Niebieskie kropki : Metoda bisekcji



|  |  |  |  |
| --- | --- | --- | --- |
| **Dokladnos** | **Iteracja-B** | **Iteracja-N** | **Iteracja-S** |
| 0.000001 | 23 | 6 | 4 |
| 0.000002 | 22 | 6 | 4 |
| 0.000004 | 21 | 6 | 4 |
| 0.000008 | 20 | 6 | 4 |
| 0.000015 | 19 | 6 | 4 |
| 0.000031 | 18 | 6 | 4 |
| 0.000061 | 17 | 6 | 4 |
| 0.000122 | 16 | 6 | 4 |
| 0.000244 | 15 | 6 | 3 |
| 0.000488 | 14 | 5 | 3 |
| 0.000977 | 13 | 5 | 3 |
| 0.001953 | 12 | 5 | 3 |
| 0.003906 | 11 | 5 | 3 |
| 0.007813 | 10 | 5 | 3 |
| 0.015625 | 9 | 5 | 2 |
| 0.031250 | 9 | 5 | 2 |
| 0.062500 | 7 | 4 | 2 |
| 0.125000 | 6 | 4 | 2 |
|  |  |  |  |



|  |  |  |  |
| --- | --- | --- | --- |
| **Dokladnos** | **Iteracja-B** | **Iteracja-N** | **Iteracja-S** |
| 0.000001 | 24 | 148 | 5 |
| 0.000002 | 23 | 148 | 5 |
| 0.000004 | 22 | 148 | 4 |
| 0.000008 | 21 | 148 | 4 |
| 0.000015 | 20 | 148 | 4 |
| 0.000031 | 19 | 148 | 4 |
| 0.000061 | 18 | 148 | 4 |
| 0.000122 | 17 | 148 | 4 |
| 0.000244 | 16 | 148 | 4 |
| 0.000488 | 15 | 148 | 4 |
| 0.000977 | 14 | 147 | 3 |
| 0.001953 | 13 | 147 | 3 |
| 0.003906 | 12 | 147 | 3 |
| 0.007813 | 11 | 147 | 3 |
| 0.015625 | 10 | 147 | 3 |
| 0.031250 | 9 | 147 | 2 |
| 0.062500 | 8 | 147 | 2 |
| 0.125000 | 7 | 146 | 2 |

# Opis programu

Program rozpoczynam od załączenia pliku nagłówkowego oraz źródłowego z funkcją bisec załączonych w zadaniu. Deklaruję niezbędne biblioteki oraz prototypy funkcji. Otwieram plik w którym będę przechowywał ilość iteracji oraz dokładność obliczeń. Zabezpieczam program przed ewentualnym niepoprawnym otwarciem lub błędem przy otwieraniu. Następnie proszę użytkownika o podanie z klawiatury dolnej i górnej granicy badanego przedziału. Program jest wyposażony w zabezpieczenie przeciw niepoprawnemu formatowi danych to jest np. znak zamiast cyfry. Używam do tego funkcji wyczyscbufor. Program od razu sprawdza czy zadany przedział spełnia warunek f(a)\*f(b)<0

Osiąga to poprzez wywołanie funkcji bisec i sprawdzenie wartości zmiennej iter, której przypisywane jest wartość -1 gdy przedział nie spełnia warunku. Następnie zostaje otwarte okno graficzne oraz przeskalowanie przedziałki na wykresie zgodnie z wymaganiami (logarytmiczna skala dla dokładności). W pętli for wywoływane są funkcje bisec, newton i siecz, które realizują poszczególne metody rozwiązywania równań nieliniowych. Funkcja bisec podana w zadaniu, natomiast newton oraz siecz są napisane tak by zwracały wartość miejsca zerowego wyliczanego metodami Newtona i Siecznych gdy osiągnięta zostanie dana dokładność. W funkcjach tych narzucona jest maksymalna ilość iteracji po której zostanie zwrócony otrzymany wynik niezależnie od dokładności. Przy dużych przedziałach może to być problemem i trzeba by odpowiednio te liczbę zwiększyć. Niestety nie da się jej przewidzieć przed rozwiązaniem równania daną metodą.

Wyniki funkcji są wypisywane na ekranie zgodnie z poleceniem dla dokładności idących . Ponadto ilość iteracji jest zapisywana do pliku. Przed każdym wywołaniem funkcji zmieniany jest kolor punktu na wykresie by odróżnić poszczególne metody. Ostatecznie stawiany na wykresie jest punkt dla każdej metody o współrzędnych (numer iteracji, zadana dokładność).

W następnej części programu rozwiązywane jest równanie . Funkcje zostały odpowiednio zmienione by możliwe było obliczanie tej funkcji. Zostało to zrealizowane przez dodanie zmiennej globalnej w, której na początku programu przypisywana jest wartość 1. Po rozwiązaniu pierwszej funkcji w drugiej pętli for, zmieniane są zgodnie z poleceniem wartości w=0.5,0.6,…,15.0 i obliczane jest miejsce zerowe funkcji wszystkimi trzema metodami. Wyniki wypisywane są na ekranie. Na końcu programu zamykany jest plik.

# Wnioski

Przy dużym przedziale, szczególnie zawierającym zarówno liczby dodatnie oraz ujemne, ilość iteracji w metodzie Newtona zdecydowanie się zwiększa. Widać to na drugim zamieszczonym wykresie. Natomiast metoda siecznych ma stosunkowo niewielką ilość iteracji i dużą dokładność.