MOwNiT – Interpolacja - zagadnienie Hermite'a

Przygotował:
Szymon Budziak

Problem:

Dla poniższej funkcji:

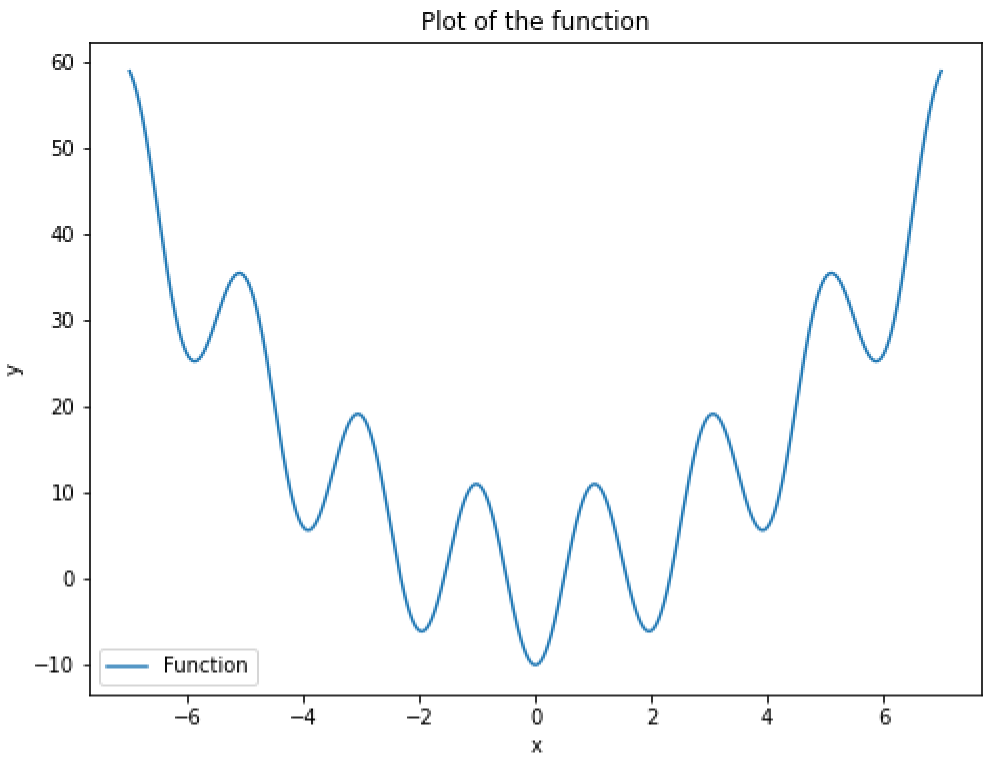
$$f(x) = x^2 - m \cdot \cos\left(\frac{\pi x}{k}\right)$$

$$k=1,\ m=10,\ [-7, 7]$$

wyznacz dla zagadnienia Hermite’a wielomian interpolujący. 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.

Uwaga: Zalecane jest rysowanie wykresów funkcji, wielomianów interpolujących, …, czyli graficzne ilustrowanie przeprowadzonych eksperymentów numerycznych. W sprawozdaniu należy umieścić wykresy jedynie dla wybranych przypadków!

Wykres funkcji



Interpolacja Hermite'a

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

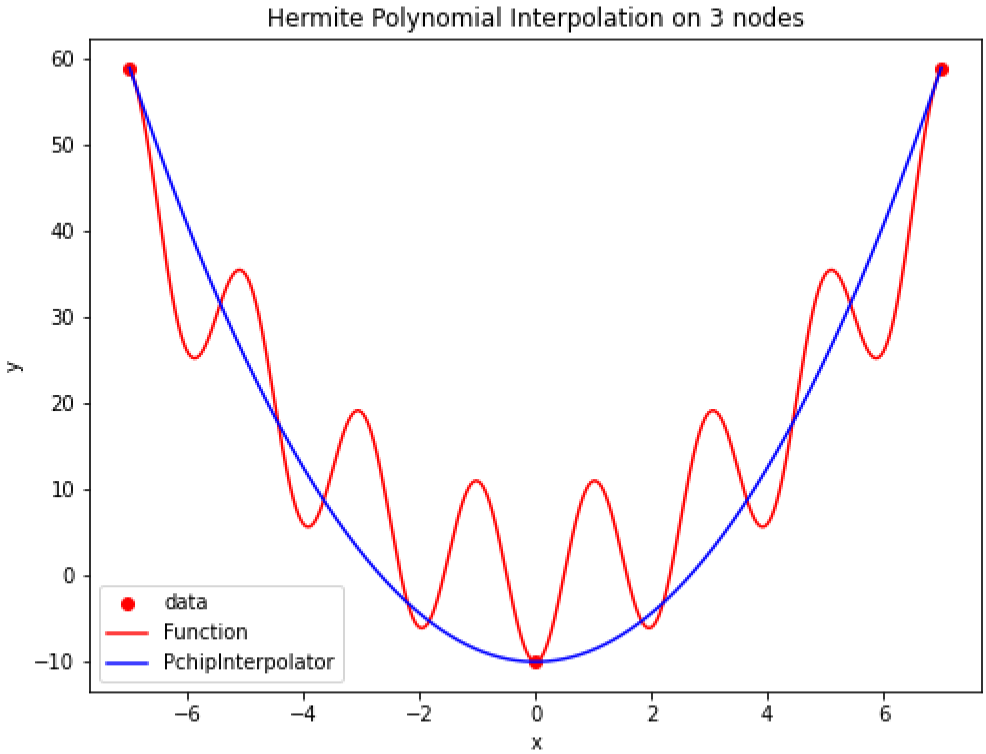
Budowanie tablicy ilorazów różnicowych:

$f'()$

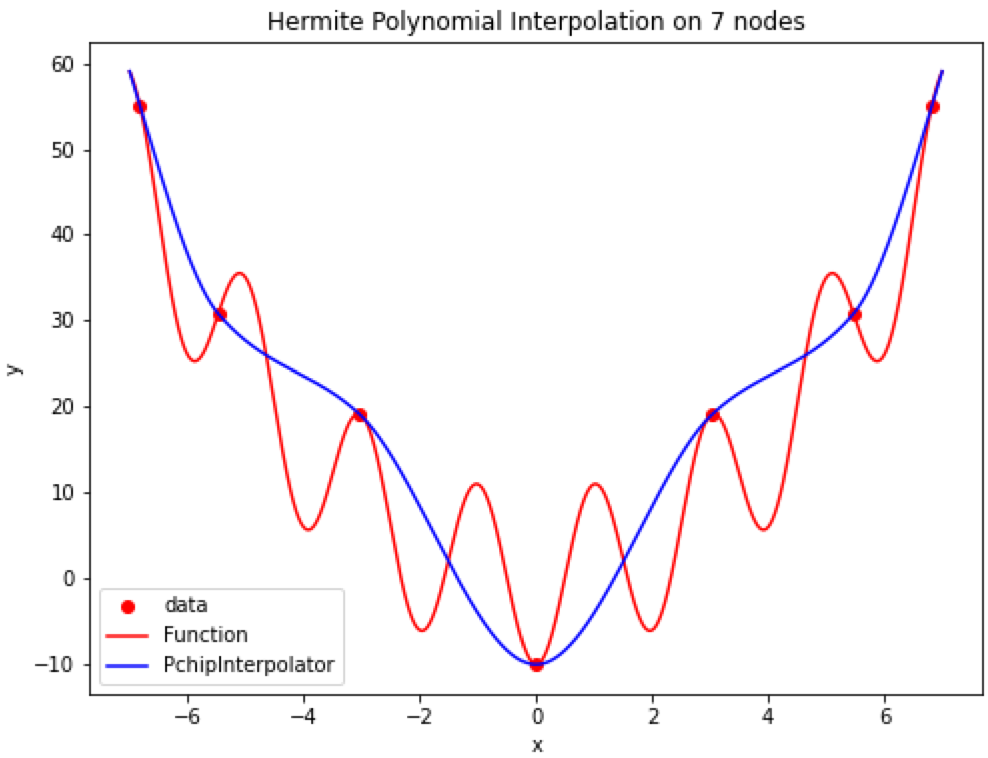
$$\begin{array}{llllll}
x_0 & f(x_0) & & & & \\
x_1 & f(x_1) & f[x_0, x_1] & & & \\
x_1 & f(x_1) & f'(x_1) & f[x_0, x_1, x_1] & & \\
x_1 & f(x_1) & f'(x_1) & \frac{f''(x_1)}{2!} & f[x_0, x_1, x_1, x_1] & \\
\cdots & \cdots & \cdots & \cdots & \cdots & \\
x_n & f(x_n) & f[x_{n-1}, x_n] & .... & ... &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

Do interpolacji Hermite'a został użyty wzór:
Wykonanie interpolacji Hermite'a zostało wykonane dla trzech funkcji: dwóch z pythonowej biblioteki SciPy oraz jednej własnej implementacji. Wyniki wszystkich trzech funkcji zostały porównane i można zauważyć, że wyniki własnej implementacji do pewnej liczby węzłów interpolacyjnych niczym nie różnią się od wyników otrzymanych za pomocą obydwu funkcji z biblioteki SciPy. Można wysunąć wniosek, iż własna implementacja funkcji interpolacji Hermite'a jest poprawna.

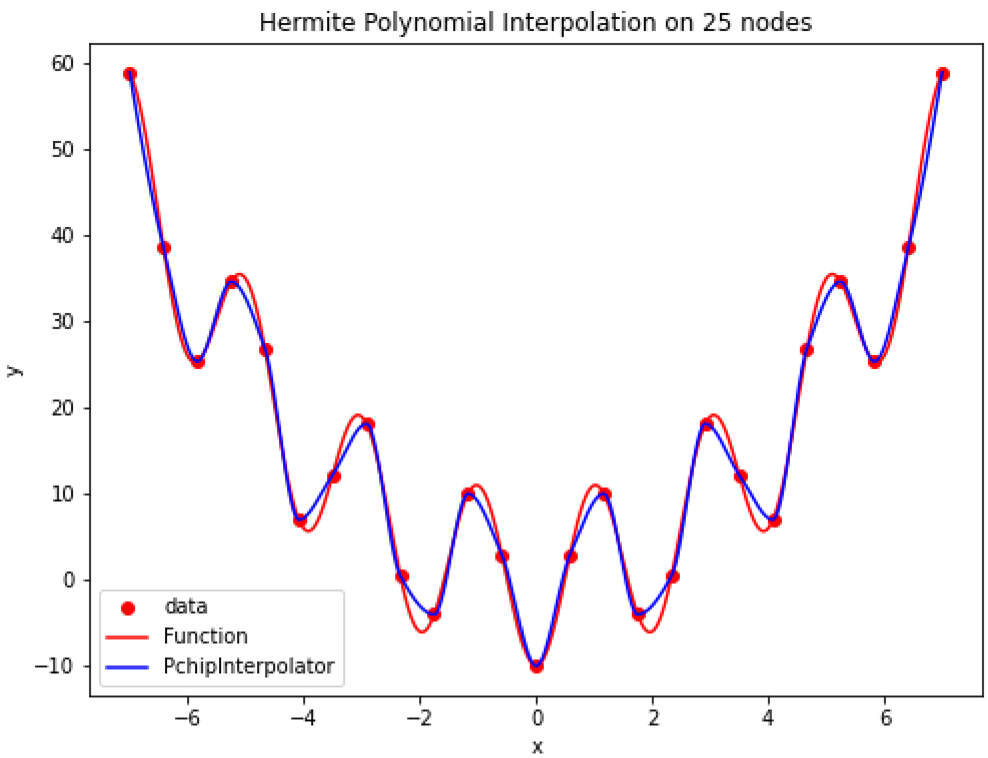
Przykładowe wykresy dla interpolacji Hermite'a przy pomocy funkcji z biblioteki SciPy (interpolate.pchip_interpolate)



Wykres 2: Wykres interpolacji Hermite'a dla funkcji z biblioteki SciPy (interpolate.pchip_interpolate) i 3 węzłów rozmieszczonych równoodlegle

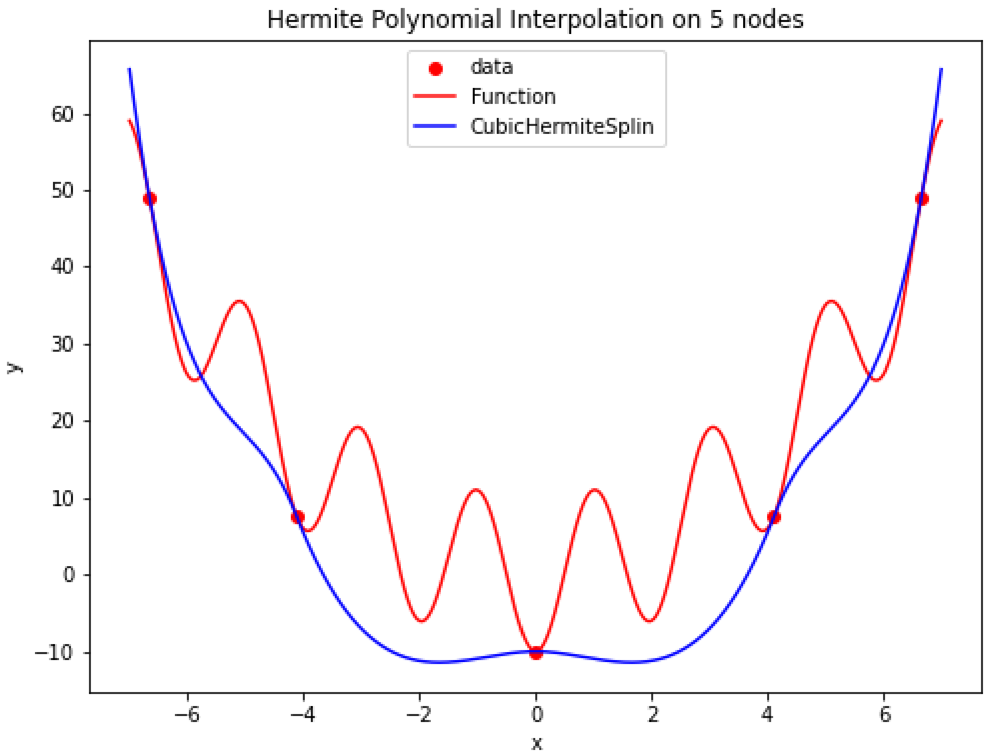


Wykres 3: Wykres interpolacji Hermite'a dla funkcji z biblioteki SciPy (interpolate.pchip_interpolate) i 7 węzłów Czebyszewa

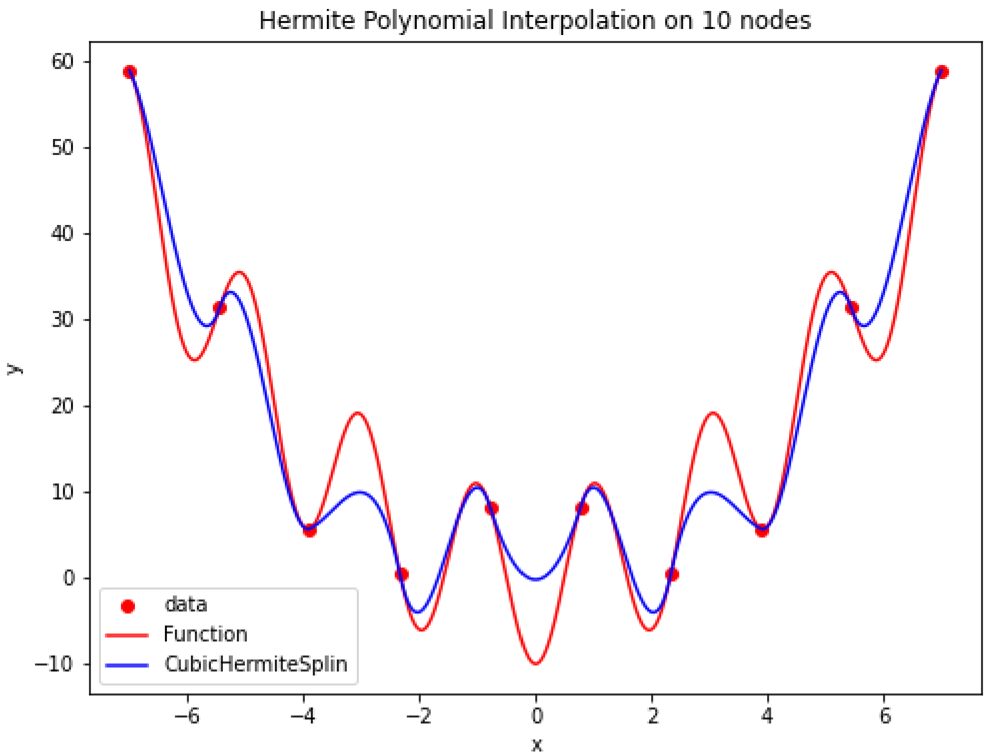


Wykres 4: Wykres interpolacji Hermite'a dla funkcji z biblioteki SciPy (interpolate.pchip_interpolate) i 25 węzłów rozmieszczonych równoodlegle

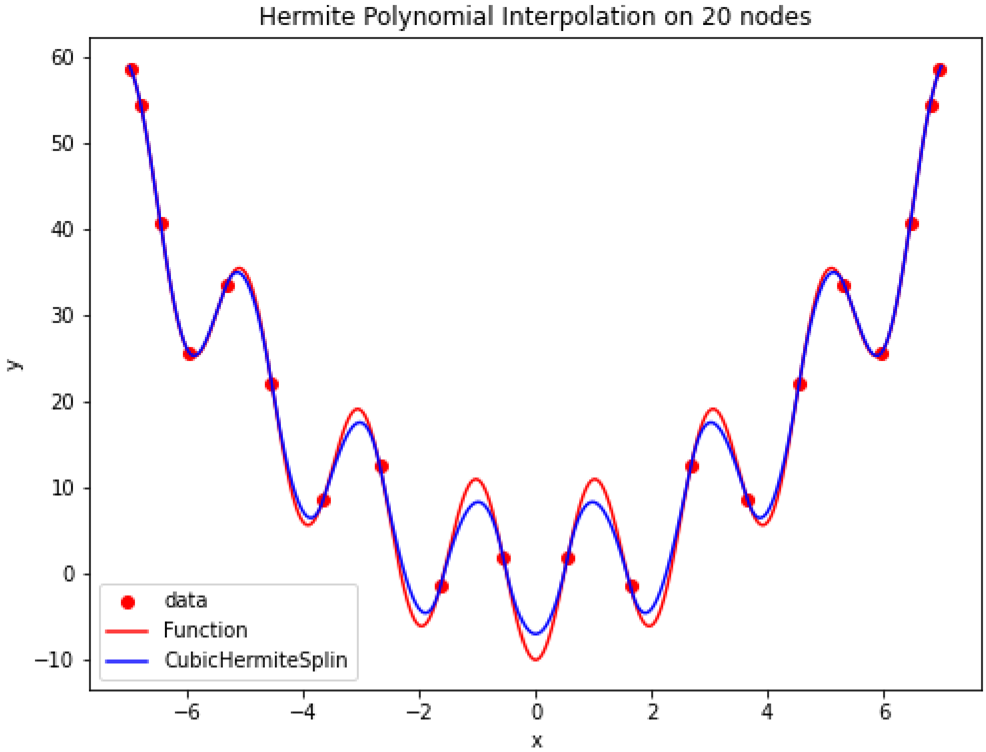
Przykładowe wykresy dla interpolacji Hermite'a przy pomocy funkcji z biblioteki SciPy (interpolate.CubicHermiteSpline)



Wykres 5: Wykres interpolacji Hermite'a dla funkcji z biblioteki SciPy (interpolate.CubicHermiteSpline) i 5 węzłów Czebyszewa

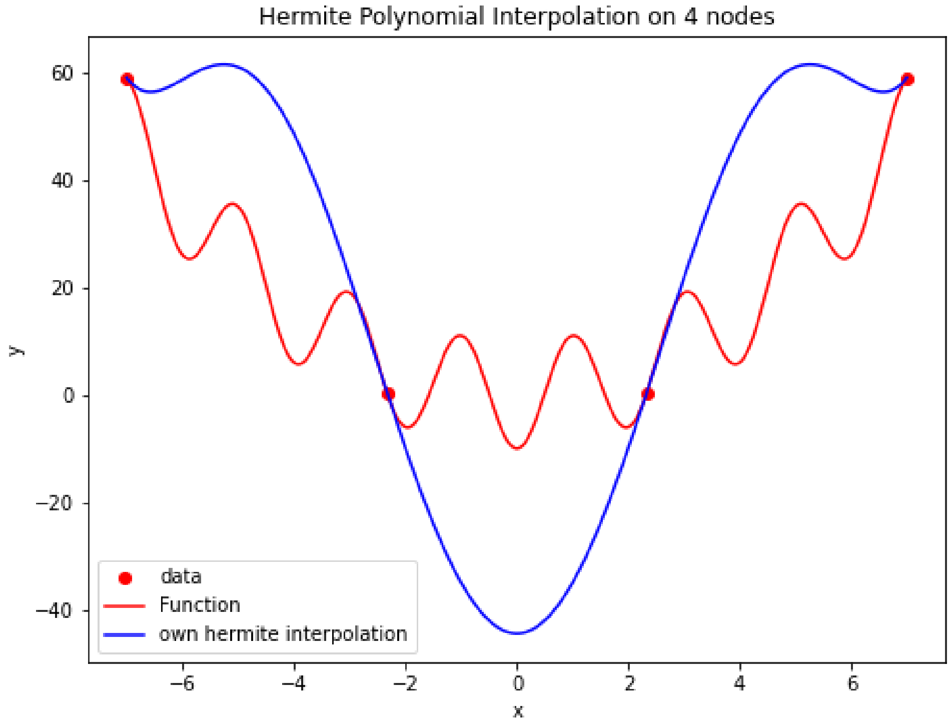


Wykres 6: Wykres interpolacji Hermite'a dla funkcji z biblioteki SciPy (interpolate.CubicHermiteSpline) i 10 węzłów rozmieszczonych równoodlegle

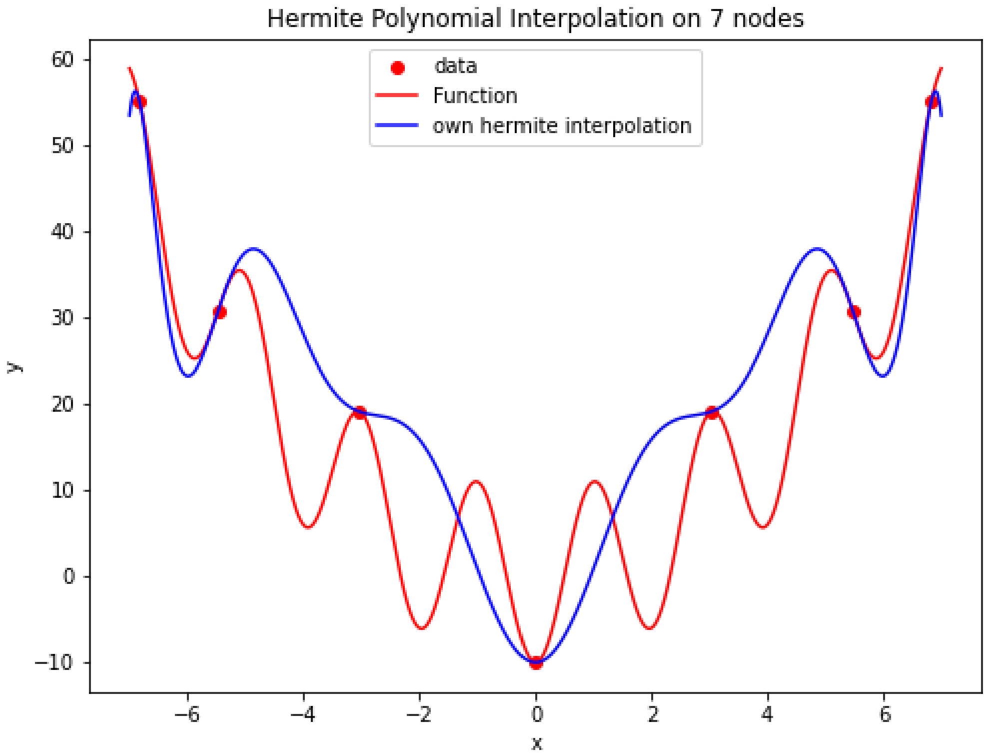


Wykres 7: Wykres interpolacji Hermite'a dla funkcji z biblioteki SciPy (interpolate.CubicHermiteSpline) i 25 węzłów Czebyszewa

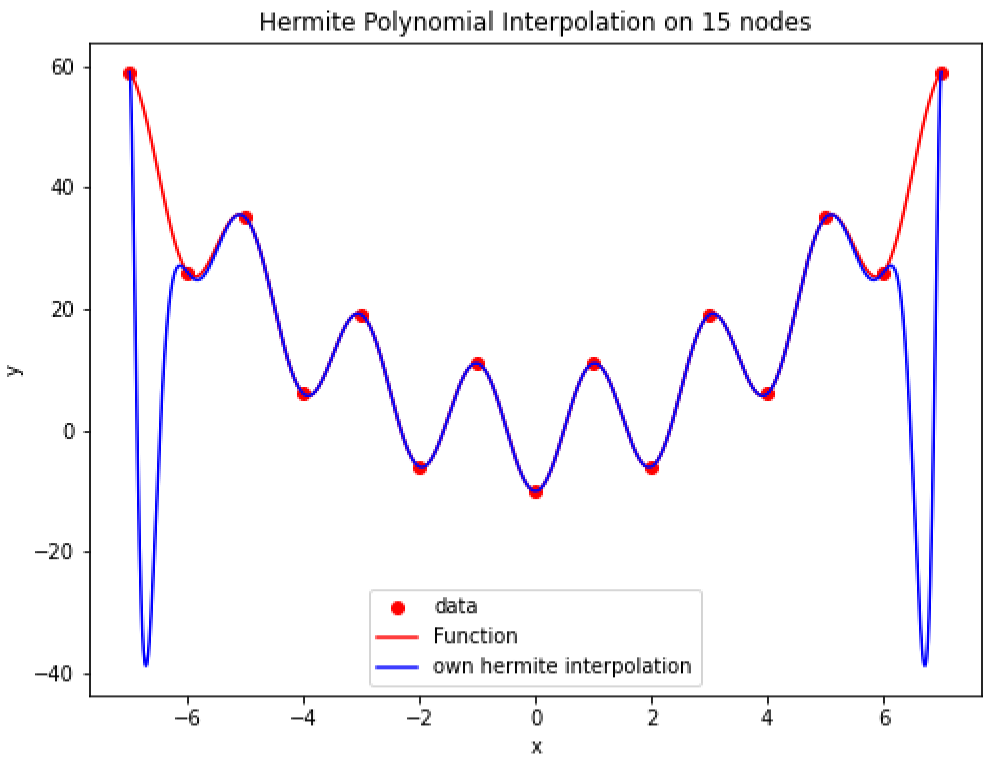
Przykładowe wykresy dla interpolacji Hermite'a przy pomocy własnej implementacji funkcji



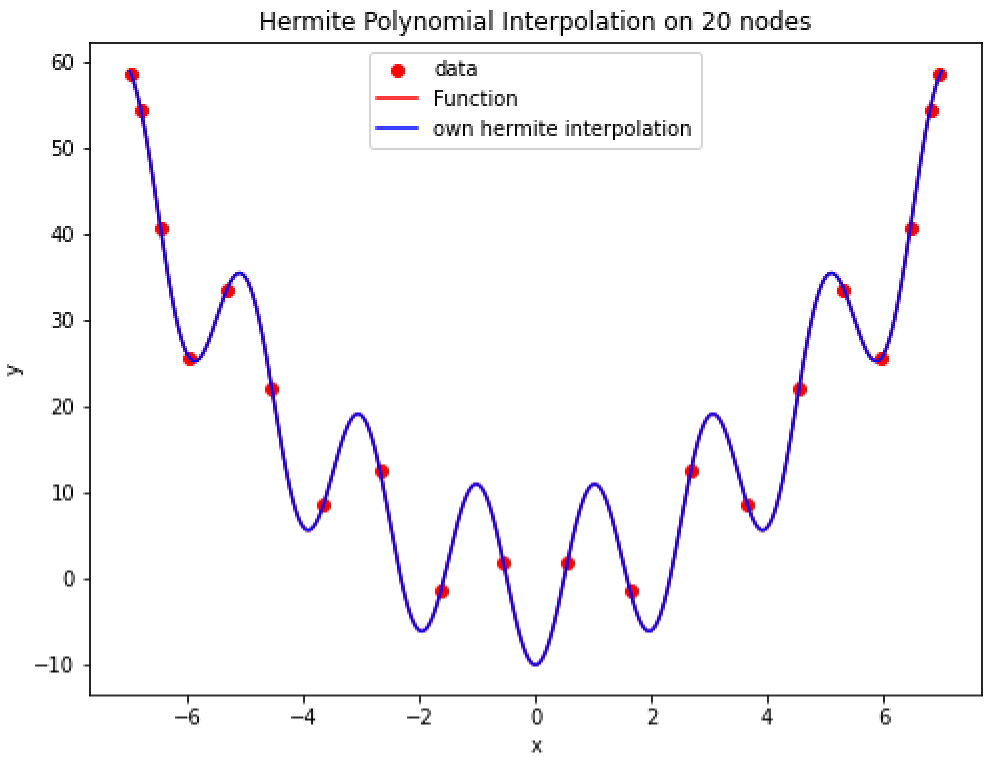
Wykres 8: Wykres interpolacji Hermite'a dla własnej implementacji i 4 węzłów rozmieszczonych równoodlegle



Wykres 9: Wykres interpolacji Hermite'a dla własnej implementacji i 7 węzłów Czebyszewa



Wykres 10: Wykres interpolacji Hermite'a dla własnej implementacji i 15 węzłów rozmieszczonych równoodlegle



Wykres 11: Wykres interpolacji Hermite’a dla własnej implementacji i 20 węzłów Czebyszewa

Błędy obliczeniowe

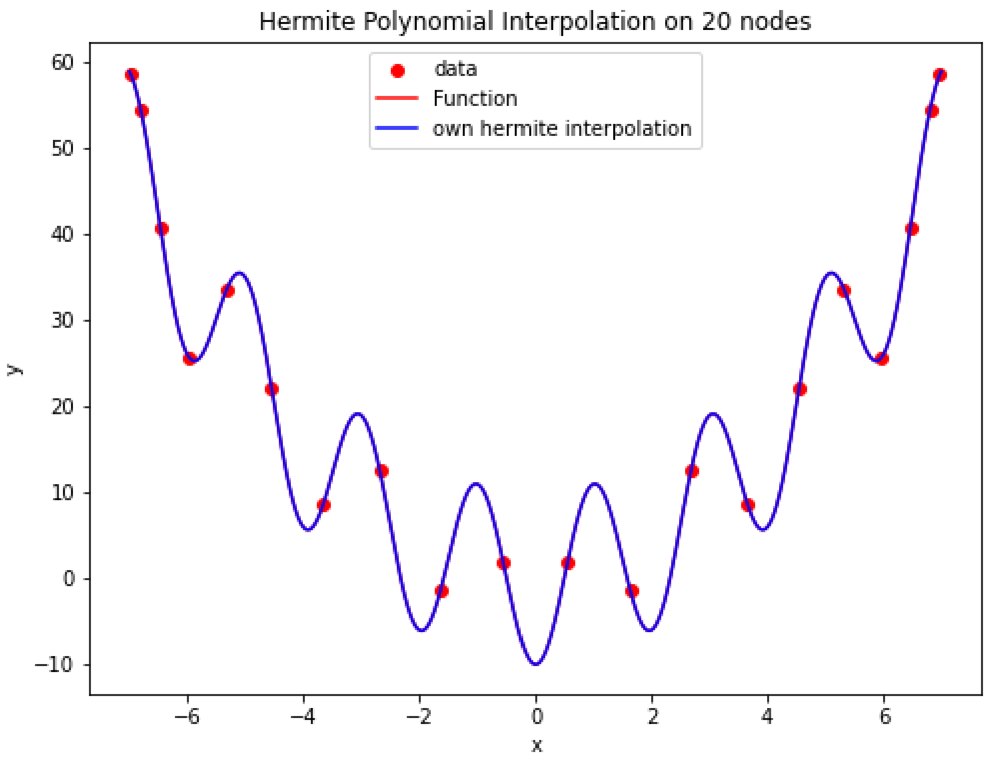
Błędy obliczeniowe zostały wykonane dla błędu maksymalnego punktów (maksymalny błąd z wartości bezwzględnej różnicy pomiędzy kolejnymi punktami) oraz dla błędu sumy kwadratów punktów (suma kwadratów różnic kolejnych punktów). Liczby węzłów jakie zostały wzięte pod uwagę to: 3, 4, 5, 7, 8, 9, 10, 15, 20, 30, 40, 50, 60, 70, 80.

Wyniki dla błędu interpolacji Hermite'a przy zastosowaniu funkcji z biblioteki SciPy (interpolate.pchip_interpolate)

Liczba węzłów	Błąd maksymalny	Błąd sumy kwadratów
3	$1.958 * 10^{1}$	$1.667 * 10^{5}$
4	$3.064 * 10^{1}$	$2.151 * 10^{5}$
5	$5.266 * 10^{1}$	$8.462 * 10^{5}$
7	$1.403 * 10^{2}$	$2.564 * 10^{6}$
8	$2.741 * 10^{1}$	$3.075 * 10^{5}$
9	$6.858 * 10^{2}$	$4.067 * 10^{7}$
10	$1.306 * 10^{3}$	$1.259 * 10^{8}$
15	$9.048 * 10^{1}$	$3.519 * 10^{5}$
20	$6.229 * 10^{-2}$	$1.132 * 10^{-1}$
30	$2.755 * 10^{-1}$	$3.829 * 10^{-1}$
40	$3.192 * 10^{6}$	$3.857 * 10^{13}$
50	$1.026 * 10^{17}$	$4.664 * 10^{34}$
60	$3.422 * 10^{26}$	$4.582 * 10^{53}$
70	$7.649 * 10^{36}$	$1.159 * 10^{74}$
80	$1.504 * 10^{45}$	$7.548 * 10^{90}$

Tabela 1: Błąd obliczeniowy dla interpolacji Hermite'a (własna implementacja)

Z tabeli można zauważyć, że najmniejszy błąd przy interpolacji Hermite'a funkcja ma dla 20 węzłów.



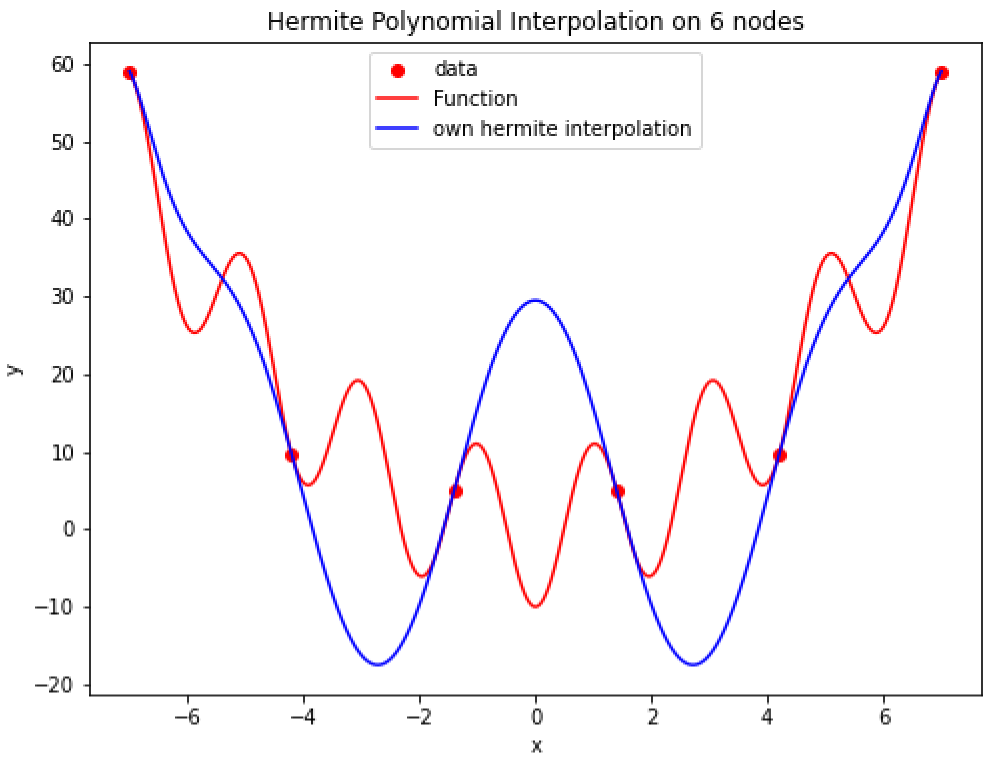
Wykres 12: Wykres interpolacji Hermite'a dla własnej implementacji i 20 węzłów Czebyszewa

Wykres dla interpolacji Hermite'a jest identyczny dla 20 węzłów w przypadku rozmieszczenia równomiernego oraz Czebyszewa. Różnica jest tylko w rozmieszczeniu punktów jakie powstają przy użycie tych dwóch metod, oraz przez które przechodzi nasza funkcja interpolowana.

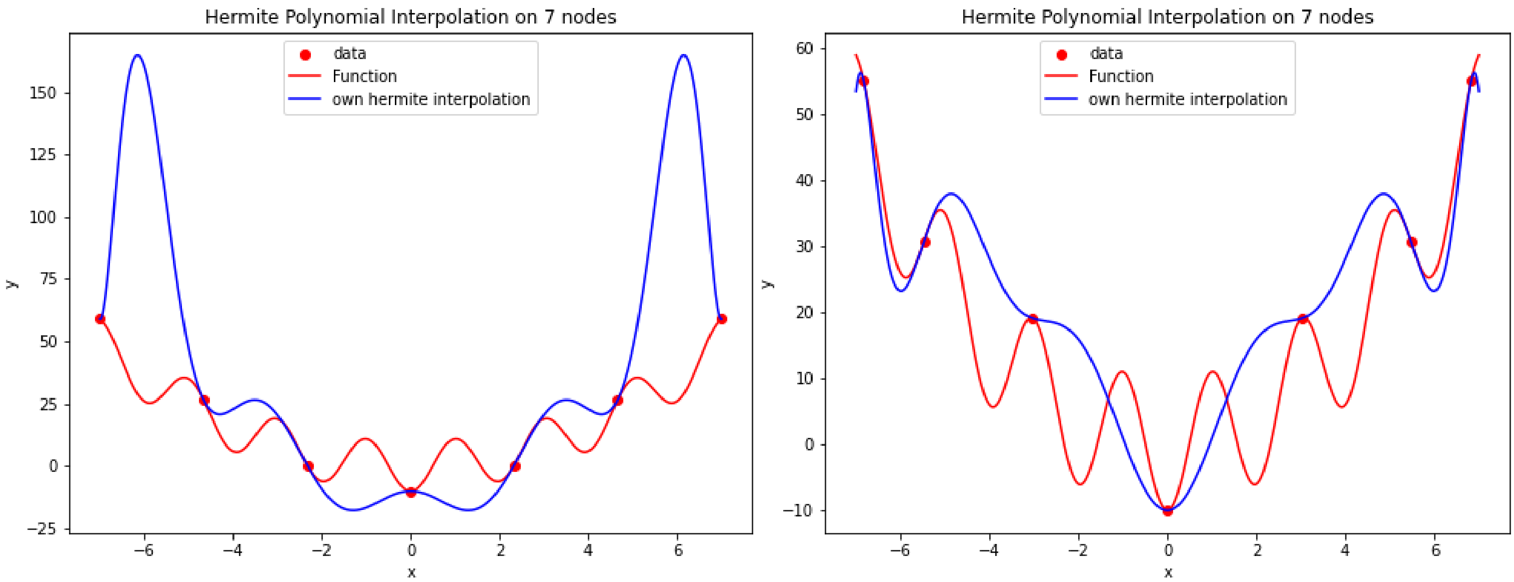
Na co warto jednak zwrócić uwagę to bardzo duże błędy przy większej liczbie węzłów, co wynika z bardzo wysokiego stopnia wielomianu interpolującego. Już dla 50 węzła interpolacji rząd wielkości błędu w przypadku błędu maksymalnego to 10^{17} a w przypadku błędu sumy kwadratów to 10^{34} dla tej samej liczby węzłów interpolacji.

Efekt Rungego

Efekt Rungego jest to pogorszenie jakości interpolacji wielomianowej, mimo zwiększenia liczby jej węzłów. Początkowo ze wzrostem liczby węzłów *n* przybliżenie poprawia się, jednak po dalszym wzroście *n,* zaczyna się pogarszać, co jest szczególnie widoczne na końcach przedziałów.
Dla naszej funkcji efekt Rungego możemy już zaobserwować dla liczby węzłów 7 przy ich równoodległym rozmieszczeniu. Sprawdźmy jednak jak wygląda nasza funkcja dla 6 węzłów interpolacyjnych przy ich równoodległym rozmieszczeniu.



Wykres 13: Wykres interpolacji Hermite'a dla własnej implementacji i 6 węzłów rozmieszczonych równoodlegle (brak efektu Rungego)



Wykres 11: Wykres interpolacji Hermite’a dla własnej implementacji i 7 węzłów rozmieszczonych równoodlegle (po lewo) oraz 7 węzłów Czebyszewa (po prawo), możemy tutaj zaobserwować efekt Rungego dla równoodległego rozmieszczenia węzłów

Dla liczby węzłów interpolacji równej 7 mamy pierwsze wystąpienie efektu Rungego. Wraz ze wzrastającą liczbą węzłów efekt ten zanika przez pewien czas jednak później znowu się pojawia i powiększa się (więcej przykładów wykresów w jupyter notebooku).

Na przedstawionych wykresach można zauważyć odgięcia na krańcach przedziału zwiększające błąd. Można również zauważyć, że zastosowanie węzłów Czebyszewa zapobiega występowaniu efektu Rungego, ulepszając dopasowanie funkcji interpolującej do funkcji interpolowanej. Najlepsze dopasowanie przy metodzie Hermite’a było dla 20 węzłów (wykres tej funkcji był wyżej) Czebyszewa, ponieważ mieliśmy wtedy najmniejszy błąd.

Literatura:

[1] Wykłady nr 2 oraz nr 3 dr Rycerz z przedmiotu MOwNiT

[2] Wikipedia na temat interpolacji Hermite’a, krzywej sześciennej Hermita oraz efektu Rungego