Analiza numeryczna(M) - Pracownia 2 - Zadanie 6

Wojciech Balik

Grudzień 11, 2016

Spis treści

1 Postacie wielomianów

Problem interpolacji dla danych punktów (x_k, y_k) $(k = 0, 1, ..., n)$ ma w przestrzeni Π_n(przestrzeni wielomianów stopnia co najwyżej n) dokładnie jedno rozwiązanie.Można jednak przedstawić ten wielomian na różne sposoby.Najprostszą do skonstruowania jest postać Lagrange'a:

$$W(x) = \sum_{k=0}^{n} y_k \lambda_k(x), \qquad \lambda_k(x) = \prod_{\substack{j=0 \\ j \neq k}}^{n} \frac{x - x_j}{x_k - x_j} \tag{1}$$

Nie jest ona jednak używana do celów praktycznych ze względu na to że obliczenie wartości $W(x)$ ma złożoność obliczeniową $O(n^2)$ a duża liczba wykonywanych operacji arytmetycznych może stwarzać dość duże błędy numeryczne.Wadą tej postaci jest także to że dodanie nowego węzła interpolacyjnego zmusza nas do obliczania wszystkich współczynników od nowa.
Zdefiniujmy:

$$\lambda(x) = (x - x_0)(x - x_1) \dots (x - x_n) = \prod_{j=0}^{n} x - x_j \tag{2}$$

$$w_k = \frac{1}{\prod\limits_{\substack{j=0 \\ j \neq k}}^{n} x_k - x_j} \tag{3}$$

Zauważmy że we wzorze (1), λ_k możemy zapisać w następującej postaci:

$$\lambda_k(x) = \lambda(x) \frac{w_k}{x - x_k} \tag{4}$$

Co po podstawieniu do wzoru (1) daje:

$$W(x) = \lambda \sum_{k=0}^{n} y_k \frac{w_k}{x - x_k} \tag{5}$$

Załóżmy teraz że interpolujemy wielomian stopnia zerowego $Q(x) \equiv 1$ w punktach $x_0, x_1, \ldots, x_n$,wtedy:

$$Q(x) \equiv 1 = \lambda \sum_{k=0}^{n} \frac{w_k}{x - x_k} \tag{6}$$

Dzieląc wzór (5) przez (6) otrzymujemy postać barycentryczną:

$$W(x) = \frac{\sum_{k=0}^{n} y_k \frac{w_k}{x - x_k}}{\sum_{k=0}^{n} \frac{w_k}{x - x_k}} \tag{7}$$

Zauważmy że dla wartość $W(x)$ jest określona tylko dla $x \neq x_k$.Aby wielomian interpolacyjny był dobrze określony definiujemy:

$$P(x) = \begin{cases} y_k & x = x_k, (k = 1, 2, \ldots, n) \\ W(x) & wpp. \end{cases}$$

Wartości w_k możemy obliczyć w czasie $O(n^2)$ ale gdy już to zrobimy, koszt obliczenia wartości wielomianu P w punkcie x to $O(n)$.Warto dodać że w przeciwieństwie do postaci Lagrange'a, dodając nowy węzeł interpolacyjny, możemy zaktualizować wartości w_k w czasie liniowym.

Ciekawą alternatywą jest także postać Newtona:

$$W(x) = \sum_{k=0}^{n} a_k \prod_{j=0}^{k-1} x - x_j, \qquad (\prod_{j=0}^{-1} \cdot \equiv 1) \tag{8}$$

Dla ilorazu różnicowego zdefiniowanego jako:

$$f[x_0, x_1, \ldots, x_k] = \sum_{i=0}^{k} \frac{f(x_i)}{\prod\limits_{\substack{j=0 \\ j \neq i}}^{k} x_i - x_j} \tag{9}$$

Współczynniki a_k możemy zapisać w postaci:

$$a_k = f[x_0, x_1, \ldots, x_k] \tag{10}$$

Zauważmy że używając wzoru rekurencyjnego:

$$f[x_i, x_{i+1}, \ldots, x_{i+j}] = \frac{f[x_{i+1}, \ldots, x_{i+j}] - f[x_i, \ldots, x_{i+j-1}]}{x_{i+j} - x_i} \tag{11}$$

Możemy obliczyć współczynniki a_k w czasie $O(n^2)$, gdzie w każdej iteracji wypełniana zostaje kolejna kolumna macierzy,korzystając z wyliczonych już wcześniej wartości:

$$\begin{array}{llll} f[x_0] & & & \\ f[x_1] & f[x_0, x_1] & & \\ f[x_2] & f[x_1, x_2] & f[x_0, x_1, x_2] & \\ \vdots & \vdots & \vdots & \ddots \\ f[x_n] & f[x_{n-1}, x_n] & f[x_{n-2}, x_{n-1}, x_n] & \cdots \quad f[x_0, x_1, \ldots, x_n] \end{array}$$

Szczegóły można doczytać w [**?**, s.312]

2 Algorytmy związane z interpolacją wielomianową

2.1 Obliczanie wartości wielomianu interpolacyjnego

Jak już zostało wspomniane, postać Lagrange'a jest dość niewygodna ze względów praktycznych.Lepszym pomysłem jest obliczenie wartości $w_i, (i = 0, 1, \ldots, n)$ postaci barycentrycznej, podanych we wzorze (3).Możemy to zrobić za pomocą algorytmu:

1. $M_0^{(0)} := 1$
2. $M_k^{(0)} := 0, \quad (k = 1, 2, \ldots, n)$
3. Dla każdego $i = 1, 2, \ldots, n \quad k = 0, 1, \ldots, i-1$ wykonaj:
   (a) $M_k^{(i)} := M_k^{(i-1)}/(x_k - x_i)$
   (b) $M_i^{(k+1)} := M_i^{(k)} - M_k^{(i)}$
4. $w_i := M_i^{(n)}, \quad (i = 0, 1, \ldots, n)$

Okazuje się że jeśli punkty $x_0, x_1, \ldots, s_n$ uporządkujemy względem średniej $\overline{X} = \sum_{k=0}^{n} x_k/(n+1)$, czyli:

$$x_i \prec x_j \Leftrightarrow |x_i - \overline{X}| < |x_j - \overline{X}| \tag{12}$$

To wpływa to pozytywnie na otrzymany wyniki, pod względem potencjalnych błedów numerycznych powastałych w wyniki wykonania algorytmu.(O inncyh sposobach porządkowania punktów x_k można doczytać w [**?**])

2.2 Zamiana postaci Lagrange'a na postać Newtona

Załóżmy że mamy dane współczynniki σ_k wielomianu w postaci Lagrange'a(1):

$$W(x) = \sum_{k=0}^{n} \sigma_k \prod_{\substack{j=0 \\ j\neq k}}^{n} x - x_j$$

$$\sigma_k = \frac{f(x_k)}{\prod_{\substack{j=0 \\ j\neq k}}^{n} x_k - x_j}$$

Wtedy z definicji ilorazu różnicowego:

$$a_k = \sum_{i=0}^{k} \sigma_i \prod_{j=k+1}^{n} (x_i - x_j), \qquad (k = 0, \ldots, n) \tag{13}$$

Wówczas wartości a_k można obliczać następującym algorytmem, operując na macierzy M rozmiaru $(n+1) \times (n+2)$:

1. $M_i^{(0)} := \sigma_i, \quad (i = 0, 1, \ldots, n)$
2. Dla każdego k od 0 do n:
   (a) $M_{n-k}^{(k+1)} := \sum_{i=0}^{n-k} M_i^{(k)}$.
   (b) $M_i^{(k+1)} := (x_i - x_{n-k}) M_i^{(k)}, \quad (i = n-k-1, \ldots, 0)$.
3. $a_i := M_i^{(n+1-i)}, \quad (i = 0, 1, \ldots, n)$

$$
\begin{array}{cccccc}
\sigma_0 & (x_0 - x_n)\sigma_0 & (x_0 - x_n)(x_0 - x_{n-1})\sigma_0 & \ldots & \prod_{j=1}^{n}(x_0 - x_j) & \prod_{j=1}^{n}(x_0 - x_j) \\
\sigma_1 & (x_1 - x_n)\sigma_1 & (x_1 - x_n)(x_1 - x_{n-1})\sigma_1 & \ldots & \sum_{i=0}^{1}\prod_{j=2}^{n}(x_1 - x_j)\sigma_i & 0 \\
\sigma_2 & (x_2 - x_n)\sigma_2 & (x_2 - x_n)(x_2 - x_{n-1})\sigma_2 & \ldots & 0 & 0 \\
\vdots & \vdots & \vdots & & \vdots & \vdots \\
\sigma_{n-1} & (x_{n-1} - x_n)\sigma_{n-1} & \sum_{i=0}^{n-1}(x_i - x_n)\sigma_i & \ddots & \vdots & \vdots \\
\sigma_n & \sum_{i=1}^{n}\sigma_i & 0 & \ldots & 0 & 0
\end{array}
$$

Zapominając o pierwszej kolumnie ,widzimy że wartościami a_k są elementy na przekątnej macierzy M

2.3 Wnioski

Zauważmy że przy implementacji powyższych algorytmów nie musimy korzystać z tablicy dwuwymiarowej, co pozwala na użycie jedynie $O(n)$ pamięci.Ponadto, algorytm (2.1) wykonuje $n(n-1)$ odejmowań oraz $\frac{n(n-1)}{2}$ dzieleń, w przeciwieństwie do algorytmu naiwnego(obliczania tych wartośći korzystając wprost ze wzoru z definicji) w którym występuje n^2 odejmowań oraz dzieleń.Powinno to zmniejszyć nie tylko czas wykonania, ale także ilość błedów numerycznych powstałych w wyniku zaokrąglania obliczanych wartości.

3 Testy Numeryczne

Obliczenia zostały przprowadzone w następujący sposób:

1. Zostało wylosowane 100 wielomianów stopnia co najwyżej 6
2. Każdy z tych wielomianów został zinterpolowany w n równoodległych punktach na przedziale $[-10, 10]$, na 4 sposoby:
   (a) Postać Lagrange'a, gdzie wartości są obliczane wprost ze wzoru (1)
   (b) Postać Newtona(używając algorytmu ilorazów różnicowych), wartości są obliczane za pomocą uogólnionego schematu Hornera
   (c) Postać Barycentryczna, wyznaczona poprzez algorytm (2.1)
   (d) Postać Barycentryczna, wyznaczona poprzez algorytm (2.1) z porządkiem względem średniej
   (e) Poprzez postać Lagrange'a, zmienonej na postać Newtona za pomocą algorytmu (2.2)

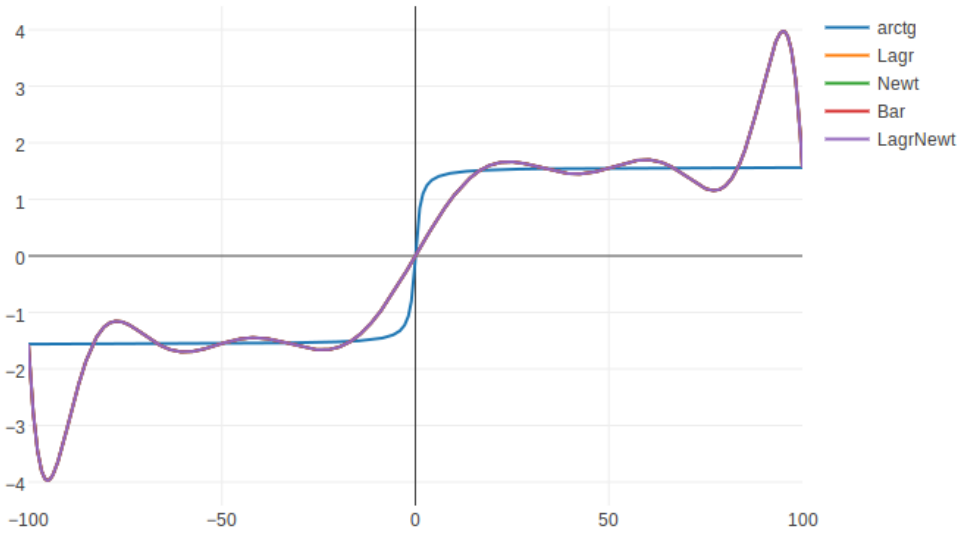
   Zatem jak już zostało wspomniane, dla $n > 6$ istnieje dokładnie jeden taki wielomian interpolacyjny.W praktyce jednak, te wielomiany różnią się ze względu na błędy numeryczne powstałe poprzez poprzez zaokrąglenia przy wyznaczaniu współczynników wielomianów (a)-(e).Chcemy sprawdzić jak bardzo wartości otrzymanych wielomianów różnią się od wartości rzeczywistych.W tym celu dla każdej z metod (a)-(e):
3. Dla każdego ze stu wielomianów na przedziale $[-100, 100]$ obliczamy błąd względny dla 1000 punktów, sumujemy wszystkie błędy i uśredniamy wynik, dzieląc przez 1000.
4. Podobnie jak w punkcie poprzednim, sumujemy średni błąd dla każdego wielomianu i dzielimy przez n, uzyskując średni błąd względny dla danego n

Poniższa tabelka prezentuje wyniki wyżej opisanych obliczeń, wykonanych używając arytmetyki podwójnej precyzji(double)

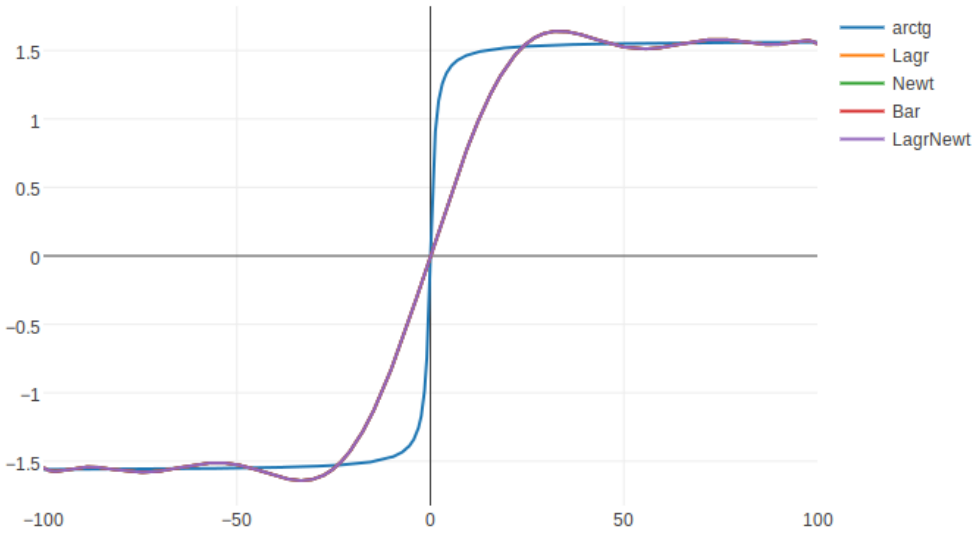
n	(a)	(b)	(c)	(d)	(e)
7	2.045e-14	7.134e-12	1.829e-10	1.500e-10	1.162e-11
10	1.565e-12	8.546e-12	4.620e-06	4.080e-06	2.445e-11
13	4.523e-09	2.941e-09	2.030e-02	2.261e-02	4.101e-09
16	4.276e-05	7.042e-06	5.922e-01	5.110e-01	2.260e-05
19	4.455e-01	1.119e-01	6.638e-01	7.050e-01	2.843e-01
22	7.614e+03	1.870e+03	7.919e-01	8.282e-01	3.240e+03
25	5.464e+07	7.745e+06	8.012e-01	8.129e-01	3.977e+07

Na poniższych wykresach można zobaczyć jak dane metody sprawdziły sie w interpolacji funkcji $arctg(x)$ oraz $\frac{1}{1+25x^2}$

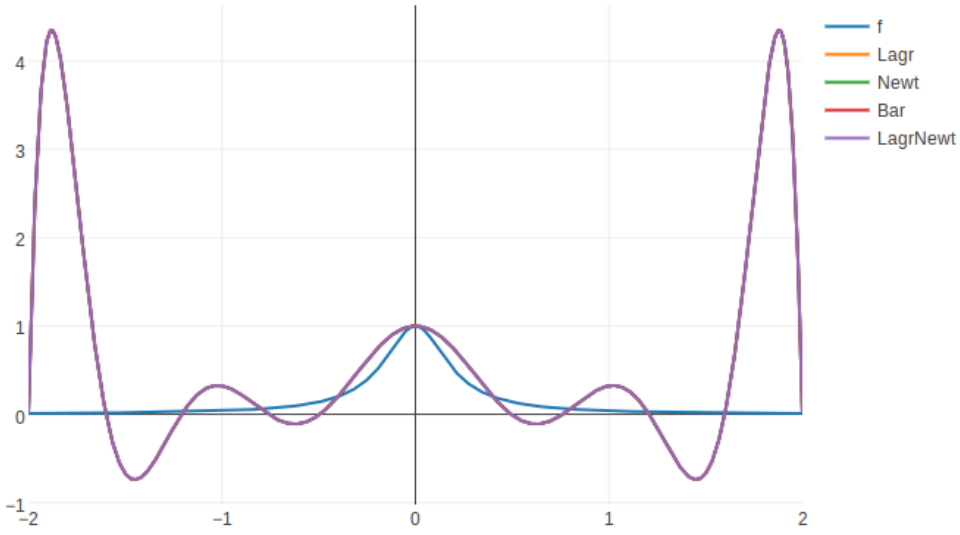
Wykres dla funkcji $arctg(x)$ z węzłami równo odległymi:



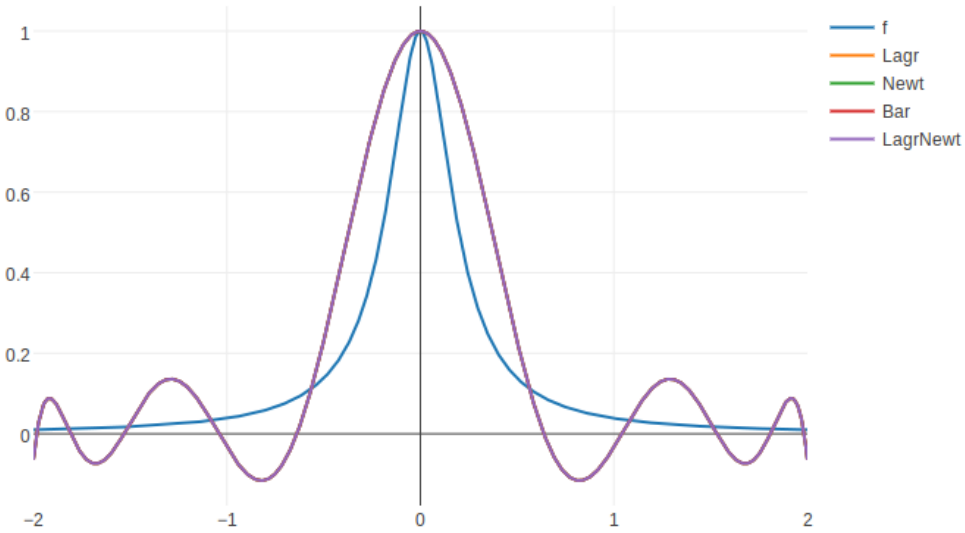
Wykres dla funkcji $arctg(x)$ z węzłami Czebyszewa:



Wykres dla funkcji $\frac{1}{1+25x^2}$ z węzłami równo odległymi:



Wykres dla funkcji $\frac{1}{1+25x^2}$ z węzłami Czebyszewa:



4 Podsumowanie

Ze wszystkich postaci wielomianu interpolacyjnego, najlepszą okazała się postać Barycentryczna, ze względu na swoją stabilność numeryczną, łatwość w obliczaniu jej wartości oraz dodawania nowych węzłów interpolacyjnych.Algorytm wyznaczania jej współczynników co prawda generuje błędy numeryczne które można delikatnie zmniejszyć porządkując punkty x_k, lecz ogólnie rzecz biorąc, bilans zysków i strat jest dodatni.
Algorytm przekształcania wielomianu z postaci Lagrange'a do postaci Newtona daje akceptowalne wyniki.Uzyskane współczynniki są nieco gorsze od tych obliczanych algorytmem ilorazów różnicowych, ale z kolei średni błąd względny jest mniejszy niż ten który otrzymujemy w przypadku obliczania wartości wielomianu w postaci Lagrange'a.
Pozytywnym aspektem jest także to, że dla niewielkiej liczby węzłów interpolacyjnych, wielomiany otrzymane za pomocą powyższych metod nie różnią się prawie wcale, co widać na przedstawionych wykresach - różnica między wielomianami jest niezauważalna.

Literatura

[1] David Kincaid, Ward Cheney, przekł. Stefan Paszkowski, *Analiza Numeryczna*, Warszawa, WNT, 2006.

[2] Wilhelm Werner, *Mathematics of Computation 43 (1984) 205-217*

[3] Jean-Paul Berrut Lloyd N. Trefethen *Barycentric Lagrange Interpolation (2004)*