Mikołaj Korobczak

Pracownia z analizy numerycznej
Pracownia P2 Zadanie 8

Wrocław, 4 kwietnia 2020

Prowadzący: dr hab. Paweł Woźny

Spis treści

1. Wstęp

Aproksymacja to przybliżenie jakiejść niewygodnej lub skomplikowanej funkcji f przy pomocy łatwiejszej funkcji g, gdzie łatwa funkcja to zazwycznaj wielomian i taką aproksymację będziemy tutaj omawiać. Wielomian W dobrze aproksymuje funkcję f na przedziale $\langle a, b\rangle$ gdy funkcja błędu

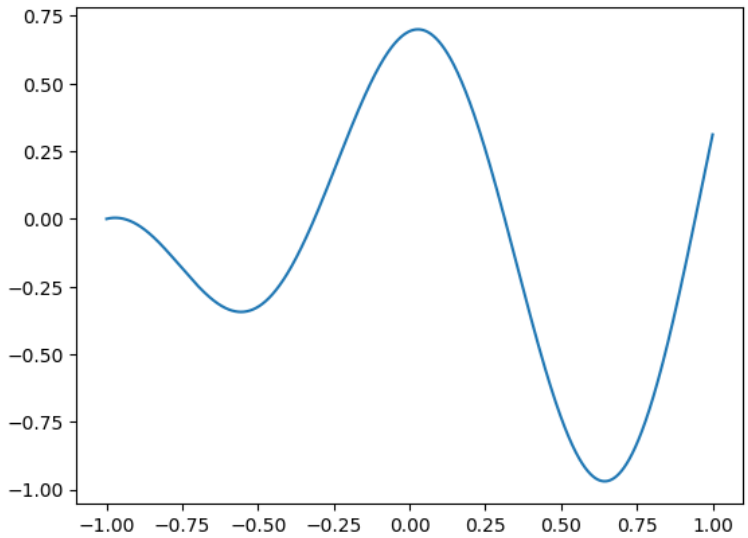
$$||f - w||_\infty = \max_{a \leqslant x \leqslant b} |f(x) - w(x)| \tag{1}$$

jest mała. Natomiast n-tym wielomianem optymalnym W_n nazywamy wielomian, który spełnia równanie

$$\inf_{w \in \prod_n} ||f - w||_\infty = ||f - W_n||_\infty. \tag{2}$$

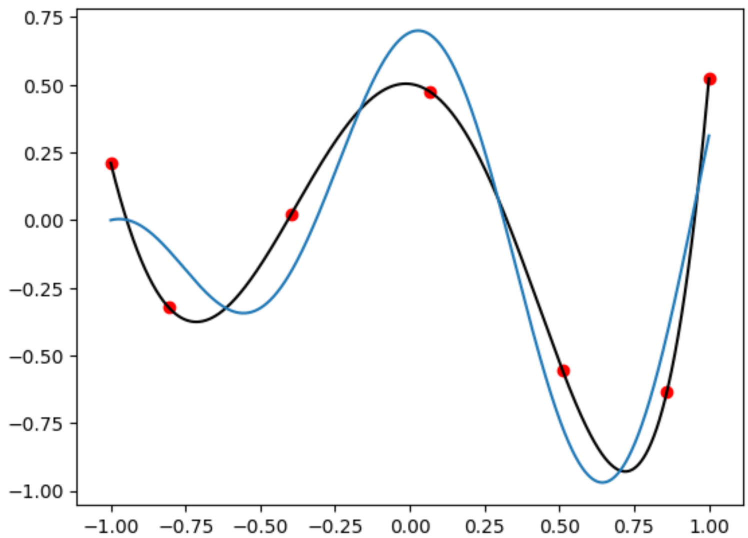
Wiemy, że istnieje tylko jeden taki wielomian dla każdej funkcji f i każdego n.[1] Znalezienie takiego wielomianu okazuje się jednak trudnym zadaniem, ponieważ nie da się podać jawnego wzoru na wyznaczenie W_n, a algorytm, który go wyznacza (algorytm Remeza) działa długo i jest skomplikowany. Dlatego szukamy prostsztych wielomianów, które łatwo policzyć, a które dają wyniki zbliżone do tych, które daje W_n. Postanowiłem przetestować skuteczność aproksymacji trzech takich wielomianów I_n, J_n i K_n. W kolejnych sekcjach będę omawiał te wielomiany i pokazywał ich aproksymacje na przykładowej funkcji

$$f(x) = \log(x + 2) * \cos(5x). \tag{3}$$



Rysunek 1. Wykres funkcji f

Aby unaocznić zadanie aproksymacji pokazuję wykres wielomianu W_5 dla wskazanej funkcji. Czerwone punkty to punkty alternansu.



Rysunek 2. Aproksymacja wielomianem optymalnym W

2. Wielomian interpolacyjny I_n

Wielomian I_n jest wielomianem interpolacyjnym stopnia n, który interpoluje zadaną funkcję f w punktach $t_{n,k}$ zadanych wzorem

$$t_{n,k} = \cos \frac{(2k-1)\pi}{2n}, \tag{4}$$

które są zerami n-tego wielomiany Czebyszewa. Wielomian ten bierzemy nie bez powodu, ponieważ spośród wszystkich wielomianów interpolacyjnych chcielibyśmy znaleźć taki, żeby jego norma była najmniejsza, a wiemy z [2], że

$$||f - I_n||_\infty \leqslant \frac{1}{(n+1)!}||Z||_\infty||f^{(n+1)}||_\infty \tag{5}$$

gdzie Z jest wielomianem $(n+1)$-tego stopnia. Tak więc norma ta zależy od normy wielomianu Z, a najmniejszą normę równą 2^{-n} na przedziale $\langle -1, 1\rangle$ ma wielomian o zerach w zerach wielomanu Czebyszewa T_{n+1} [3]. Wielomian interpolujący więc funkcję w zerach Czebyszewa powinien być blisko wielomianu optymalnego. Wielomian ten zadany jest dwoma równoważnymi wzorami[1]

$$I_n(x) = \frac{1}{n+1}T_{n+1}(x)\sum_{k=1}^{n+1}(-1)^{k-1}f(t_{n+1,k})\Big(\sin\frac{(2k-1)\pi}{2(n+1)}\Big)/(x - t_{n+1,k}) \tag{6}$$

$$I_n(x) = \frac{2}{n+1}\sum_{j=0}^{n}{}'\Big(\sum_{k=1}^{n+1} f(t_{n+1,k})T_j(t_{n+1,k})\Big)T_j(x). \tag{7}$$

Dowód. [4] Aby pokazać, że I_n jest wielomianem interpolującym w punktach (4) rozważmy współczynnik przy $f(t_{n+1,k})$ w (6)

$$Q_k(x) = \frac{1}{n+1}T_{n+1}(x)(-1)^{k-1}\sin\frac{(2k-1)\pi}{2(n+1)}/(x - t_{n+1,k}) \tag{8}$$

Widać, że jest to wielomian, który ma wszystkie zera wielomianu T_{n+1} oprócz $t_{n+1,k}$. Jest to więc wielomian co najwyżej n-tego stopnia. Jednocześnie wielomian ten w punktach $t_{n+1,j}$ $j \neq k$ ma wartość 0. Możemy też zapisać, że

$$Q_n(t_{n+1,k}) = \frac{1}{n+1}(-1)^{k+1}\Big(\sin\frac{(2k-1)\pi}{2(n+1)}\Big)T'_{n_1}(t_{n+1,k}), \tag{9}$$

a ponieważ

$$T'(t_{n,k}) = (-1)^{k-1}n\Big(\sin\frac{(2k-1)\pi}{2n}\Big)^{-1} \tag{10}$$

to $Q_k(t_{n+1,k}) = 1$, więc (6) spełnia warunki interpolacji. Trzeba jeszcze pokazać równoważność dwóch wzorów na wielomian I_n. Skorzystamy do tego z równoważności

$$2(x - t_{n+1,k})\sum_{j=0}^{n}{}'T_j(t_{n+1,k})T_j(x) = T_{n+1}(x)T_n(t_{n+1,k}) \tag{11}$$

gdzie po rozpisaniu $T_n(t_{n+1,k})$ na

$$T_n(t_{n+1,k}) = (-1)^{k-1}\sin\frac{(2k-1)\pi}{2(n+1)} \tag{12}$$

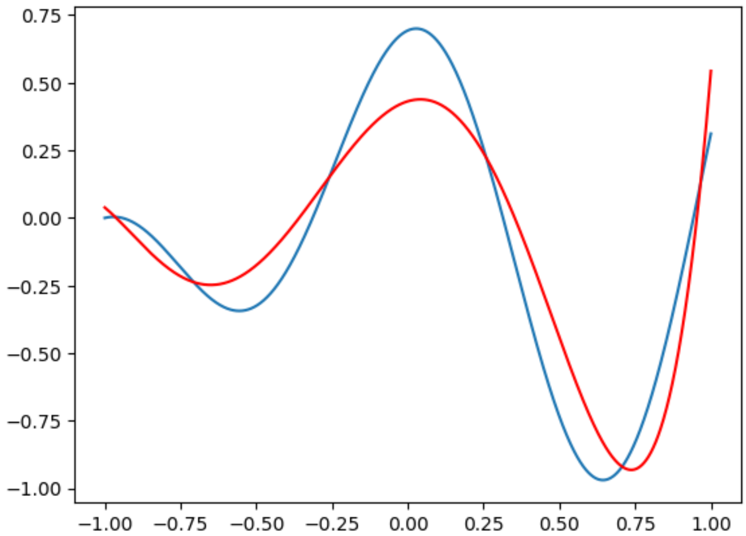
dostajemy, że

$$(-1)^{k-1}\Big(\sin\frac{(2k-1)\pi}{2(n+1)}\Big)T_{n+1}(x)/(x - t_{n+1,k}) = 2\sum_{j=0}^{n}{}'T_j(t_{n+1,k})T_j(x), \tag{13}$$

więc oba wzory są równoważne. □

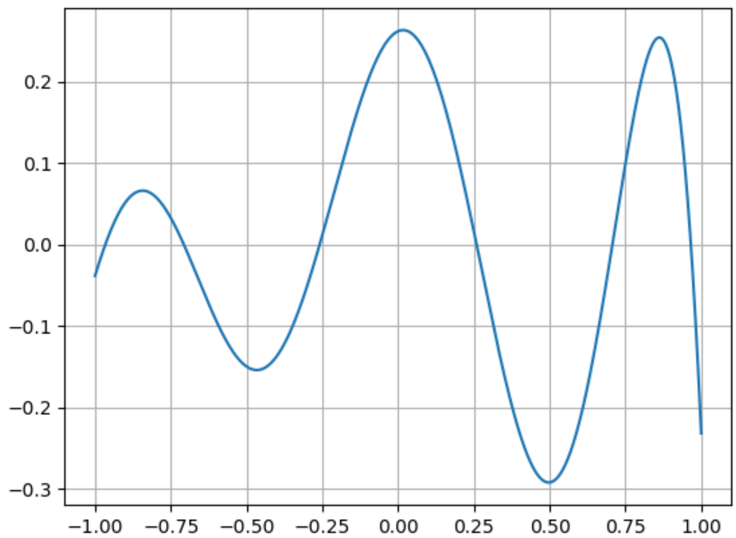
[1] Wzór na wielomian I_n wziąłem z książki Paszkowskiego i tego się trzymałem pomimo drobnej różnicy względem wzoru podanego na wykładzie.

Postanowiłem przetestować możliwości tego wielomianu dla różnych przykładów. Jednym z nich może być poniższy, w którym aproksymuję funkcję (3) naszym wielomianem:



Rysunek 3. Aproksymacja wielomianem I

Wydaje się, że wykres naszego wielomianu zachowuje się tak, jak byśmy chcieli, czyli jest podobny do wykresu wielomianu optymalnego. Wykres błędu naszego wielomianu wygląda tak:



Rysunek 4. Wykres błędu aproksymacji wielomianem I

Po wykresie widać jednak, że nie można tu pokazać alternansu, więc z twierdzenia Czebyszewa o alternansie nie może to być wielomian optymalny, co dodatkowo widać, gdy porówna się maksymalny błąd. Błąd naszego wielomianu I_5 wynosi 0.29226, podczas gdy błąd wielomiany optymalnego wynosił 0.21101. Przeprowadziłem jeszcze inne testy dla różnych funkcji, których wyniki przedstawiłem w rozdziale 6.

3. Wielomian interpolacyjny J_n

Innym podejściem do aproksymacji może być próba interpolacji funkcji w ekstremach Czebyszewa, czyli w zerach wielomianów Czebyszewa drugiego rodzaju, zadanych wzorem

$$u_{n-1,k} = \cos\frac{k\pi}{n} \tag{14}$$

Okazuje się, że dla niektórych funkcji aproksymowanych wielomian J_n interpolujący w tych punktach zachowuje się lepiej niż wielomian I_n. Wielomian J_n wyraża się wzorem

$$J_n(x) = \frac{2}{n}\sum_{j=0}^{n}{}''\Big(\sum_{k=0}^{n}{}'' f(u_{n-1,k})T_j(u_{n-1,k})\Big)T_j(x). \tag{15}$$

Dowód. [5] Przekształćmy wzór na J_n do postaci

$$J_n(x) = \frac{2}{n}\sum_{j=0}^{n}{}''\Big(\sum_{k=0}^{n}{}'' T_j(x)T_j(u_{n-1,k})\Big)f(u_{n-1,k}). \tag{16}$$

wiemy, że jeśli zdefiniujemy iloczyn skalarny dla wielomianów wzorem

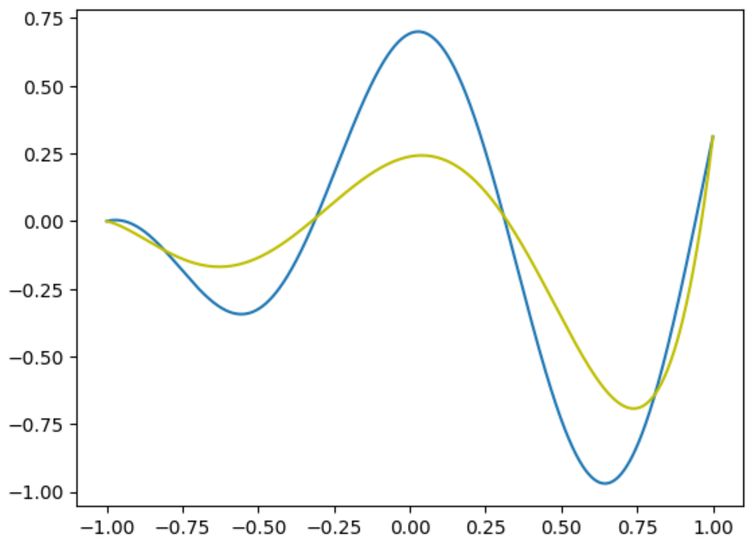
$$\langle f, g\rangle = \sum_{k=0}^{n}{}'' f(x_k)g(x_k), \tag{17}$$

to

$$\langle T_k, T_l\rangle = \begin{cases} 0 & (k \neq l) \\ n & (k = l = 0 \text{ lub } k = l = n) \\ \frac{n}{2} & (0 < k = l < n), \end{cases} \tag{18}$$

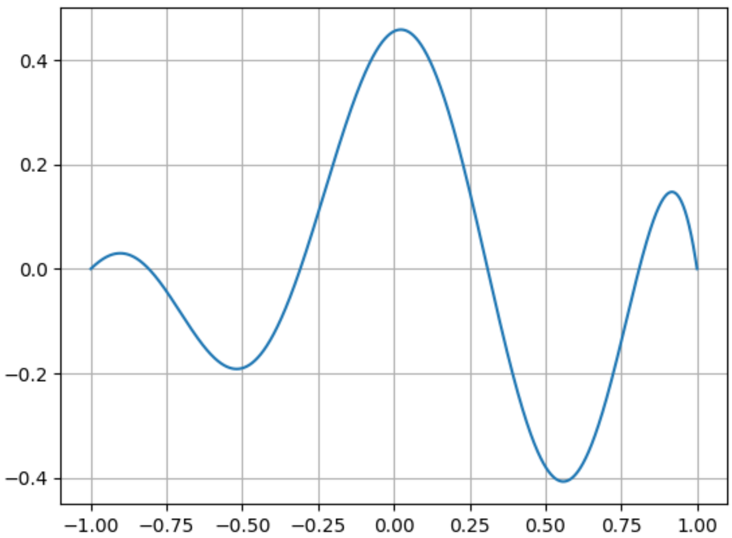
więc biorąc pod uwagę dzielenie pierwszego i ostatniego wyrazu sumy przez 2 w zewnętrznej sumie widzimy, że $f(u_{n-1,k}) = J_n(u_{n-1,k})$. $\square$

Wróćmy znowu do naszej funkcji (3). Wykres aproksymacji prezentuje się następująco:



Rysunek 5. Aproksymacja wielomianem J

natomiast wykres błedu wygląda tak:



Rysunek 6. Błąd aproksymacji wielomianem J

Wygląda na to, że nasz wielomian J_5 aproksymuje znacznie gorzej niż wielomian I_5 i widać to już po samym wykresie, a wykres błędu tylko potwierdza nasze podejrzenia. Natomiast po jednym przykładzie nie można jeszcze tak od razu stwierdzić, że J_n jest gorszy. Dlatego zrobiłem więcej testów, a ich wyniki przedstawiłem w rozdziale 6.

4. Wielomian K_n

Ostatnim badanym przeze mnie wielomianem był wielomian K_n. Wiemy, że aby wielomian W_n był n-tym wielomianem optymalnym w sensie aproksymacji jednostajnej, musi posiadać $(n+2)$ punktowy alternans, w którym wartość błędu na zmianę przyjmuje maksymalną wartość błędu. Wielomian K_n jest tak zadany, że gdyby dla wielomianu W_n alternans wypadał by na zbiorze

$$\{u_{n,0}, u_{n,1}, \ldots, u_{n,n+1}\} \tag{19}$$

to $W_n = K_n$. Wzór na wielomian K_n, to

$$K_n(x) = \frac{1}{n+1} \sum_{k=0}^{n+1}{}'' (-1)^k f(u_{n,k})((x^2-1)U_n(x)/(x-u_{n,k}) - T_{n+1}(x)) \tag{20}$$

oraz

$$K_n(x) = \frac{2}{n+1} \sum_{j=0}^{n+1}{}' \Big(\sum_{k=0}^{n+1}{}'' f(u_{n,k}) T_j(u_{n,k}) \Big) T_j(x), \tag{21}$$

a n-ty bład aproksymacji na zbiorze (19) wynosi

$$\epsilon = \frac{1}{n+1} \Big| \sum_{k=0}^{n+1}{}'' (-1)^k f(u_{n,k}) \Big|. \tag{22}$$

Dowód. [6] Innym wzorem na J_n z poprzedniego rozdziału jest wzór

$$J_{n+1}(x) = \frac{1}{n+1}(x^2-1)U_n(x) \sum_{k=0}^{n+1}{}'' 9-1)^k f(u_{n,k})/(x-u_{n,k}). \tag{23}$$

Współczynnik stojący przy x^{n+1} w J_{n+1} wynosi

$$\frac{2^n}{n+1}\sum_{k=0}^{n+1}{}''(-1)^k f(u_{n,k}). \tag{24}$$

Natomiast ponieważ $T_{n+1}(x) = 2^n x^{n+1} + \dots$, więc różnica

$$J_{n+1} - \Big(\frac{1}{n+1}\sum_{k=0}^{n+1}{}''(-1)^k f(u_{n,k})\Big)T_{n+1} \tag{25}$$

jest wielomianem stopnia co najwyżej n-tego i jest to przy okazji nasz wielomian K_n. Jednocześnie widać, że w punktach $u_{n,k}$ przyjmuje on wartość

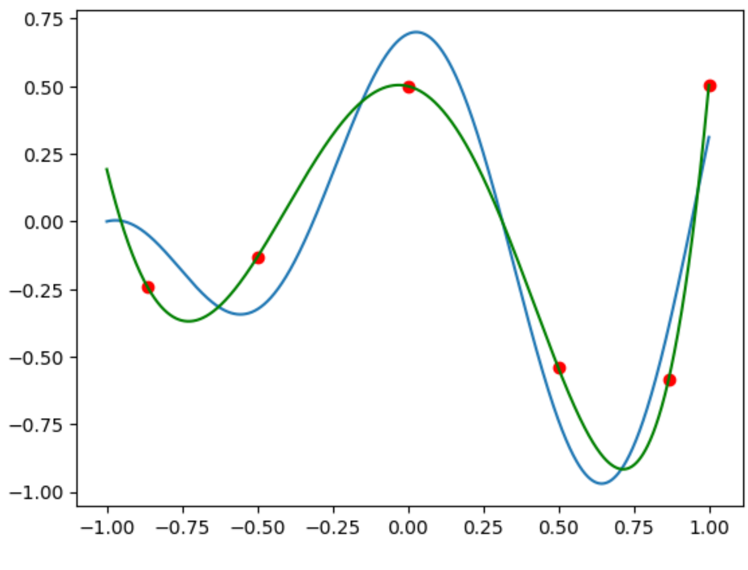
$$f(u_{n,k} - \frac{(-1)^j}{n+1}\sum_{k=0}^{n+1}{}''(-1)^k f(u_{n,k}) \tag{26}$$

czyli posiada $n+2$ punktowy zbiór na którym przyjmuje na zmianę wartość błędu ϵ. Wzór (21) wynika bezpośrednio z określenia K_n jako

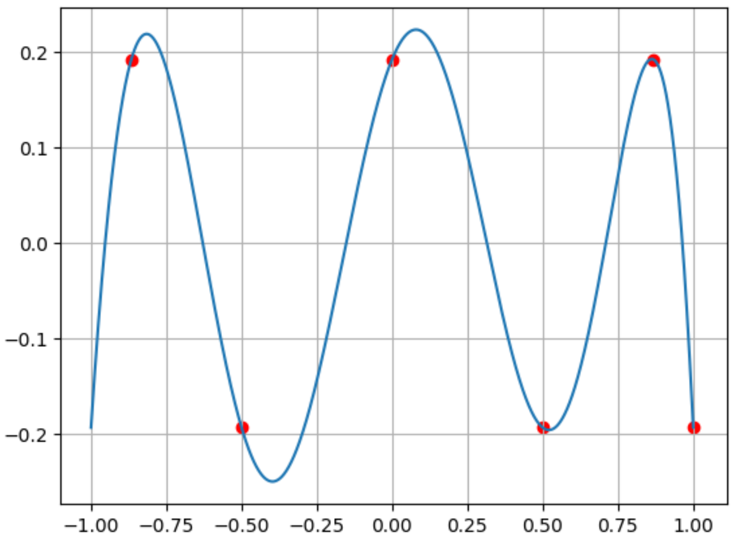
$$\begin{aligned}
K_n &= J_{n+1} - \epsilon T_{n+1} = \\
&= \frac{2}{n+1}\sum_{j=0}^{n+1}{}''\Big(\sum_{k=0}^{n+1}{}'' f(u_{n,k})T_j(u_{n,k})\Big)T_j - \\
&- \frac{1}{n+1}\sum_{k=0}^{n+1}{}'' f(u_{n,k})T_{n+1}(u_{n,k})T_{n+1} = \\
&= \frac{2}{n+1}\sum_{j=0}^{n}{}'\Big(\sum_{k=0}^{n+1}{}'' f(u_{n,k})T_j(u_{n,k})\Big)T_j.
\end{aligned}$$

□

Ponownie przetestujmy aproksymację funkcji (3). Tym razem na wykresach na czerwono zaznaczam punkty (19), które tworzą alternans naszego wielomianu.



Rysunek 7. Aproksymacja wielomianem K



Rysunek 8. Błąd aproksymacji wielomianem K_5.

Nasz wielomian K_5 wygląda już niemal tak samo, jak wielomian optymalny W_5, a wykres błędu pokazuje, że jesteśmy już bardzo blisko tego, aby wielomian ten był wielomianem optymalnym. Widać to również porównując błąd maksymalny, który wynosi 0.24930 do błedu wielomianu optymalnego W_5, a przypomnijmy, że wynosi on 0.21100. Wydaje się zatem, że jest to bardzo wydajny sposób aproksymacji. Tak jak poprzednie wielomiany przetestowałem ten wielomian dla różnych funkcji, a wyniki przedstawiłem w rozdziale 6.

5. Algorytm Clenshowa

Do liczenia wartości wielomianów postaci

$$w(x) = \sum_{k=0}^{n}{}' a_k T_k(x).$$

można użyć algorytmu Clensowa [7]

```
b[n+2] := b[n+1] := 0,
FOR k = n, n-1, ... , 1
    b[k] := a[k] - b[k+2] + 2 * b[k+1] * x
END
RETURN 1/2 * a[0] - b[2] + b[1] * x
```

Używałem go do liczenia wartości wielomianów I_n, J_n, K_n. Aby móc użyć tego kilka razy korzystałem z faktu

$$T_j(u_{n,k}) = T_k(u_{n,j}) \tag{27}$$

na zbiorze $\langle -1, 1\rangle$.

Dowód.

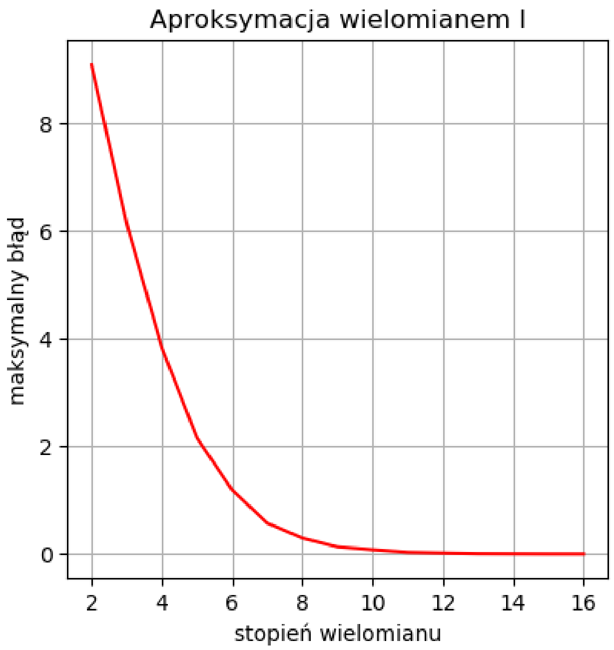
$$T_j(u_{n,k}) = \cos(n \arccos(u_{n,k})) = \cos(n \arccos(\cos\left(\frac{k\pi}{n+1}\right))) =$$
$$= \cos\left(\frac{nk\pi}{n+1}\right) = \cos(k \arccos(u_{n,j})) = T_k(u_{n,j})$$

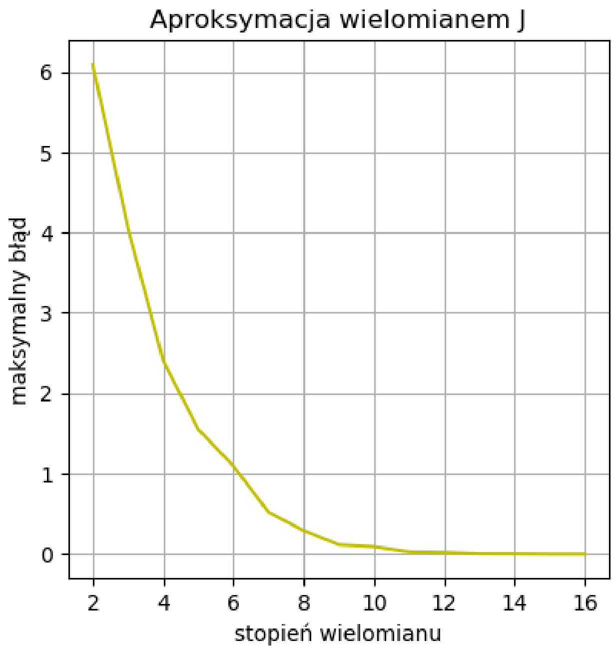
□

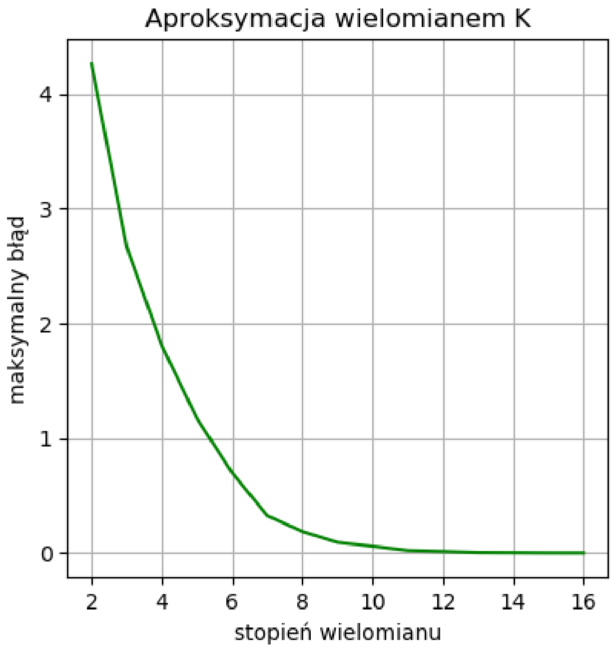
6. Wyniki testów

Testy, które przeprowadziłem były zautomatyzowane. Wszystkie funkcje były generowane losowo (wielomiany i inne funkcje miały losowe wartości z zadanych przedziałów). Przeprowadziłem testy dla wielomianów, funkcji trygonometrycznych postaci $a \sin x + / * b \cos x$, oraz funkcje mieszane, w których używane były takie funkcje jak $a \log(x+3)$, funkcje trygonometryczne i wielomiany. Zależało mi, aby wszystkie funkcje wygenerowane automatycznie były ciągłe na $\langle -1, 1 \rangle$, dlatego nie ma tu funkcji homograficznych, ani wykładniczych. Aproksymowałem te funkcje wielomianami od stopnia $n = 2, 3, \ldots, 100$ w przypadku wielomianów (zatrzymałem testy, w momencie kiedy wartości błędów były bliskie 0) oraz $n = 2, 3, \ldots, 10$ dla pozostałych funkcji. Na wykresach przedstawiłem moje wyniki:

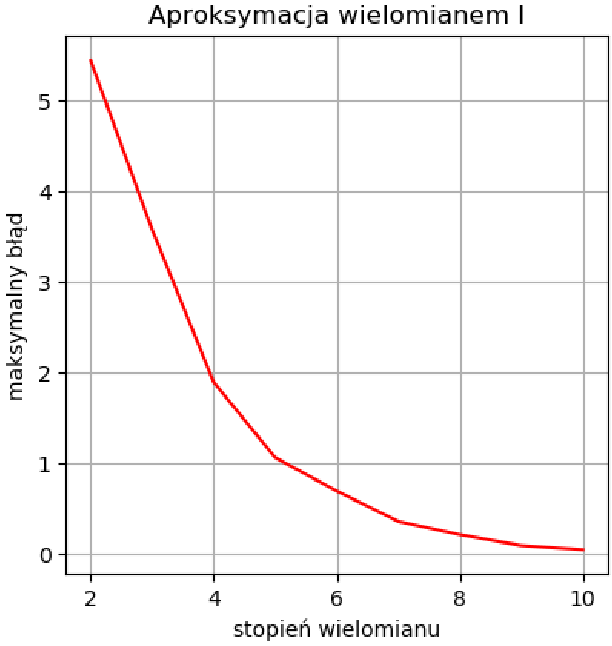
- Dla wielomianów

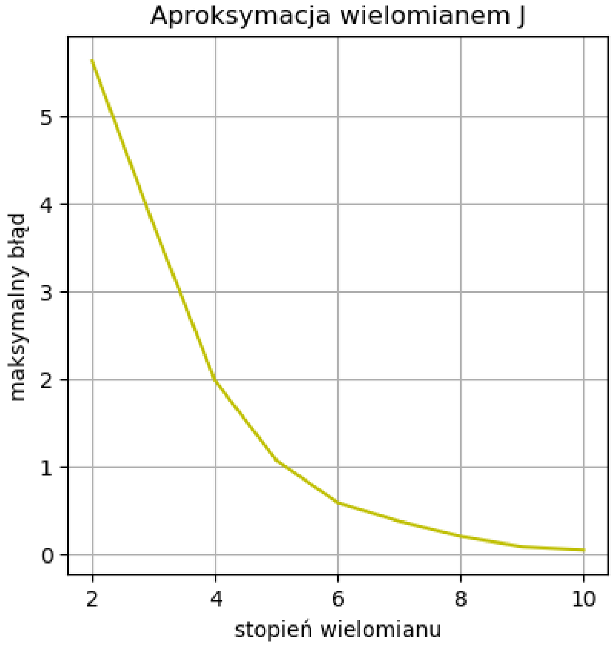


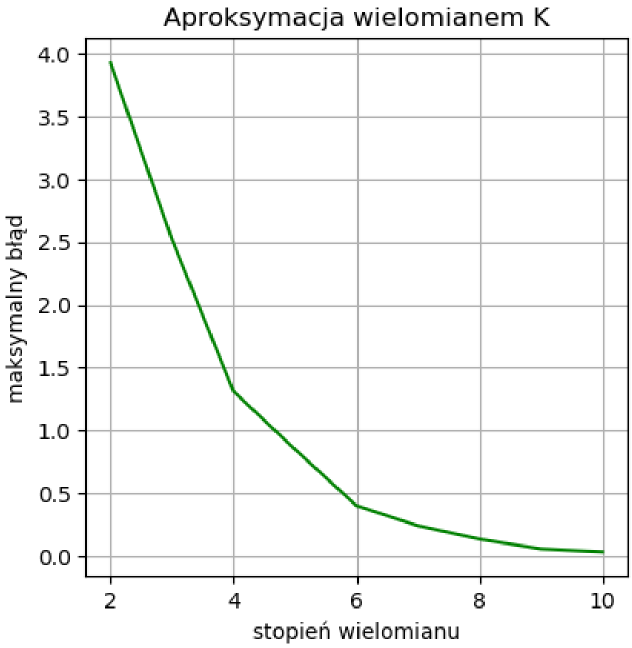




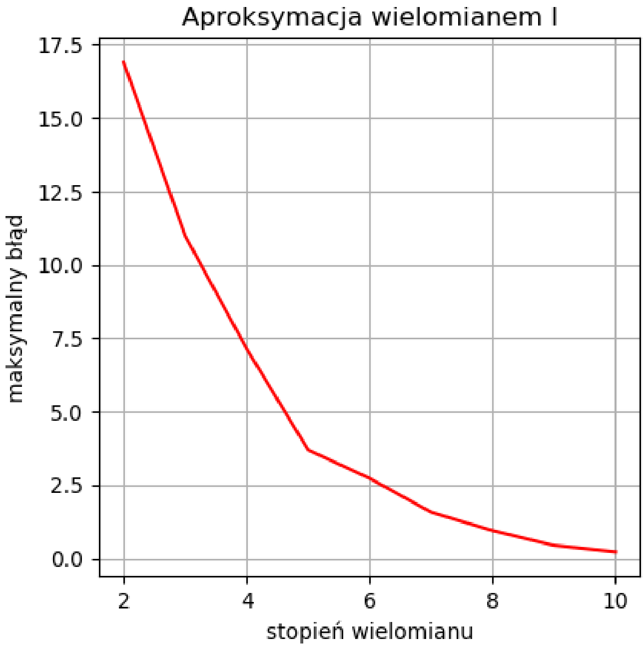
• Dla funkcji trugonometrycznych

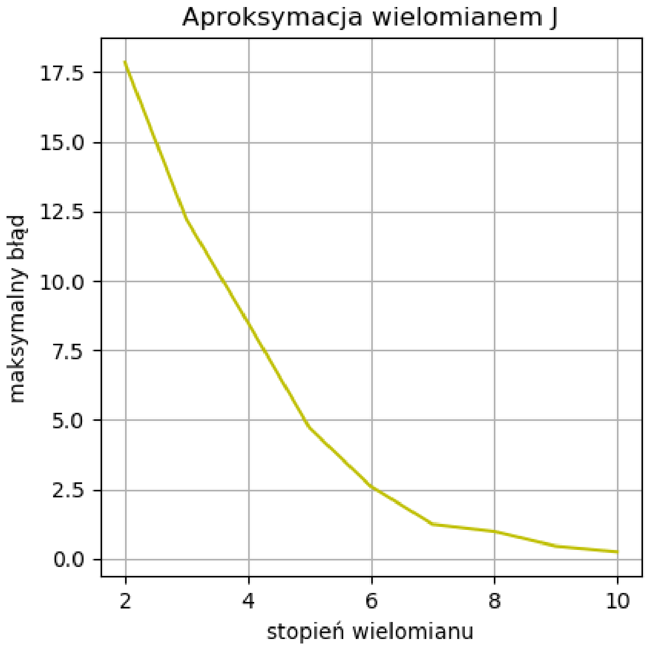


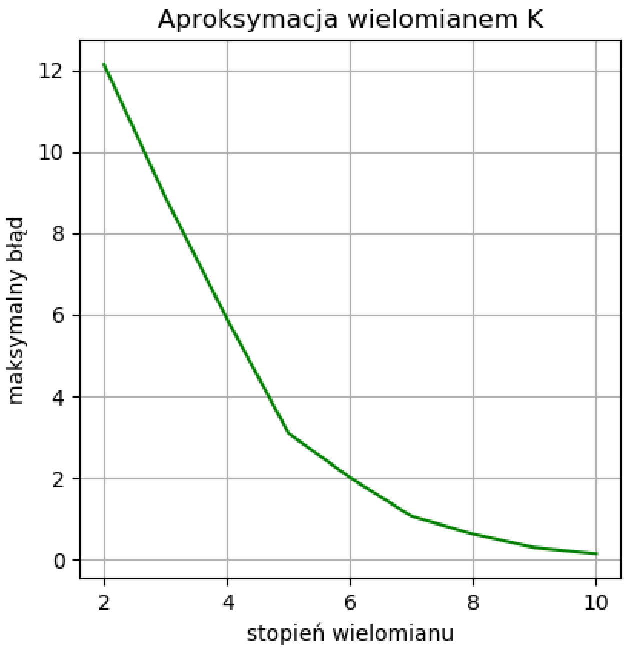




- Dla funkcji mieszanych







7. Podsumowanie i wnioski

Nietrudno było zauważyć, że wielomian K_n zdecydowanie lepiej się zachowywał niż wielomiany I_n i J_n. Wykres błędu tego wielomianu często przypomina taki, który pownien posiadać wielomian optymalny, jego błąd bardzo szybko maleje wraz ze wzrostem wartości n. Dzięki algorytmowi Clenshowa policzenie jego wartości okazuje się łatwym zadaniem, co jest kolejnym argumentem przemawiającym na korzyść tego wielomianu.
Jeśli zaś chodzi o wielomiany interpolacyjne, ich zachowanie przez cały czas było dość podobne. Widać, że wartości błędu maleją z podobną predkością wraz ze wzrostem wartości n, a jeśli chodzi o błędy to na wykresach widać, że są dość zbliżone i chociaż czasami (jak w przypadku wielomianów) wygląda na to, że wielomian I_n jest gorszy, to dla innych funkcji (innych ziaren w losowej generacji) można było

zaobserwować, że jest na odwrót. Podobnie z innymi wykresami. Należy więc przyjąć, że wielomiany te dają bardzo zbliżone wartości błędów. Na korzyść wielomianu J_n przemawia jednak jego prostota liczenia dzięki algorytmowi Clenshowa. We wzorze na wielomian I_n mamy sumę postaci

$$\sum_{k=1}^{n+1} f(t_{n+1,k})T_j(t_{n+1,k}),$$

którego nie można policzyć tym algorytmem, co oznacza, że liczenie wartości I_n jest znacznie trudniejsze. Dlatego, ponieważ zachowują się one tak podobnie, lepiej jest chyba używać wielomianu J_n jeżeli oczywiście nie chcemy używać wielomianu K_n.

Literatura

[1] Na podstawie Stefan Paszkowski „Zastosowania numeryczne wielomianów i szeregów Czebyszewa" wyd. 1975 wstęp do rozdziału 6.

[2] Stefan Paszkowski „Zastosowania numeryczne wielomianów i szeregów Czebyszewa" wyd. 1975 twierdzenie T7.8.

[3] Stefan Paszkowski „Zastosowania numeryczne wielomianów i szeregów Czebyszewa" wyd. 1975 twierdzenie T6.3.

[4] Twierdzenie i dowód Stefan Paszkowski „Zastosowania numeryczne wielomianów i szeregów Czebyszewa" wyd. 1975 T7.9.

[5] Twierdzenie i dowód Stefan Paszkowski „Zastosowania numeryczne wielomianów i szeregów Czebyszewa" wyd. 1975 T7.11.

[6] Twierdzenie i dowód Stefan Paszkowski „Zastosowania numeryczne wielomianów i szeregów Czebyszewa" wyd. 1975 T7.13.

[7] Stefan Paszkowski „Zastosowania numeryczne wielomianów i szeregów Czebyszewa" wyd. 1975 algorytm A14.4, implementacja 19.7.