

Interpolacja w zagadnieniu Hermita

Jakub Kędra

Gr. nr 4

Spis treści

Informacje techniczne	3
Zadanie	3
Funkcja	3
Węzły równoodległe oraz wielomiany Czebyszewa	3
Interpolacja w zagadnieniu Hermita	4
Wzór	4
Tablica ilorazów różnicowych	4
Węzły równoodległe	5
Węzły rozmieszczone zgodnie z zerami wielomianu Czebyszewa	0
Błędy	0
Błędy w interpolacji Lagrange'a	1
Efekt Rungego	1
Wnioski	0

Spis tabel

Tabela 1. Informacje techniczne	3
Tabela 2. Błędy obliczeniowe interpolacji Lagrange'a	1

Spis wykresów

Wykres 1. Hermit - równomierny - $n=3$	5
Wykres 2. Hermit - równomierny - $n=3$ bez f. interpolowanej	5
Wykres 3. Hermit - równomierny - $n=4$	5
Wykres 4. Hermit - równomierny - $n=4$ bez f. interpolowanej	5
Wykres 5. Hermit - równomierny - $n=7$	0
Wykres 6. Hermit - równomierny - $n=7$ bez f. interpolowanej	0
Wykres 7. Hermit - równomierny - $n=5$	0
Wykres 8. Hermit - równomierny - $n=8$	0
Wykres 9. Hermit - równomierny - $n=9$	0
Wykres 10. Hermit - równomierny - $n=10$	0
Wykres 11. Hermit - równomierny - $n=15$	0
Wykres 12. Hermit - równomierny - $n=20$	0
Wykres 13. Hermit - wielomiany Czebyszewa - $n=8$	0
Wykres 14. Hermit - wielomiany Czebyszewa - $n=9$	0
Wykres 15. Hermit - wielomiany Czebyszewa - $n=10$	0
Wykres 16. Hermit - wielomiany Czebyszewa - $n=15$	0
Wykres 17. Hermit - wielomiany Czebyszewa - $n=20$	0
Wykres 29. Hermit - równomierny - $n=8$	1
Wykres 30. Hermit - równomierny - $n=9$	1
Wykres 31. Hermit - równomierny - $n=10$	1

Wykres 32. Hermit – wielomiany Chebyschewa - $n=8$	1
Wykres 33. Hermit – wielomiany Chebyschewa - $n=9$	1
Wykres 34. Hermit – wielomiany Chebyschewa - $n=10$	1

Informacje techniczne

Poniższa tabela zawiera informacje sprzętowe

System operacyjny	Windows 10 Home (64bit, kompilacja 19045)
Procesor	i7 9750h
Język programowania	Python
Kompilator	Python 3.8.10

Tabela 1. Informacje techniczne

Zadanie

Celem zadania było wyznaczenie, przeprowadzenie oraz zbadanie wielomianu interpolującego w postaci Hermita dla dwóch różnych rozmieszczeń węzłów.

Funkcja

Interpolowana funkcja prezentuje się następująco:

$$f(x) = e^{-k\sin(mx)} + k\sin(mx) - 1$$

Dla:

- $k = 1$
- $m = 2$
- $x \in [0, 3\pi]$

Podstawiając parametry do wzoru otrzymujemy:

$$f(x) = e^{-\sin(2x)} + \sin(2x) - 1$$

Z miejscami zerowymi dla

$$x = \left\{ \pi n, \pi n + \frac{\pi}{2} \right\}, n \in \mathbb{Z}$$

Oraz jej pochodną:

$$f'(x) = 2 \cdot (1 - e^{-\sin(2x)}) \cdot \cos(2x)$$

Z miejscami zerowymi dla

$$x = \left\{ \frac{\pi n}{2}, \frac{\pi n}{2} - \frac{\pi}{4} \right\}, n \in \mathbb{Z}$$

Węzły równoodległe oraz wielomiany Czebyszewa

Obie interpolacje zostały przeprowadzone dla dwóch rodzajów rozmieszczeń n węzłów:

- Równoodległych – rozmieszczonych równolegle na całej dziedzinie interpolowanej funkcji
- Zgodnie z zerami wielomianu Czebyszewa, zadanych wzorem:

$$x_k = \frac{1}{2}(a + b) + \frac{1}{2}(b - a)\cos\left(\frac{2k - 1}{2n}\pi\right)$$

Dla $k = 1, \dots, n$ oraz dla $[a, b] = [0, 3\pi]$

Interpolacja w zagadnieniu Hermita

Wzór

Interpolacja w zagadnieniu Hermita jest jedną z odmian interpolacji Newtona. W odróżnieniu od metody Newtona, po za informacjami o wartości f w węzłach, mamy też informację o k -krotnych wartościach pochodnych w tych węzłach. Dla celów tego sprawozdania, będziemy korzystali tylko z pierwszej pochodnej.

Do interpolacji w zagadnieniu Hermita został wykorzystany następujący wzór:

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

Gdzie

b_l oznacza współczynnik, otrzymany na podstawie utworzonej w poniższy sposób tablicy ilorazów różnicowych

p_l oznacza wielomian, zdefiniowany wzorem:

$$p_{s(0)} = 1$$

$$p_{s(i)+j}(x) = (x - x_0)^{m_0} (x - x_1)^{m_1} \dots (x - x_{i-1})^{m_{i-1}} (x - x_i)^j$$

Dla $i = 0, 1, \dots, k; j = 0, 1, \dots, m_i - 1$

Tablica ilorazów różnicowych

Aby obliczyć współczynnik b_l skorzystamy z tablicy ilorazów różnicowych jak w metodzie Newtona.

Iloraz różnicowy definiujemy za pomocą wzoru:

- 0-wy iloraz różnicowy:

$$f[x_i] = f(x_i)$$

- 1-szy:

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

- oraz k -ty:

$$f[x_i, x_{i+1}, \dots, x_{i+k}] = \frac{f[x_{i+1}, x_{i+2}, \dots, x_{i+k}] - f[x_i, x_{i+1}, \dots, x_{i+k-1}]}{x_{i+k} - x_i}$$

Tam, gdzie nie da się utworzyć ilorazu, wykorzystamy informację o jego pochodnej, korzystając ze wzoru

$$\exists_{\eta \in (a,b)} f[x_0, x_1, \dots, x_n] = \frac{f^{(n)}(\eta)}{n!}$$

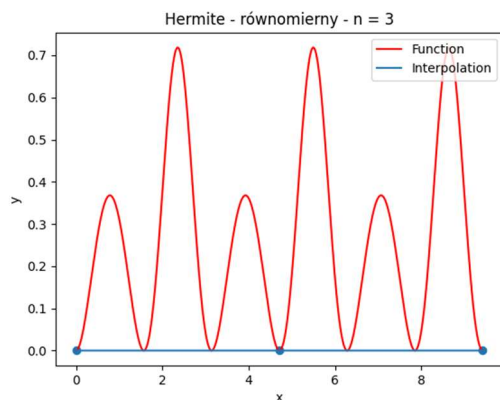
Gdzie

- $f \in C^n[a, b]$

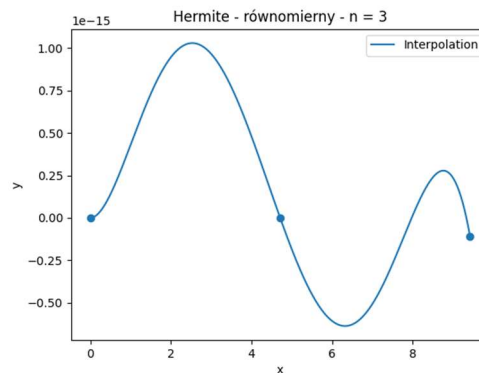
- $x_0, x_1, \dots, x_n \in [a, b]$ i są różne

Węzły równoodległe

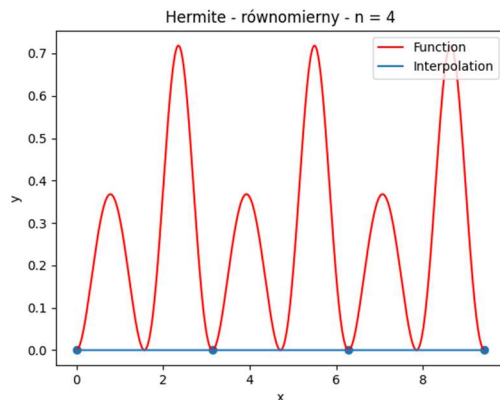
Dla $n \in \{3, 4, 7\}$ możemy zaobserwować dokładnie to samo zjawisko, co w przypadku wcześniejszych interpolacji.



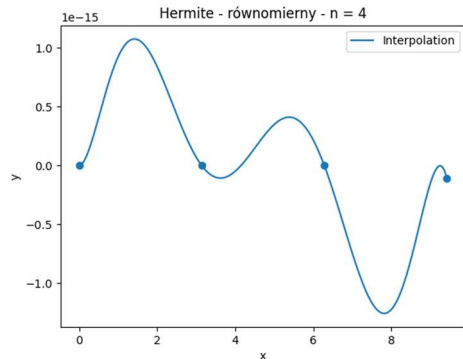
Wykres 1. Hermite - równomierny - n=3



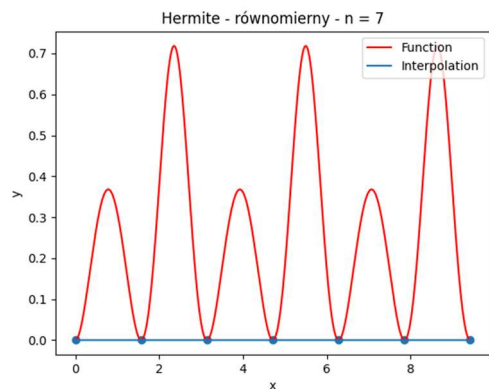
Wykres 2. Hermite - równomierny - n=3 bez f. interpolowanej



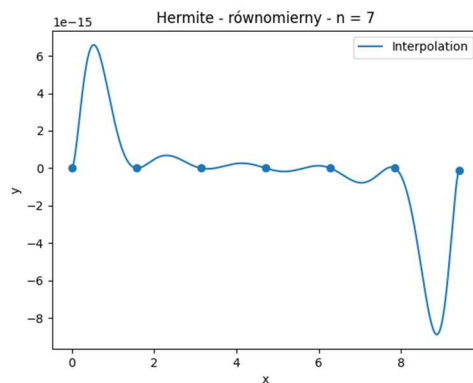
Wykres 3. Hermite - równomierny - n=4



Wykres 4. Hermite - równomierny - n=4 bez f. interpolowanej



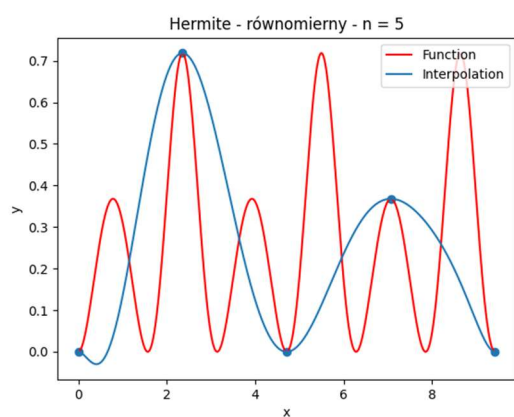
Wykres 5. Hermite - równomierny - $n=7$



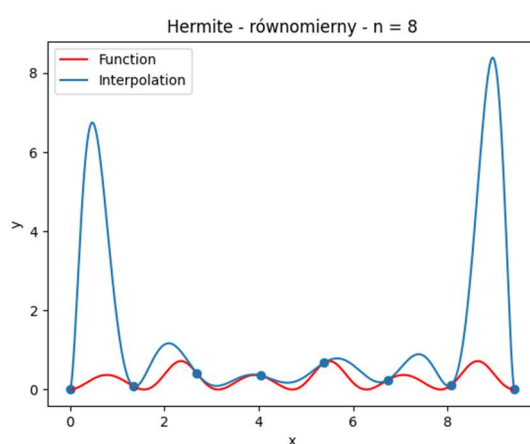
Wykres 6. Hermite - równomierny - $n=7$ bez $f.$ interpolowanej

Interpolowana funkcja wydaje się być prostą, lecz w rzeczywistości wielomiany te tak naprawdę mają bardzo małe amplitudy (rzędu 10^{-15}), przez co w porównaniu z funkcją interpolowaną wygląda dla nas jak zwykła prosta zadana wzorem $y = 0$.

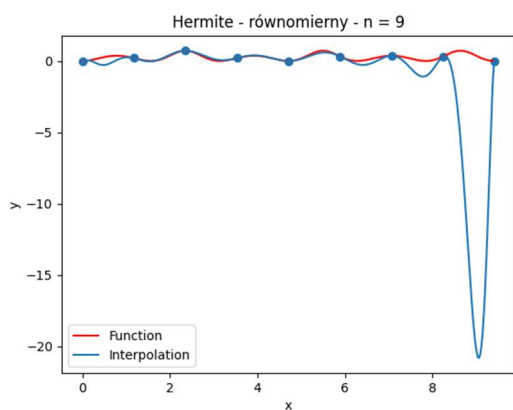
Dla pozostałych przypadków



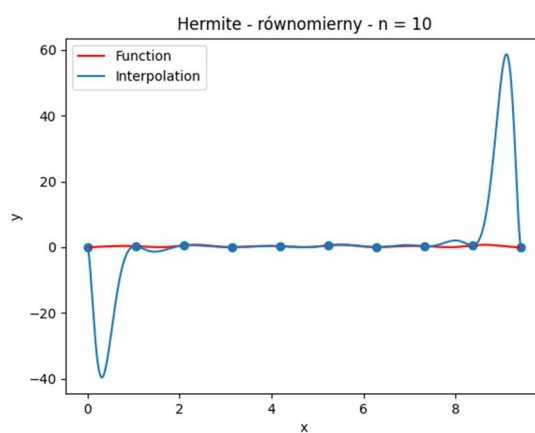
Wykres 7. Hermite - równomierny - $n=5$



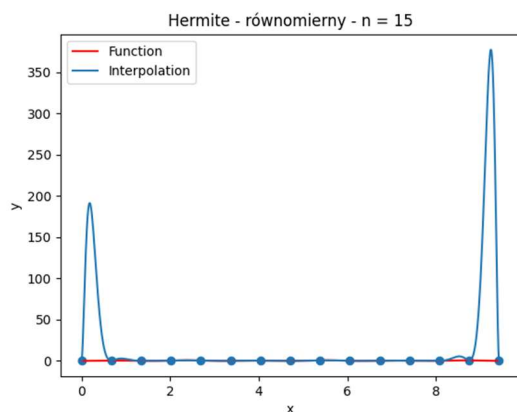
Wykres 8. Hermite - równomierny - $n=8$



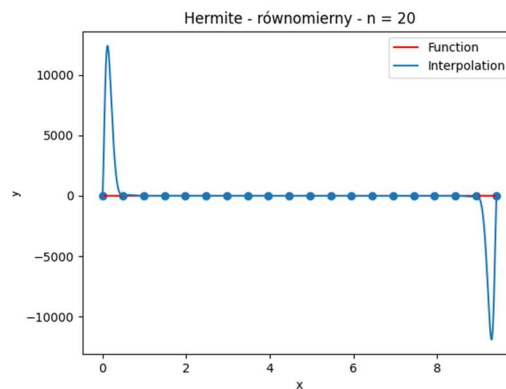
Wykres 9. Hermite - równomierny - $n=9$



Wykres 10. Hermite - równomierny - $n=10$



Wykres 11. Hermit - równomierny - $n=15$

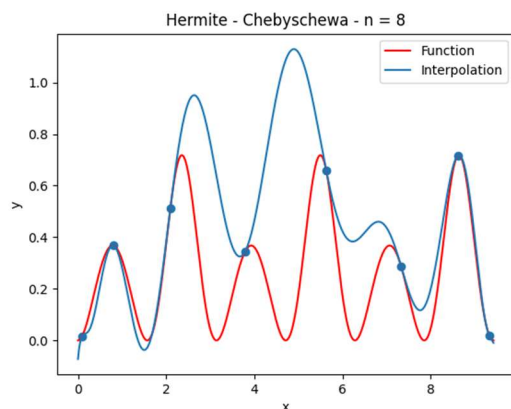


Wykres 12. Hermit - równomierny - $n=20$

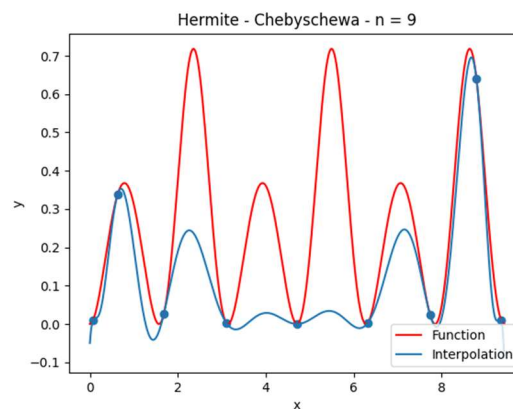
Jak możemy zauważyć, interpolowanie za pomocą węzłów równoodległych niezbyt dokładnie interpoluje naszą funkcję. Dodatkowo, już od dla $n = 8$ możemy zauważyć efekt Runge'go

Węzły rozmieszczone zgodnie z zerami wielomianu Czebyszewa

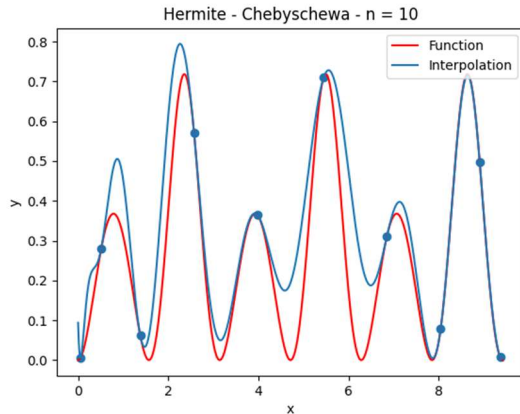
Interpolacja za pomocą wielomianów Czebyszewa daje nam lepsze efekty.



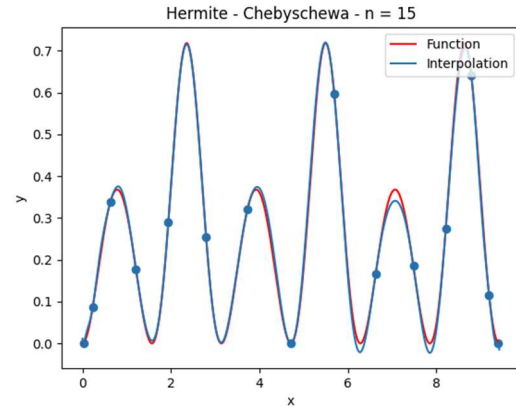
Wykres 13. Hermit - wielomiany Czebyszewa - $n=8$



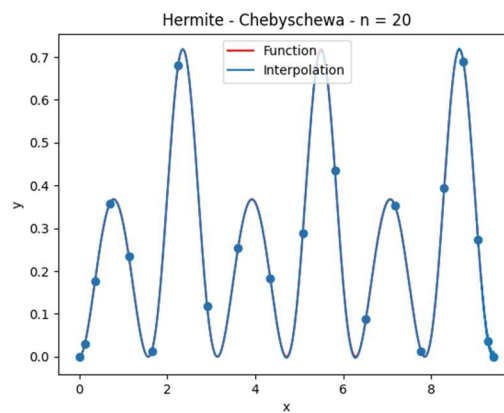
Wykres 14. Hermit - wielomiany Czebyszewa - $n=9$



Wykres 15. Hermit - wielomiany Czebyszewa - $n=10$



Wykres 16. Hermit - wielomiany Czebyszewa - $n=15$



Wykres 17. Hermit - wielomiany Czebyszewa - $n=20$

Jak możemy zauważyć, wraz z zwiększającą się liczbą węzłów, interpolowany wykres staje się coraz dokładniejszy, lecz również, jak w przypadku interpolacji Newtona, od pewnego momentu dokładność interpolacji zaczyna maleć, co możemy zauważyć z poniższej tabeli błędów. Od $n = 15$ interpolowany wykres jest niemal identyczny do wykresu funkcji pierwotnej, a dla $n = 20$ pokrywa ją w całości.

Z powyższych danych wynika, że również w przypadku tej interpolacji rozmieszczenie węzłów ma wpływ na jej dokładność.

Błędy

Do wyliczenia stosownych błędów skorzystałem z poniższych wzorów:

- Wzór na błąd maksymalny punktów:

$$\max_{i=0,\dots,500} |f(x_i) - w(x_i)|$$

- Wzór na błąd sumy kwadratów punktów:

$$\frac{1}{500} \sqrt{\sum_{i=0}^{500} (f(x_i) - w(x_i))^2}$$

Poniżej prezentują się skumulowane w tabelach wyniki obliczeń błędów

Błędy w interpolacji Lagrange'a

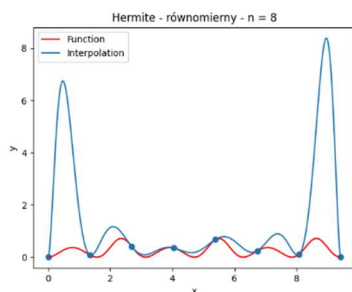
n	równ. max	Chebyshev max	równ. sqrt.	Chebyshev sqrt.
3	7,18E-01	9,63E-01	1,53E-02	1,65E-02
4	7,18E-01	1,32E+00	1,53E-02	2,82E-02
5	6,10E-01	9,89E-01	1,30E-02	1,94E-02
7	7,18E-01	1,02E+00	1,53E-02	1,72E-02
8	7,97E+00	1,10E+00	1,04E-01	1,63E-02
9	2,11E+01	6,85E-01	1,94E-01	1,01E-02
10	5,84E+01	3,13E-01	6,04E-01	4,86E-03
15	3,77E+02	2,68E-02	2,79E+00	4,65E-04
20	1,24E+04	8,16E-03	9,43E+01	6,17E-05
30	6,84E+05	7,05E+03	3,75E+03	3,22E+01
40	3,38E+09	1,26E+10	1,04E+07	3,41E+07
50	2,95E+15	6,79E+15	7,96E+12	2,03E+13

Tabela 2. Błędy obliczeniowe interpolacji w zagadnieniu Hermita

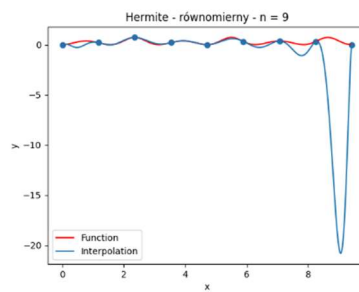
Z powyższej tabeli wynika, że najmniejszy błąd interpolacji możemy uzyskać poprzez rozmieszczenie węzłów zgodnie z zerami wielomianu Czebyszewa. Do $n=20$ dokładność interpolacji zerami Czebyszewa wraz ze wzrostem liczby węzłów się zwiększa, lecz podobnie jak w przypadku interpolacji Newtona, od pewnego momentu dokładność interpolacji zaczyna wraz ze wzrostem liczby węzłów maleć.

Efekt Rungego

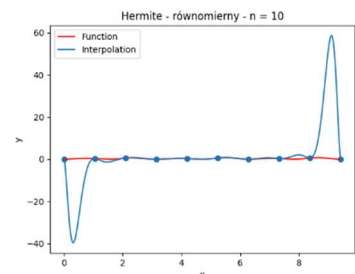
Efekt Rungego polega na występowaniu zmniejszenia dokładności wraz ze zwiększaniem liczby węzłów. Dodatkowo, zachodzi on tylko w przypadku gdy równocześnie interpolujemy wielomianami oraz gdy węzły interpolacyjne są równoodległe. W naszym przypadku, efekt ten możemy zauważyć już dla $n = 8$ w przypadku obu interpolacji z równomiernym rozmieszczeniem węzłów. Poniżej znajdują się porównania interpolowanych funkcji w zależności od wykorzystywanego sposobu rozmieszczenia węzłów.



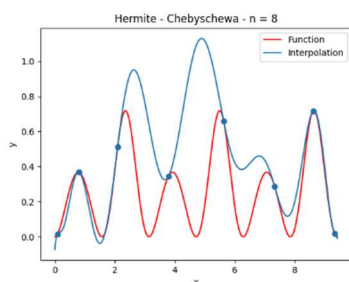
Wykres 18. Hermite - równomierny
- $n=8$



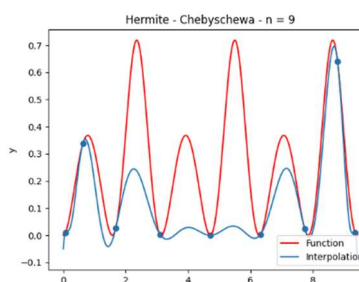
Wykres 19. Hermite - równomierny
- $n=9$



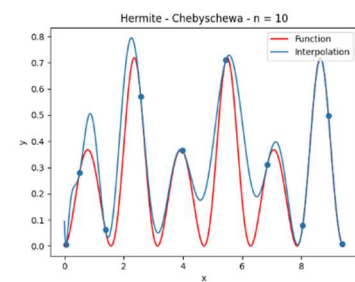
Wykres 20. Hermite - równomierny
- $n=10$



Wykres 21. Hermite - wielomiany
Chebyschewa - $n=8$



Wykres 22. Hermite - wielomiany
Chebyschewa - $n=9$



Wykres 23. Hermite - wielomiany
Chebyschewa - $n=10$

Wnioski

Interpolowane wykresy posiadały największą dokładność wtedy, gdy węzły były rozmieszczone zgodnie z zerami wielomianu Czebyszewa. Wynika z tego, że rozmieszczenie węzłów również w tym przypadku ma znaczenie.

Dokładność interpolacji w zagadnieniu Hermita, wykorzystującej zera Czebyszewa, wzrasta tylko do pewnej liczby węzłów, powyżej której dokładność ta zaczyna maleć. W przypadku interpolowanej na potrzeby tego sprawozdania funkcji, liczba ta wynosi $n = 20$. Jednocześnie w przy tej samej liczbie węzłów co w przypadku interpolacji w zagadnieniu Newtona, daje nam dokładniejszy wynik (w przypadku Newtona, aby uzyskać taki rezultat, potrzeba nam 40 węzłów). Trzeba jednak pamiętać, że po za informacją o liczbie wartości funkcji w danym węźle, posiadamy również informację o wartości pochodnych tej funkcji w tym samym węźle, przez co finalnie operujemy na takiej samej liczbie danych. Na podstawie tego można stwierdzić, że interpolacja w zagadnieniu Hermita ma podobne właściwości do interpolacji w zagadnieniu Newtona.