Projekt z labu - metoda Bairstowa

Termin: ostatni lab czyli styczeń 2013

Projekt składa się z dwóch części.

Funkcja z metodą Bairstowa

Napisać funkcję octave'a z metodą Bairstowa: tzn. zaprogramować funkcję octave'a bairstow() w m-pliku bairstow.m z ***metodą Bairstowa*** opisaną np w skrypcie prof Kiciaka:
Metody Numeryczne, str. 1.24, czy w książce Kincaid, Cheney, Analiza numeryczna, WNT, 2006, str. 107-111, czy inne źródła w sieci.

Metoda służy znalezieniu przybliżenia zer wielomianu $p = \sum_{k=0}^{n} a_k x^k$ stopnia $n > 2$ (nawet zespolonych!), a dokładniej znajduje przybliżenia u_1, u_2 takich, że $d = c = 0$ dla

$$p(x) = (x^2 - u_1 x - u_2) * w(x) + c * (x - u_1) + d \tag{2.1}$$

Tutaj $w(x) = \sum_{k=0}^{n-2} b_k x^k$ wielomian stopnia $n - 2$, zależny od u_1, u_2.

Oczywiście $d = d(u_1, u_2)$ i $c = c(u_1, u_2)$ tak samo współczynniki w zależą od u_1, u_2, metoda Bairstowa to po prostu wielowymiarowa metoda Newtona zastosowana do równania $F(u_1, u_2) = (d(u_1, u_2), c(u_1, u_2)) = 0$, szczegóły np. w skrypcie prof. Kiciaka.

Jeśli dla $\vec{u}^{(k)} = (u_1^{(k)}, u_2^{(k)})$ zachodzi

$$p(x) = (x^2 - u_1^{(k)} x - u_2^{(k)}) * w_k(x) + c_k * (x - u_1^{(k)}) + d_k \tag{2.2}$$

i $\|F(\vec{u}^{(k)})\|_\infty = \max(|c_k|, |d_k|) \leq TOL = 10^{-12}$ lub przekroczona została maksymalna ilość iteracji ustalona na $MAXIT = 100$, wtedy kończymy iteracje metody.

Parametrami funkcji octave'a bairstow() mają być:

- $\vec{p}$ wektor wymiaru $n + 1$ współczynników wielomianu p
- wektor $\vec{u}_0$ wymiaru dwa, czyli przybliżenie startowe dla metody Bairstowa
- n stopień wielomianu

Jeśli stopień $n \leq 2$ to funkcja powinna zwrócić komunikat o błędnych danych wejściowych.

Funkcja ma zwracać trzy rzeczy

- wektor $\vec{u}$ wymiaru dwa z obliczonym za pomocą metody Bairstowa przybliżeniem u_1, u_2.
- wektor $\vec{w}$ wymiaru $n - 1$ z przybliżeniami współczynników wielomianu w z (2.1).
- r liczbę równą $\max(|c_k|, |d_k|)$ dla c_k, d_k jak w (2.2).

W razie przekroczenia ustalonej maksymalnej ilości iteracji funkcja ma wyświetlić komunikat na ekranie.

Testy

Trzeba przeprowadzić testy tej funkcji w porównaniu z funkcją octave'a **roots**().

Przetestować na następujących przykładach jak działa metoda tzn. wyznaczyć wszystkie zera następujących wielomianów korzystając z funkcji bairstow() i porównać z tym co zwróci funkcja **roots**():

1. $w_1(x) = (x-1)(x-2)(x-3)(x-4)$
2. $w_2(x) = x^k - 1$ dla $k = 2, 3, 4, 10$ - pierwiastki zespolone z jedynki.
3. $w_3(x) = x^4 - 4 * x^3 + 7x^2 - 5x - 2$
4. $w_4(x) = (x^2 - 2 * x + 2)^2 (x-2)^2$ - tu dwukrotne pierwiastki w tym para zespolonych sprzężonych pierwiastków krotności dwa
5. $w_4(x) = (x^2 - 2 * x + 2)^4$ - tu para sprzężonych pierwiastków krotności cztery

Za przybliżenia startowe brać albo wektor zerowy, albo wektor losowy albo przy wyszukiwaniu kolejnych pierwiastków danego wielomianu ostatnie przybliżenie z poprzedniej aplikacji metody Bairstowa.

Przy zaliczaniu projektu mogę się spytać o idee metody Bairstowa i oczywiście o przeprowadzenie testu dla jakiegoś innego wielomianu.

Uwaga Funkcja octave'a **roots**() (i inne funkcje działające na wielomianach w octave'ie) inaczej numeruje współczynniki wielomianu w bazie jednomianów tzn. wielomian stopnia n jest przedstawiany jako $w(x) = \sum_{k=1}^{n+1} a_k x^{n-k+1}$, np. aby znaleźć zera $2x^4 - 1$ należy wywołać **roots**($[2\ 0\ 0\ 0\ -1]$) . W funkcji bairstow() i testach możecie Państwo zachować konwencje octave'a lub przyjąć np. standardową.