Interpolacja funkcji sposobem Hermite’a.

Wiktor Gut, 411 761

Środowisko

Program wykonany w systemie Windows x64, na procesorze AMD Ryzen 5 3500U with Radeon Vega Mobile Gfx, za pomocą programu Pycharm w języku Python.

Treść zadania:

Dla funkcji

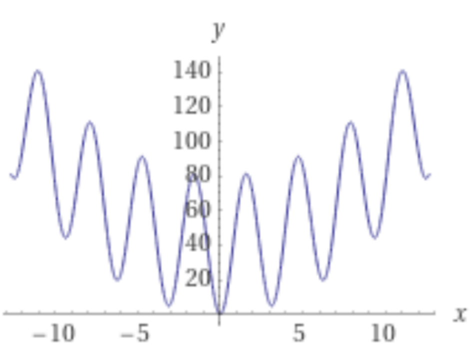
f(x) = 40 + x2/2 - 40*cos(2*x)   na przedziale [-4*pi, 4*pi]**

wyznacz dla zagadnienia ~~Lagrange’a~~ Hermite’a wielomian interpolujący ~~w postaci Lagrange’a i Newtona~~. Interpolację przeprowadź dla różnej liczby węzłów (np. n = 3, 4, 5, 7, 10, 15, 20). Dla każdego przypadku interpolacji porównaj wyniki otrzymane dla różnego rozmieszczenia węzłów: równoodległe oraz Czebyszewa* . Oceń dokładność, z jaką wielomian przybliża zadaną funkcję. Poszukaj wielomianu, który najlepiej przybliża zadaną funkcję. Wyszukaj stopień wielomianu, dla którego można zauważyć efekt Rungego (dla równomiernego rozmieszczenia węzłów). Porównaj z wyznaczonym wielomianem dla węzłów Czebyszewa.

Tekst przekreślony został celowo celem pokazania podobieństwa zadania do poprzedniego tematu. Wielomian interpolacyjny wyznaczamy sposobem Hermite’a jednoznacznie, w przeciwieństwie do zagadnienia Lagrange’a.

Funkcja rysująca wykres a jego prawidłowy wygląd

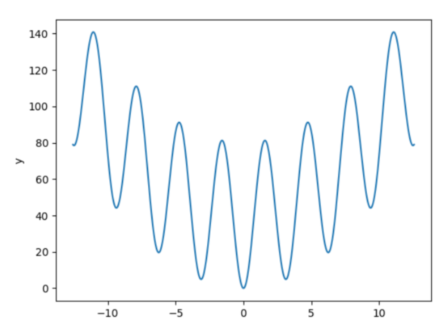
Na potrzeby zadania funkcja została przepisana na język python a następnie narysowana przez zaimplementowaną funkcję rysującą wykres według zadanych n punktów, tutaj dla n=1000. Obok został przedstawiony obraz wyglądu funkcji według programu WolphramAlpha.



<- Zdjęcie 1

Zdjęcie 2 ->

Widać więc, że funkcja została zapisana dobrze, a program rysuje wykres w dokładny sposób.



Implementacja funkcji liczącej wielomian metodą Hermite'a opartej na wzorze Newtona.

Zaimplementowane funkcje zostały zaczerpnięte z wykładu i przełożone na język python bez zewnętrznej pomocy, korzystając m.in. ze Wzorów 1 i 2:

$$H_n(x) = \sum_{l=0}^{n} b_l \cdot p_l(x) = \sum_{i=0}^{k} \sum_{j=0}^{m_{i-1}} b_{(s(i)+j)} \cdot p_{(s(i)+j)}(x)$$

Wzór 1.

$$\begin{array}{llllll}
x_0 & f(x_0) & & & & \\
x_1 & f(x_1) & f[x_0, x_1] & & & \\
x_1 & f(x_1) & f'(x_1) & f[x_0, x_1, x_1] & & \\
x_1 & f(x_1) & f'(x_1) & \frac{f''(x_1)}{2!} & f[x_0, x_1, x_1, x_1] & \\
\dots & \dots & \dots & \dots & \dots & \dots \\
x_n & f(x_n) & f[x_{n-1}, x_n] & \dots & \dots & f[x_0, \dots, x_n]
\end{array}$$

Wzór 2.

W kodzie można zauważyć delikatne podobieństwa do sposobu rozwiązania z metody Newtona; tam też posługiwaliśmy się macierzą pomocniczą do wyliczania współczynników, tutaj możemy to zrobić dokładniej, ponieważ bierzemy pod uwagę również wartość funkcji pochodnej.

Węzły w liczeniu funkcji

Wyżej wymieniona funkcja licząca wielomiany interpolujące została uruchomiona dla dwóch zbiorów węzłów: rozmieszczonych równomiernie na zadanym przedziale oraz wyznaczonych zerami wielomianu Czebyszewa. Drugi z nich posiada znacznie gęściej rozsiane punkty na końcach przedziału, co teoretycznie ma wpływać na zniwelowanie efektu Rungego, czego działanie zweryfikujemy.

Zanim jednak efekt Rungego, przyjrzyjmy się dokładności funkcji interpolowanych dla liczby węzłów z przedziału [5, 50], liczonej zgodnie z błędem maksymalnym spośród 300 równomiernie rozłożonych punktów w przedziale.

Porównanie wpływu rozkładu na interpolację

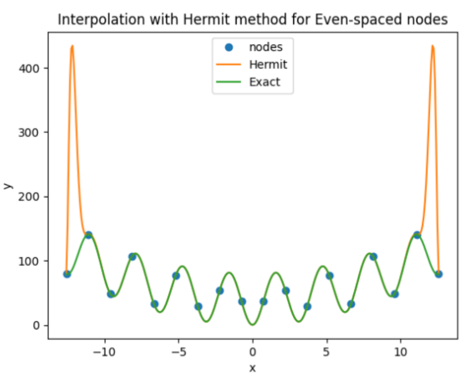
liczba węzłów	stopień Wielomianu	Błąd maksymalny węzły równo rozłożone	Błąd maksymalny Chebyshev
11	21	5,7 * 10^9	7,1 * 10^5
12	23	8,1 * 10^9	3* 10^5
13	25	5,6 * 10^9	6,7 * 10^4
14	27	2,3 * 10^9	8,6 * 10^3
15	29	5,9 * 10 ^8	7 * 10^2
16	31	1 * 10*8	3,8 * 10
17	33	1,3 * 10^7	1,43
18	35	1,3 * 10^6	4,01*10^-2
19	37	9,2 * 10^4	**4,4 * 10^-3**
20	39	5,2 * 10^3	2,27 * 10^-2
21	41	2,3 * 10^2	4,4 * 10^-2
22	43	8,83	2,73 * 10^-1
23	45	3,44 * 10^-1	4,94 * 10^-1
24	47	2,69 * 10^-1	3,43
25	49	**2,48 * 10^-1**	9,36
26	51	1,79	1,5 * 10
27	53	2,32	3,9 * 10
28	55	8,50 * 10^-1	9,9 * 10
29	57	4,15	7,4 * 10^2
30	59	6,19	2,9 * 10^3
31	61	2 * 10	2,5 * 10^3
32	63	5,1 * 10	3,5 * 10^3
33	65	1,6 * 10^2	1,5 * 10^4

Tabela 1

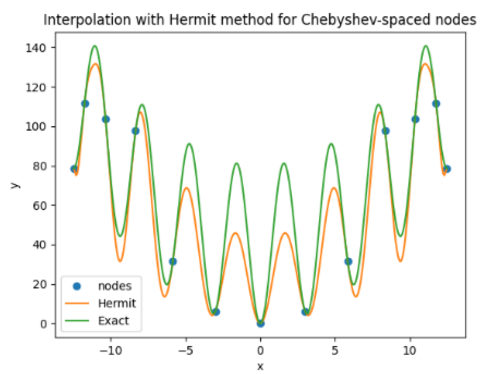
W powyższej tabeli zamieściłem błędy maksymalne algorytmu Hermite’a dla różnej liczby węzłów

Jak widzimy w **Tabeli 1**, zgodnie z przewidywaniami dla bardzo niskich ilości węzłów funkcja jest mocno niedokładna dla obu rozkładów ze względu na małą liczbę informacji o występujących wartościach. Pomyślna interpolacja funkcji zachodzi znacznie szybciej w przypadku rozkładu według zer wielomianu Czebyszewa, bo już przy 19 węzłach jesteśmy w stanie powiedzieć, że funkcja jest dokładna, kiedy w przypadku równomiernego rozkładu otrzymujemy wciąż bardzo duże błędy. W tabeli zauważalna jest również rosnąca tendencja błędu niedługo po otrzymaniu najdokładniejszego z wyników;

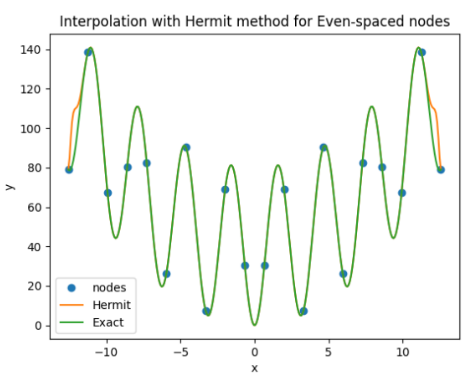
Analiza wykresów interpolacji Hermite'a na różnych poziomach liczby węzłów:



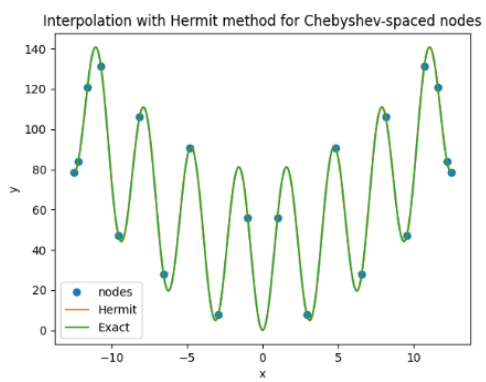
Wykres 1.



Wykres 2.



Wykres 3.

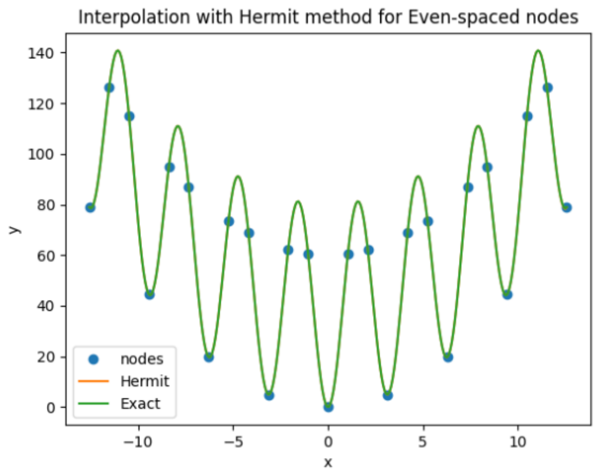


Wykres 4.

Po lewej stronie widać dwa porównania wykresów dla dwóch rozkładów punktów wymienionych przed chwilą. Już na pierwszy rzut oka widać, że prawe wykresy (2,4) są dokładniejsze, niż ich lewe (1,3) odpowiedniki.
W pierwszej parze mamy wykres wielomianu interpolowanego dla rozkładu równomiernego 18 punktów (**1**) i wielomianu interpolowanego dla rozkładu względem zer wielomianu Czebyszewa 13 punktów (**2**). Pomimo mniejszej ilości węzłów, wielomian "Czebyszewa" znacznie lepiej dopasowuje się do funkcji oryginalnej.

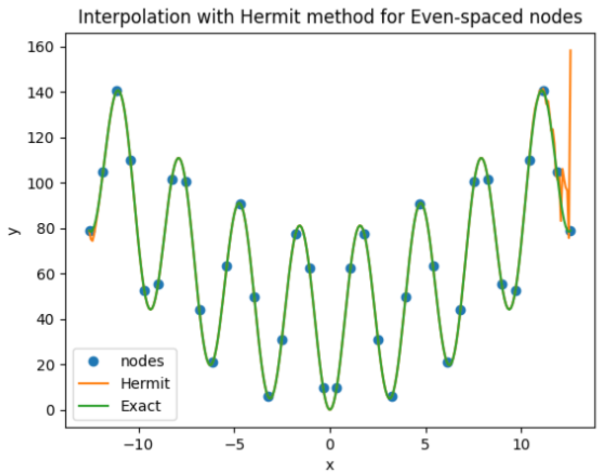
W drugiej parze natomiast widać wykres dla 20 węzłów równomiernie rozłożonych (**3**) i według wielomianu Czebyszewa (**4**) - drugi spośród nich zdążył się już dopasować (oraz osiągnąć najlepsze dopasowanie już przy 19 węzłach), kiedy pierwszy dopiero "dokleja" się do funkcji oryginalnej. Wnioskując po tych obserwacjach możemy stwierdzić, że przy metodzie Hermite'a, tak samo jak w przypadku Lagrange'a, rozkład zgodnie z zerami wielomianu Czebyszewa przynosi rezultaty szybciej.

Śledząc wykresy dalej od 20 węzła jesteśmy w stanie stwierdzić, że:



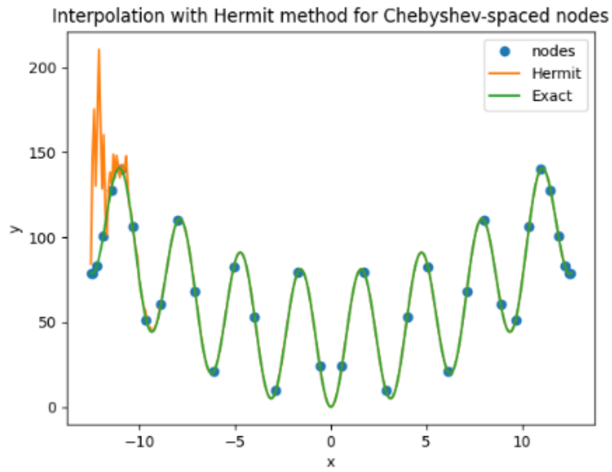
Najbardziej optymalna interpolacja z równomiernie rozłożonymi węzłami (25).

Wykres 5.



Zauważalny i odczuwalny błąd arytmetyki na interpolacji o równomiernie rozłożonych węzłach (36 węzłów co daje 71 stopień wielomianu interpolującego).

Wykres 6.



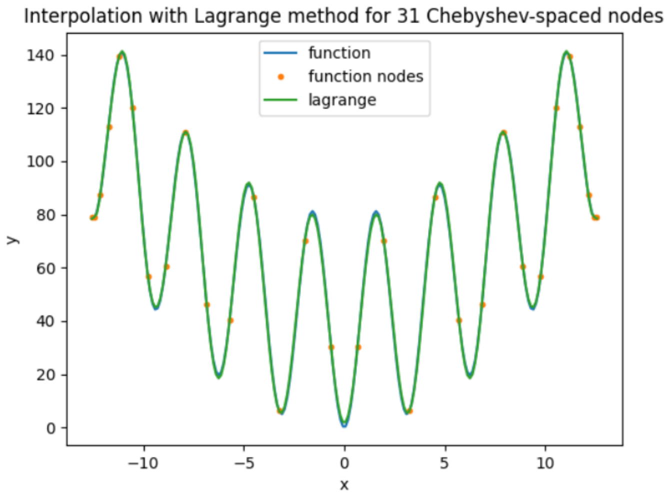
Zauważalny i znacząco odczuwalny błąd arytmetyki na interpolacji o węzłach rozłożonych wg. zer wielomianu Czebyszewa (31 węzłów co daje 61 stopień wielomianu interpolującego)

Wykres 7.

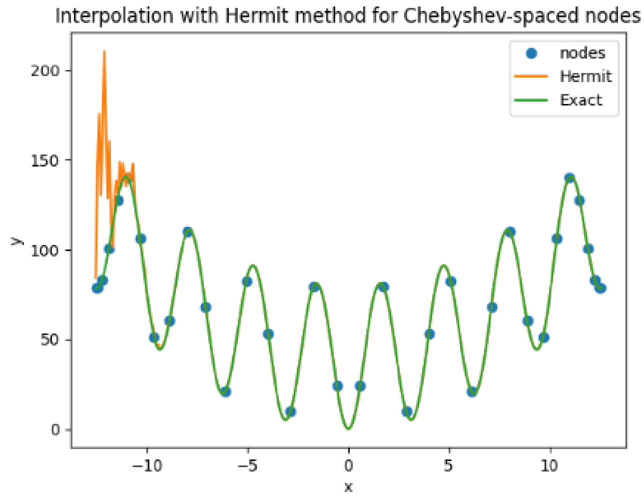
Porównanie zagadnienia Lagrange'a i Hermite'a:

Aby porównać przybliżania funkcji korzystające z tych dwóch zagadnień możemy to zrobić na dwa sposoby: Wyniki najlepszych metod przybliżających dla równej liczby węzłów lub równego stopnia wielomianów interpolujących.

31 węzłów (moment największej dokładności w zagadnieniu Lagrange'a):

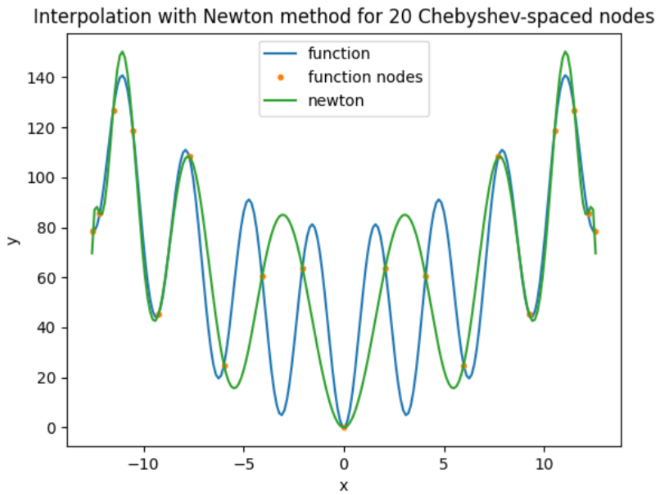


Wykres 8.

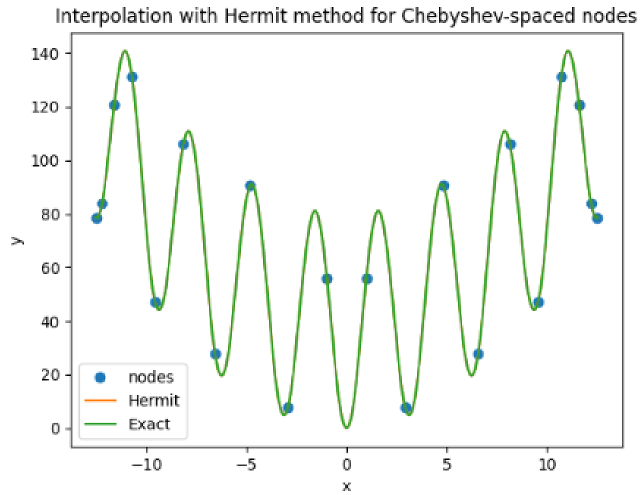


Wykres 9.

20 węzłów (moment największej dokładności w zagadnieniuHermite'a):



Wykres 10.



Wykres 11.

Przeprowadzonymi doświadczeniami w zagadnieniu Lagrange'a oraz Hermite'a potwierdzamy przeczuwany efekt: funkcja Hermite'a przybliża zadaną funkcję szybciej i dokładniej, ponieważ posiada więcej informacji o jej wartościach (**Wykresy 10 i 11**). Widać również różnicę w szybkości straty dokładności, spowodowanej błędami arytmetyki- funkcja Hermite'a traci dokładność szybciej niż funkcja Lagrange'a. Jest to logiczny efekt spowodowany stopniami wielomianów omawianych funkcji: w momencie 31 węzłów kiedy funkcje z zagadnienia Lagrange'a (**Wykres 8**) posiadają wielomian 30 stopnia, funkcja Hermite'a są już na 61 (**Wykres 9**). Jeśli porównać moment w którym obie funkcje zaczynają mieć mały względny błąd maksymalny (29 węzłów dla Lagrange'a i 17 dla Hermite'a) to zauważamy, że obie funkcje zaczynają być dokładne przy podobnym stopniu wielomianu (28 dla Lagrange'a i 33 dla Hermite'a).

Wnioski końcowe:

1. **Tak jak przy zagadnieniu Lagrange'a nawet dla najbardziej optymalnej liczby węzłów równomierne rozłożenie przynosi gorsze skutki niż Czebyszew (najmniejsza wartość 0.2484 vs 0.0044).**
2. **Wyniki interpolacji z węzłami według wielomianu Czebyszewa padają ofiarą błędów arytmetyki szybciej, niż te rozłożone równomiernie.**
3. **Punkt 2 końcowo nie ma większego znaczenia przy porównywaniu który rozkład punktów jest lepszy, bo przy wiedzy o najbardziej optymalnej ilości węzłów jedyną porównywanym parametrem jest dokładność przybliżenia, w którym wygrywa Czebyszev osiągając większą dokładność dla dodatkowo mniejszej liczby węzłów.**
4. **Przyrównując metody Lagrange'a i Hermite'a końcowo dochodzę do wniosków: Zagadnienie Lagrange'a pozwala na najdokładniejsze przybliżenie przy wiedzi o 31 wartościach funkcji, za to zagadnieniu Hermite'a wystarczy 20 wyników, lecz potrzebne są jeszcze informacje o wartościach pochodnych. Trudno więc jednoznacznie powiedzieć która metoda jest ogólnie lepsza - zależy to od danych o funkcji jakimi dysponujemy w momencie potrzeby jej przybliżenia.**

Bibliografia: sprawozdanie powstało na bazie 2 i 3 wykładu z przedmiotu MOWNiT oraz doświadczeń własnych wykonanych w programie Pycharm z użyciem języka Python.