Dokumentacja projektu nr1.

Autor: Marcel Kaliński

# Wstęp

Celem projektu było obliczenie miejsca zerowego metodą bisekcji w zadanym przez użytkownika przedziale. Ma być dostępny również wybór dokładności z jaką te miejsce zerowe wyznaczamy.

# Specyfikacja

Metoda bisekcji jest chyba najprostszą metodą numerycznego wyznaczania miejsca zerowego. Polega ona na zmniejszaniu obszaru poszukiwań o połowę w każdej iteracji. Aż do osiągnięcia zadanej dokładności lub przekroczenia maksymalnej liczby iteracji. Warunkiem koniecznym, aby metoda ta była zbieżna jest, aby krańce przedziału miały różne znaki. Kod do mojej funkcji wygląda następująco.

*double bisekcja(Function fun, double left, double right, double eps)*

*{*

*double a = left;*

*double b = right;*

*double x = 0;*

*//sprawdzenie, czy na przedziale jest miejsce zerowe*

*if(f(fun, a)\* f(fun, b) > 0)*

*{*

*printf("\*\*\*\*ERROR\*\*\*\*\*\n\*\*funkcja nie musi mieć miejsca zerowego na zadanym przedziale\*\*\n");*

*return -1;*

*}*

*int iters = 0;*

*while(fabs(a-b) > eps)*

*{*

*x = (a+b)/2;*

*double val = f(fun, x);*

*if(fabs(val) <= eps)*

*{*

*break;*

*printf("osiągnięto zadaną dokładność\n");*

*}*

*else if(val \* f(fun, a) < 0)*

*b = x;*

*else*

*a = x;*

*++iters;*

*}*

*printf("nr of iterations: %d\n", iters);*

*return (a+b)/2;*

*}*

Funkcję przechowuję w specjalnej strukturze:

*typedef struct f {*

*double a;*

*double b;*

*double c;*

*double d;*

*double e;*

*double f;*

*double g;*

*}Function;*

Jest ona postaci ax^b + cx^d + dx^e +ex^f + g.