Pracownia z analizy numerycznej
Sprawozdanie do zadania **P2.8**
Prowadzący pracownię: Paweł Woźny

Łukasz Klasiński

Wrocław, 17 grudnia 2017

1 Wstęp

Problem przybliżania funkcji może być rozwiązany między innymi dzięki użyciu interpolacji wielomianowej. Głównym celem niniejszego sprawozdania będzie testowanie kilku algorytmów do obliczania wielomianu **Lagrang'a**, którego postać wygląda następująco:

Wielomian $L_n \in \Pi_n$, spełniający dla danych parami różnych liczb $x_0, x_1, \dots x_n$ oraz wartości funkcji f w tych punktach, taki że

$$L_n(x_k) = f(x_k), (k = 0, 1, \dots, n)$$

można zapisać w postaci

$$L_n(x) = \sum_{i=0}^{n} \lambda_i f(x_i) \prod_{j=0, j\neq i}^{n} (x - x_j) \tag{1}$$

gdzie

$$\lambda_i = \prod_{j=0, j\neg i}^{n} \frac{1}{(t_i - t_j)} \tag{2}$$

Zbadana zostanie stabilność różnych algorytmów na wybór węzłów oraz jaki generują błąd dla danej ich ilości.

Wszystkie testy numeryczne przeprowadzono przy użyciu języka programowania **Julia v.0.6.1** w trybie podwójnej (**Double**) precyzji, czyli 64 bitowej dokładności.

2 Postać Barycentryczna

Weźmy równanie (1) i oznaczmy l_i jako $\lambda_i * \prod_{j\neq i}^{n} = \prod_{j\neq i}^{n} \frac{x-x_j}{x_i-x_j}$ z (2)
Zauważmy, że licznik l_i może zostać zapisany jako równość

$$l(x) = (x - x_0)(x - x_1)\dots(x - x_n)$$

dzielony przez $x - x_i$. Wtedy $\lambda_i = 1/l'(x_i)$, więc l_i można zapisać jako

$$l_i(x) = l(x)\tfrac{\lambda_i}{(x-x_i)}$$

Zauważmy, że składniki sumy (1) zawierają $l(x)$, który nie zależy od i. Dlatego go wyciągnąć przed sumę, aby otrzymać

$$L_n(x) = l(x)\sum_{i=0}^{n}\frac{\lambda}{x - x_i}f(x_i) \tag{3}$$

Teraz załózmy, że interpolujemy funkcję stałą $= 1$. Wtedy wstawiając to do (3), otrzymamy równość

$$1 = \sum_{i=0}^{n} l_i(x) = l(x)\sum_{i=0}^{n}\frac{\lambda_i}{(x - x_i)}$$

Dzieląc to przez (3), otrzymamy barycentryczną formułę wielomianu (1)

$$L_n(x) = \frac{\sum_{i=0}^{n}\frac{\lambda}{x-x_i}f(x_i)}{\sum_{i=0}^{n}\frac{\lambda}{x-x_i}} \tag{4}$$

dla λ takiej samej jak (2). Dalej, przy testowaniu będziemy go nazywać wielomianem barycentrycznym.
Zauważmy, że dzięki takiej postaci λ_i nie korzysta ze zmiennej x. Dzięki temu dla zadanych węzłów wystarczy raz ją wyliczyć i przy wyznaczaniu wartości wielomianu używać tej stałem bez ponawiania obliczeń. Ponadto jeśli dodamy nowe węzły, to w celu wyliczenia nowej wartości λ_i, wystarczy przemnożyć ją przez odpowiedni iloraz $\frac{1}{t_i-t_j}$.

2.1 Węzły Chebyszewa

Taka postać wielomianu ma również inną własność. Jeśli zaaplikujemy do niego węzły Chebyszewa pierwszego rodzaju:

$$x_i = cos\frac{(2i+1)\pi}{2n+2}, \qquad (i = 0, \dots n) \tag{5}$$

to wzór na poszczególne wartości λ_i z (4) znacząco się upraszcza.
Z własności $\lambda_i = \frac{1}{l'(x)}$, o której była mowa wcześniej, możemy uzyskać ogólną formułę na w_i. Po zaaplikowaniu do niej węzłów Chebyszewa i usunięciu czynników niezależnych od i otrzymamy

$$\lambda_i = (-1)^j sin \frac{(2j+1)\pi}{2n+2} \tag{6}$$

Podobnie można postępować w przypadku węzłów Chebyszewa innego rodzaju.

3 Algorytm Wernera

Przytoczmy wielomian w postaci Newtona. Wyraża się on wzorem

$$P_n(x) = \sum_{i=0}^{n} a_i p_i(x) \tag{7}$$

gdzie p_i jest wielomianem węzłowym

$$p_i(x) = (x - x_0)(x - x_1)\dots(x - x_{i-1})$$

natomiast współczynniki a_i obliczamy za pomocą ilorazu różnicowego

$$a_i = f[x_0, x_1 \dots x_i]$$

Główną zaletą tej formy jest mniejsza ilość operacji do wyliczenia wartości wielomianu w porównaniu do postaci Lagrang'a. Użyjemy tą postać do wyznaczenia algorytmu wyliczającego λ_i w wielominie barycentrucznym z (4). Biorąc (1) oraz podstawiając za f funkcję stałą $f(x) = 1$, otrzymamy

$$1 = \sum_{i=0}^{n} \lambda_i \prod_{j=0, j\neq i}^{n} (x - x_i) \tag{8}$$

Jeśli rozważymy lewą stronę równania (8) jako wielomian w formie Newtona, to mamy

$$P_n(x) = \sum_{i=0}^{n} \lambda_i \prod_{j=0, j\neq i}^{n} (x - x_i)$$

Możemy teraz rozwiązać ten wielomian metodą Newtona. Po przekształceniach tego równania otrzymujemy

$$a_k = \sum_{i=0}^{k} \lambda_i \prod_{j=k+1}^{n} (x_i - x_j), \qquad (k = 0, \dots n)$$

Zatem problem ogranicza się do rozwiązania układów równań:

$$
\begin{pmatrix} a_0 \\ a_1 \\ a_2 \\ \vdots \\ a_{n-1} \\ a_n \end{pmatrix}
\begin{pmatrix}
\prod_{j=1}^{n}(x_0 - x_j) & 0 & 0 & \dots & 0 \\
\prod_{j=2}^{n}(x_0 - x_j) & \prod_{j=2}^{n}(x_1 - x_j) & 0 & \dots & 0 \\
\prod_{j=3}^{n}(x_0 - x_j) & \prod_{j=3}^{n}(x_1 - x_j) & \prod_{j=3}^{n}(x_2 - x_j) & \dots & 0 \\
\vdots & \vdots & \vdots & \vdots & \vdots \\
x_0 - x_n & x_1 - x_n & x_2 - x_n & \dots & 0 \\
1 & 1 & 1 & \dots & 1
\end{pmatrix}
\begin{pmatrix} \lambda_0 \\ \lambda_1 \\ \lambda_2 \\ \vdots \\ \lambda_{n-1} \\ \lambda_n \end{pmatrix}
\tag{9}
$$

Może on być rozwiązany za pomocą metody eliminacji Gauss'a, który zastosujemy w innej kolejności niż zwykle:

1. Podzielmy pierwsze równanie z (9) przez $x_0 - x_1$, oznaczmy

$$a_0^{(1)} := a_0/(x_0 - x_1)$$

2. Odejmijmy pierwsze równianie od drugiego, oznaczmy

$$a_1^{(1)} := a_1 - a_0^{(1)}$$

3. Podzielmy pierwsze równianie przez $x_0 - x_2$, oznaczmy

$$a_0^{(2)} := a_0^{(1)}/(x_0 - x_2)$$

4. Odejmijmy pierwsze równianie od trzeciego, oznaczmy

$$a_2^{(1)} := a_2 - a_0^{(2)}$$

5. Podzielmy drugie równanie przez $x_1 - x_2$, oznaczmy

$$a_1^{(2)} := a_1^{(1)}/(x_1 - x_2)$$

6. Odejmijmy drugie równanie od trzeciego, oznaczmy

$$a_2^{(2)} := a_2^{(1)} - a_1^{(2)}$$

Kontynując w ten sposób otrzymujemy algorytmu

$$
\begin{aligned}
&a_k^{(0)} := a_k, \qquad (k = 0, \dots, n) \\
&\left.\begin{aligned} a_k^{i} &:= a_k^{(i-1)}/(x_k - x_i) \\ a_i^{(k+1)} &:= a_i^{(k)} - a_k^{(i)} \end{aligned}\right\} \qquad k = (0, \dots, i-1), i = (1, \dots, n) \\
&\lambda_i := a_i^{(n)}, \qquad (i = 0, \dots, n)
\end{aligned}
\tag{10}
$$

Po zastosowaniu go do (8) dostajemy algorytm na wydajne obliczanie wartości λ_i z (4)

$$\begin{aligned}
&a_0^{(0)} := 0, a_k^{(0)} := 0, \qquad (k = 1, \dots, n)\\
&\left.\begin{aligned} a_k^i &:= a_k^{(i-1)}/(x_k - x_i) \\ a_i^{(k+1)} &:= a_i^{(k)} - a_k^{(i)} \end{aligned}\right\} \qquad k = (0, \dots, i-1), i = (1, \dots, n)\\
&\lambda_i := a_i^{(n)}, \qquad (i = 0, \dots, n)
\end{aligned} \tag{11}$$

Zatem otrzymaliśmy algorytm Wernera, który pozwala na szybsze obliczenie wagi barycentrycznej niż w przypadku standardowego wyrażenia. Dzięki temu wykonujemy mniejsza operacji i otrzymujemy mniejszy błąd przez utratę cyfr znaczących.

4 Testy

5 Podsumowanie

Literatura

[1] C. Schneider and W. Werner, Some New Aspects of Rational Interpolation 1986

[2] J. Berrut and L. N. Trefethen, Baricentric Lagrange Interpolation 2004

[3] W. Werner, Polynomial Interpolation: Lagrange versus Newton 1984