Autorzy: Piotr Onyszczuk, Maciej Pióro, gr. F3

Sprawozdanie z projektu nr 1, Metody Numeryczne II

1. **Temat:** Metoda Bairstowa
2. **Opis metody.**

Metoda Bairstowa jest iteracyjnym algorytmem pozwalającym znaleźć pierwiastki wielomianu p(x) o współczynnikach rzeczywistych, wykorzystując wykonując obliczenia wyłącznie na liczbach rzeczywistych. Polega ona na wyznaczeniu współczynników q i r wielomianu stopnia drugiego , tak aby wielomian ten dzielił p(x).

, gdy

1. **Implementacja.**

**Parametry wejściowe:**

* **poly** - wektor współczynników wielomianu;
* **x0** - wektor z początkowymi wartościami r i q;
* **eps –** tolerancja błędu będąca warunkiem STOP dla głównej części metody;
* **eps2 –** tolerancja błędu podczas eliminowania pierwiastków wielokrotnych;
* **iter –** liczba iteracji, będąca warunkiem STOP dla głównej części metody;
* **nomultiples** – wartość logiczna, ustawiamy na true, gdy chcemy aby w wektorze wynikowym wszystkie pierwiastki wielokrotne występowały tylko raz.

Parametry eps, eps2, iter oraz nomultiples możemy pozostawić puste([]), zostaną im nadane wartości domyślne.

* 1. **Główna część metody.**

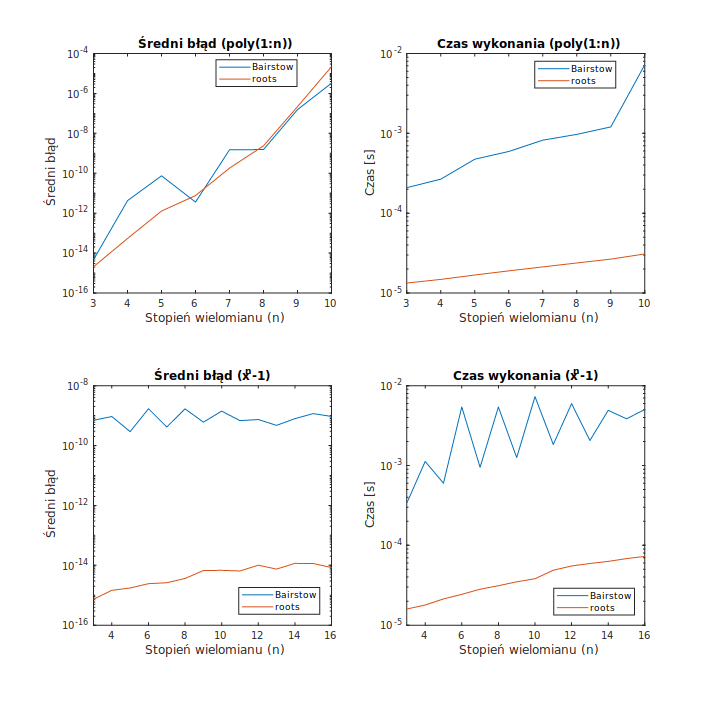
Posiadając początkowe wartości r i q, dzielimy początkowy wielomian p(x) przez wielomian stopnia drugiego (funkcja deconv). Otrzymujemy wielomian s(x) oraz pewną resztę. Korzystając z operacji określonych w metodzie Bairstowa wykonujemy odpowiednie przekształcenia, aby otrzymać dokładniejsze przybliżenie wartości r i q. Pętla wykonuje się dopóki norma wektora złożonego z współczynników otrzymanej reszty (są to dwa współczynniki) jest większa od zadanego w parametrach epsilon. Innym warunkiem stopu jest liczba iteracji. W przypadku przekroczenia liczby iteracji wywoływana jest metoda bisekcji. Po otrzymaniu odpowiedniego przybliżenia powtarzamy algorytm dla wielomianu s(x). Wykonujemy to dopóki wielomian ten ma jest co najmniej trzeciego stopnia.

* 1. **Metoda bisekcji.**

Problemem podczas używania metody bisekcji w tym projekcie jest fakt, że nie zawsze zadany wielomian ma miejsce zerowe, a jeśli posiada jakiś pierwiastek, to nie znamy również argumentów dla których przyjmowałby wartości o przeciwnych znakach. Jeśli wielomian jest stopnia nieparzystego, to na pewno posiada miejsce zerowe. Odsuwamy się zatem na osi OX od pewnego przybliżenia początkowego (w odpowiednim kierunku) tak długo aż znajdziemy argument dla którego przyjmuje wartość o przeciwnym znaku. W przypadku wielomianu o stopniu parzystym, miejsca zerowego poszukujemy w zadanym otoczeniu przybliżenia początkowego, przeszukując przedział przesuwając się o pewną wartość kroku. Gdy nie znajdziemy argumentów w których wielomian przyjmuje wartości o przeciwnych znakach, kończymy poszukiwanie miejsc zerowych informując użytkownika o niepowodzeniu i zwracając znalezione do tej pory miejsca zerowe.

* 1. **Usuwanie pierwiastków wielokrotnych.**

Jeżeli parametr nomultiples będzie miał wartość true, wszystkie pierwiastki wielokrotne wielomianu wystąpią tylko raz w wektorze wynikowym. Po znalezieniu wielomianu kwadratowego dzielącego p(x), dzielimy p(x) przez ten wielomian, dopóki reszta jest odpowiednio mała. Dla pierwiastków rzeczywistych dzielimy p(x) przez wielomiany stopnia pierwszego odpowiadające im, ponieważ w p(x) mogą być one różnej krotności.

1. **Przykłady**

* **Wielomiany o pierwiastkach 1:n, n=3,…,10**

Wielomiany wygenerowane funkcją poly. Na wykresach: suma modułów wartości kolejnych wielomianów obliczonych w pierwiastkach wyznaczonych przez funkcje Bairstow i roots podzielona przez ilość pierwiastków oraz czas wykonania metod zmierzony za pomocą timeit.

Parametry domyślne, punkt startowy [0, 0].

* **Wielomiany postaci , n=3,…,16**

Opis wykresów tak jak wyżej