Sprawozdanie – Laboratorium nr 6

Poszukiwanie zer wielomianów metodą iterowanego dzielenia (metoda Newtona).

Tomasz Rajchel
2019/04/04

Wstęp teoretyczny

Metoda dzielenia wielomianu

Dany jest wielomian, którego zera chcemy znaleźć:

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

Jeśli podzielimy wielomian przez wyraz $(x - x_j)$ to otrzymamy:

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

Z porównania współczynników przy jednakowych potęgach otrzymujemy zależność:

$$\begin{gathered}a_n=b_{n-1}\\ a_{n-1}=b_{n-2}-x_j b_{n-1}\\ \vdots\\ a_1=b_0-x_j b_1\\ a_0=R_j-x_j b_0\end{gathered}$$

Zatem współczynniki nowego wielomianu można wyliczyć rekurencyjnie:

$$\begin{gathered}b_n=0\\ b_k=a_{k+1}+x_j b_{k+1}, k=n-1, n-2, \ldots, 0\\ R_j=a_0+x_j b_0\end{gathered}$$

Dzieląc jeszcze raz wielomian otrzymamy:

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

Gdzie współczynniki c_n oraz czynnik R'_j wyznaczamy identycznie jak b_n oraz R_j.

Znajdowanie zer wielomianu

W metodzie Newtona zera wielomianu możemy wyznaczać iteracyjnie zgodnie z poniższym wzorem (Wzór jednokrokowy metody Newtona):

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

gdzie x_{j+1} to kolejne, lepsze przybliżenie zera, a czynniki R_j i R'_j (reszty z dzielenia) wyznaczamy zgodnie ze wzorami powyżej.

Po znalezieniu pierwiastka wielomianu usuwamy go (redukując tym samym stopień o 1), a następnie szukamy następnego.

Opis zadania

Celem zadania jest znalezienie wszystkich zer wielomianu:

$$f(x)=x^5+14x^4+33x^3-92x^2-196x+240$$

Dodatkowo należy zapisać do pliku numer zera(pierwiastka wielomianu), numer iteracji, wartość przybliżenia x_j oraz wartość reszty z dzielenia R_j i R'_j .

Skorzystamy z opisanego we wstępie teoretycznym wzoru jednokrokowego metody Newtona:

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

,który korzysta z napisanej przez nas funkcji do znalezienia reszt z dzielenia przez wielomian. Jako wartość startową x_0 przyjmujemy 0.

Kod napiszemy w języku C++, nie korzystając z żadnych bibliotek numerycznych.

Przerywamy działanie algorytmu gdy różnica pomiędzy kolejnymi przybliżeniami jest mniejsza od 10^{-7}.

Wyniki

L	iteracja	x_{it}	R_{it}	R'_{it}
1	1	0	240	-196
1	2	1.22449	-43.1289	-158.813
1	3	0.952919	10.5714	-228.86
1	4	0.999111	0.195695	-220.179
1	5	1	7.96468e-05	-220
1	6	1	1.32729e-11	-220

L	iteracja	x_{it}	R_{it}	R'_{it}
2	1	0	-240	-44
2	2	-5.45455	-120.975	122.071
2	3	-4.46352	-24.2755	68.3304
2	4	-4.10825	-4.31754	43.7539
2	5	-4.00957	-0.347977	36.6891
2	6	-4.00009	-0.00323665	36.0065
2	7	-4	-2.90891e-07	36

L	iteracja	x_{it}	R_{it}	R'_{it}
3	1	0	-60	4
3	2	15	5850	1009
3	3	9.20218	1687.53	460.488
3	4	5.53752	469.259	217.818
3	5	3.38316	118.159	112.767
3	6	2.33534	22.07	71.739
3	7	2.0277	1.67505	60.9441
3	8	2.00021	0.0128842	60.0073
3	9	2	7.83733e-07	60

L	iteracja	x_{it}	R_{it}	R'_{it}
4	1	0	30	13
4	2	-2.30769	5.32544	8.38462
4	3	-2.94284	0.403409	7.11433
4	4	-2.99954	0.00321531	7.00092
4	5	-3	2.10929e-07	7

L	iteracja	x_{it}	R_{it}	R'_{it}
5	1	0	10	1
5	2	-10	0	1

w kolumnach kolejno:
L – numer miejsca zerowego,
it – numer iteracji,
x_{it} – przybliżenie miejsca zerowego w danej iteracji,
R_{it} – reszta z dzielenia wielomianu w danej iteracji,
R'_{it} – reszta z powtórnego dzielenia wielomianu w danej iteracji.

Wnioski

Poprawnie wyznaczyliśmy pierwiastki wielomianu. Metoda iterowanego dzielenia jest prosta do zaimplementowania oraz dosyć szybka (zbieżność kwadratowa).