Sprawozdanie - laboratorium 6

Poszukiwanie zer wielomianów metodą iterowanego dzielenia (metoda Newtona)

Katarzyna Such, 18.04.2021

1 Wstęp teoretyczny

Poszukujemy zer wielomianu:

$$f(x) = a_n x^n + a_{n-1} x^{n-1} + ... + a_1 x^1 + a_0 = 0 \tag{1}$$

Po podzieleniu wielomianu przez $(x - x_j)$ otrzymamy:

$$f(x) = (x - x_j)(b_{n-1} x^{n-1} + b_{n-2} x^{n-2} + ... + b_0) + R_j \tag{2}$$

Współczynniki powyższego wielomianu wyznaczamy ze wzorów:

$$b_n = 0, \tag{3}$$

$$b_k = a_{k+1} + x_j b_{k+1}, \quad k = n-1, n-2, ..., 0, \tag{4}$$

$$R_j = a_0 + x_j b_0. \tag{5}$$

Po kolejnym podzieleniu przez $(x - x_j)$:

$$f(x) = (x - x_j)^2 (c_{n-2} x^{n-2} + c_{n-3} x^{n-3} + ... + c_0) + R'_j (x - x_j) + R_j \tag{6}$$

Gdzie:

$$c_n = 0, \tag{7}$$

$$c_k = b_{k+1} + x_j c_{k+1}, \quad k = n-1, n-2, ..., 0, \tag{8}$$

$$R'_j = b_0 + x_j c_0 \tag{9}$$

Korzystając z metody Newtona zera wielomianu wyznaczamy stosując formułę:

$$x_{j+1} = x_j - \frac{R_j}{R'_j}, \tag{10}$$

gdzie x_{j+1} to kolejne przybliżenia zera.

Metoda Newtona polega na prowadzaniu stycznej do wykresu badanej funkcji, w pewnym przedziale [a ,b] gdzie funkcja ma ten sam znak co jej druga pochodna. Punkt x_1, który jest przecięciem stycznej z osią OX przyjmujemy jako rozwiązanie. Sprawdzamy czy f(x_1) = 0. Jeśli nie, to wyznaczamy kolejne styczne do momentu uzyskania satysfakcjonującego wyniku:

$$|x_{k+1} - x_k| < \epsilon \tag{11}$$

2 Zadanie do wykonania

2.1 Opis problemu

Podczas laboratorium poszukiwaliśmy wszystkich zer wielomianu: $f(x) = x^5 + 14x^4 + 33x^3 - 92x^2 - 196x + 240$. Przyjęliśmy startowe $x_0 = 0$ oraz IT_MAX = 30.
Zera wielomianu wyznaczaliśmy stosując metodę iterowanego dzielenia, której implementacja wygląda następująco:

```
N = 5  // stopień wielomianu
a[0] = 240; //inicjalizacja wektora danych
a[1] = -196;
a[2] = -92.0;
a[3] = 33.0;
a[4] = 14.0;
a[5] = 1.0;

//pętla po kolejnych zerach wielomianu
for(int L=1; L<=N; L++){

    n = N-L+1; //aktualny stopień wielomianu
    x0 = 0; //inicjalizacja wzoru iteracyjnego

    for(int it=1; it<=IT_MAX; it++){
        Rj = licz_r(a,b,n, x0);
        Rj_p = licz_r(b,c,n-1, x0);
        x1 = x0 - (Rj/Rj_p);

        fprintf(f, "%12d %12d %12g %12g %12g\n", L, it, x1, Rj, Rj_p); // zapis do pliku

        if (fabs(x1-x0) < 1.0e-7) break; //warunek wcześniejszego opuszczenia pętli
        x0 = x1; // nowe przybliżenie

    }

    for(int i=0; i<=(n-1); i++) a[i] = b[i]; // redukcja stopnia wielomianu o 1
}
```

gdzie licz_r(a,b,n,x0) to funkcja wyznaczająca współczynniki wielomianu o stopień niższego:

```
double licz_r(double *a, double *b, int n, double xj){
    b[n] = 0;
    for(int k=n-1; k>=0; k--){
        b[k] = a[k+1]+xj*b[k+1];
    }
    return a[0]+xj*b[0];
}
```

Argumenty przyjmowane przez funkcje:
- $\vec{a}$ - wektor zawierający współczynniki aktualnego wielomianu,
- $\vec{b}$ - wektor do którego zostaną wpisane współczynniki o stopień niższego wielomianu,
- n - stopień wielomianu,
- x0 - wartość dla której funkcja ma zwracać Rj

Funkcja zwraca Rj dla danego x. Stosujemy te funkcje dwa razy. Pozwala to na wyznaczenie wektora $\vec{c}$, oraz wartości R'_j. Posiadając R_j i R'_j wyznaczamy kolejne przybliżenie x.

2.2 Wyniki

W każdej iteracji zapisywaliśmy do pliku: numer zera - L, numer iteracji - it, wartości kolejnych przybliżeń x_j oraz wartości reszty z dzielenia R_j i R'_j.

Znalezione zero wielomianu $x = 1$:

L	**it**	$\mathbf{x_1}$	$\mathbf{R}_j$	$\mathbf{R}'_j$
1	1	1.22449	240	-196
1	2	0.952919	-43.1289	-158.813
1	3	0.999111	10.5714	-228.86
1	4	1	0.195695	-220.179
1	5	1	$7.96468 \cdot 10^{-5}$	-220
1	6	1	$1.32729 \cdot 10^{-11}$	-220

Tabela 1: Wyniki dla pierwszego zera.

Znalezione zero wielomianu $x =$ -4:

L	**it**	$\mathbf{x_1}$	$\mathbf{R}_j$	$\mathbf{R}'_j$
2	1	-5.45455	-240	-44
2	2	-4.46352	-120.975	122.071
2	3	-4.10825	-24.2755	68.3304
2	4	-4.00957	-4.31754	43.7539
2	5	-4.00009	-0.347977	36.6891
2	6	-4	-0.00323665	36.0065
2	7	-4	$-2.90891 \cdot 10^{-7}$	36

Tabela 2: Wyniki dla drugiego zera.

Znalezione zero wielomianu x = 2:

L	**it**	$\mathbf{x}_1$	$\mathbf{R}_j$	$\mathbf{R}'_j$
3	1	15	-60	4
3	2	9.20218	5850	1009
3	3	5.53752	1687.53	460.488
3	4	3.38316	469.259	217.818
3	5	2.33534	118.159	112.767
3	6	2.0277	22.07	71.739
3	7	2.00021	1.67505	60.9441
3	8	2	0.0128842	60.0073
3	9	2	$7.83733 \cdot 10^{-7}$	60

Tabela 3: Wyniki dla trzeciego zera.

Znalezione zero wielomianu x = -3:

L	**it**	$\mathbf{x}_1$	$\mathbf{R}_j$	$\mathbf{R}'_j$
4	1	-2.30769	30	13
4	2	-2.94284	5.32544	8.38462
4	3	-2.99954	0.403409	7.11433
4	4	-3	0.00321531	7.00092
4	5	-3	$2.10929 \cdot 10^{-7}$	7

Tabela 4: Wyniki dla czwartego zera.

Znalezione zero wielomianu x = -10:

L	**it**	$\mathbf{x}_1$	$\mathbf{R}_j$	$\mathbf{R}'_j$
5	1	-10	10	1
5	2	-10	0	1

Tabela 5: Wyniki dla piątego zera.

3 Wnioski

Dzięki metodzie iterowanego dzielenia udało się nam rozwiązać problem: znaleźć zera wielomianu.

Metoda Newtona pozwala szybko odnaleźć dobre przybliżenie zera wielomianu w niewielkiej ilości iteracji. Jest też prosta do zaimplementowania.

Kolejne punkty x_k leżą coraz bliżej pierwiastka funkcji, a odległość pomiędzy x_{k+1} i x_k maleje. Do otrzymania wyniku potrzeba było minimalnie dwóch, maksymalnie dziewięciu iteracji.