Interpolacja Hermite'a

Natalia Luberda

1. Dane techniczne:

Do obliczeń użyłam języka Python 3.9, korzystając głównie z bibliotek numpy, Math i pyplot. System operacyjny, na którym pracowałam to Windows 10. Procesor komputer to Intel® Core™ i5-10210U CPU @ 1.60GHz 2.11GHz, a ilość pamięci RAM to 8GB.

2. Interpolacja Hermite'a:

Interpolacja to metoda numeryczna, która polega na wyznaczeniu funkcji (lub wielomianu) przechodzącej przez zestaw punktów danych.

Interpolacja jest użyteczna w przypadkach, gdy chcemy uzyskać wartości funkcji w punktach, dla których nie mamy bezpośrednich pomiarów lub dla których pomiar jest trudny lub kosztowny. Pozwala nam również na szacowanie wartości funkcji wewnątrz zakresu, dla którego mamy dane.

Interpolacja Hermite'a to metoda numeryczna, która pozwala na znalezienie wielomianu, który przechodzi przez zadane punkty i ma zadane pochodne w tych punktach. Metoda ta została nazwana na cześć francuskiego matematyka Charles'a Hermite'a.

Idea interpolacji Hermite'a polega na znalezieniu wielomianu o minimalnym stopniu, który spełnia zadane warunki. W szczególności, dla każdego punktu interpolacji należy określić wartość wielomianu oraz jego pochodną.

Interpolacja Hermite'a jest szczególnie przydatna w sytuacjach, gdy dane wejściowe są nieregularne lub niejednoznaczne. Metoda ta znajduje zastosowanie w wielu dziedzinach, w tym w inżynierii, matematyce, fizyce i informatyce.

3. Wzór i wykres funkcji użytej do analizy:

Otrzymana funkcja do analizy oraz jej zakres to:

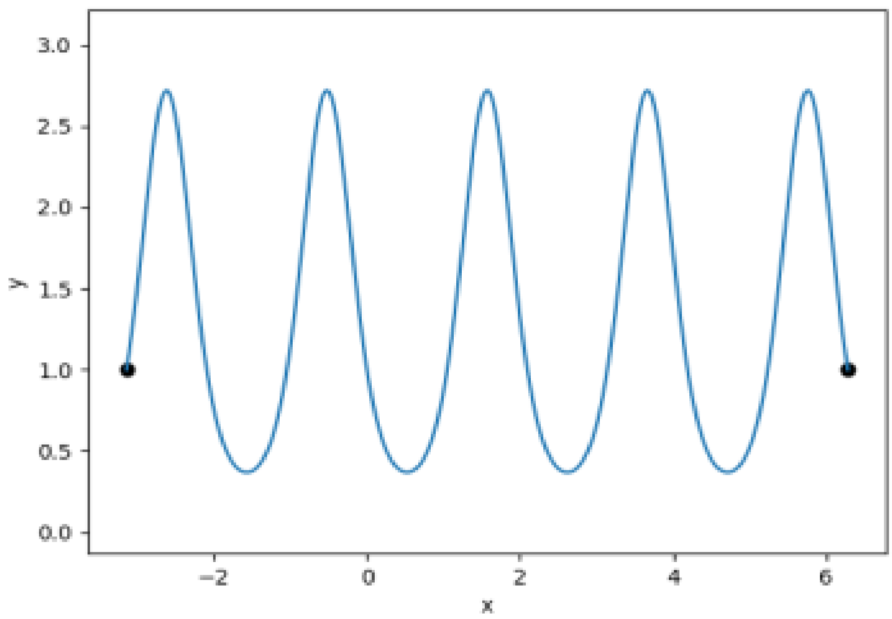
$$f(x) = e^{-\sin(3\cdot x)}, gdzie\ x \in < -\pi, 2\cdot\pi >$$

Wzór I: Wzór funkcji użytej do analizy

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

Wzór II: Wzór na dobranie n węzłów zgodnie z zerami wielomianu Czebyszewa w zadanym przedziale

Wykres I: Wykres funkcji interpolowanej



4. Wielomian interpolacyjny Hermite'a:

Dane są węzły interpolacji $x_0 < x_1 < x_2 < \cdots < x_k$, przy czym punkt x_k, występuje k_i razy na liście węzłów. Dodatkowo znana jest wartość funkcji i jej pochodnych w tych węzłach. Ustawiamy x_i w szeregu (każdy k_i krotnie) i oznaczamy je przez $x_0, \dots, x_n$, gdzie $x_i \leq x_{i+1}$. Wtedy wielomian interpolacyjny Hermite'a, jest wyznaczany numerycznie ze wzoru Newtona i ma następującą postać:

$$p(x) = f[x_0] + f[x_0, x_1](x - x_0) + \dots + f[x_0, x_1, \dots, x_n](x - x_0)(x - x_1) \cdot \ldots \cdot (x - x_{n-1})$$

Wzór III: Wzór wielomianu interpolacyjnego Hermite'a

Gdzie:

$$f[x_i, x_{i+1}, \dots, x_{i+j}] = \begin{cases} \dfrac{1}{j!} f^{(j)}(x_i) & , x_i = x_{i+j} \\ \dfrac{f[x_{i+1}, x_{i+2}, \dots, x_{i+j}] - f[x_i, x_{i+1}, \dots, x_{i+j-1}]}{x_{i+j} - x_i} & , x_i \neq x_{i+j} \end{cases}$$

Wzór IV: Wzór wyznaczania ilorazów różnicowych

5. Implementacja i opis funkcji:

Funkcja Hermite_function przeprowadza interpolację Hermite'a dla danych punktów (x,y) i zwraca wartość interpolowanej funkcji w zadanym punkcie x_point. Kod składa się z kilku etapów.

Funkcja najpierw importuje biblioteki numpy, matplotlib.pyplot oraz pandas. Biblioteka numpy służy do wykonywania obliczeń numerycznych na tablicach i macierzach, matplotlib.pyplot jest używana do tworzenia wykresów i wizualizacji danych, a pandas służy do manipulowania i analizowania danych w formie tabelarycznej.

Następnie definiowane są dwie funkcje: f i f_prim. Funkcja f zwraca wartość funkcji e^-sin(3x), a funkcja f_prim zwraca wartość pochodnej tej funkcji.

W funkcji Hermite_function inicjowana jest macierz Macierz_wart o rozmiarze (2n+1)x(2n+1), gdzie n to liczba punktów danych(węzłów). Macierz ta jest wypełniona zerami. Następnie pierwsze dwie kolumny tej macierzy są ustawione na wartości x i y dla punktów danych.

Następnie wykorzystywana jest metoda podwójnych węzłów interpolacji Hermite'a, dlatego wartości x i y dla każdego punktu są ustawiane w dwóch kolejnych wierszach macierzy. Następnie obliczane są różnice dzielone dla danych wejściowych przy użyciu rekurencyjnej formuły interpolacji Hermite'a.

Na koniec, korzystając z obliczonych różnic dzielonych, funkcja interpolowana jest w zadanym punkcie x_point przy użyciu zagnieżdżonego mnożenia formuły Newtona dla różnic dzielonych. Wynik interpolacji jest zwracany jako wartość funkcji Hermite_function.

Funkcja Hermite_function zwraca wartość interpolowanej funkcji w punkcie x_point(których jest 500 i są one albo wyznaczane równolegle, albo za pomocą zer w wielomianie Czebyszewa).

6. Wykresy i obliczanie błędów:

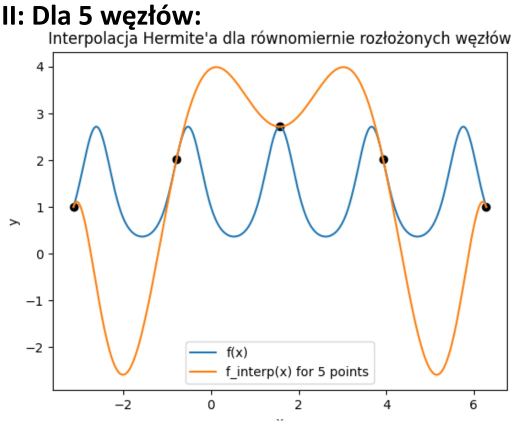
Obliczeń dokonujemy dla 5,7,9,11,13,15,50,80 węzłów. Funkcja obliczana i przybliżana za pomocą 500 punktów rozłożonych:

a) Równolegle
b) Zgodnie z zerami wielomianu Czebyszewa

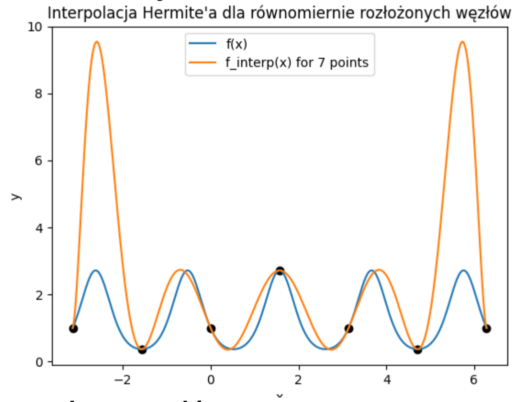
$Wzór\ V$: $\frac{1}{N} \cdot \sqrt{\sum_{i=1}^{N} \left(f\left(x_{interept,i}\right) - y_{interept,i}\right)^2}$, gdzie N duże to liczba punktów, których wartość obliczamy z wyznaczonego wielomianu interpolacyjnego

a) Węzły rozłożone równomiernie:

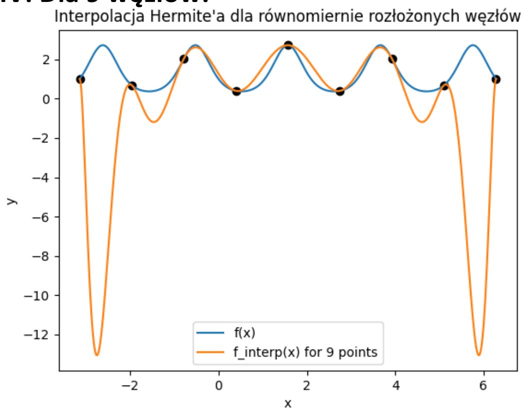
Wykres II: Dla 5 węzłów:



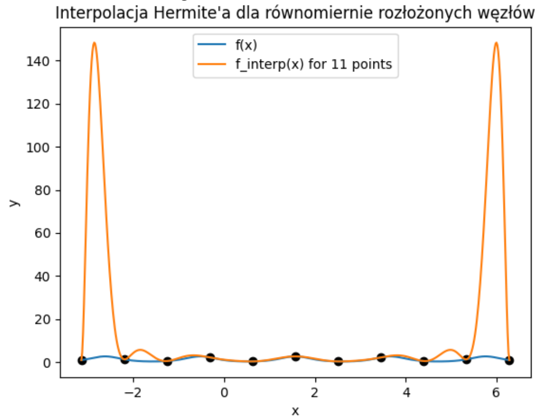
Wykres III: Dla 7 węzłów:



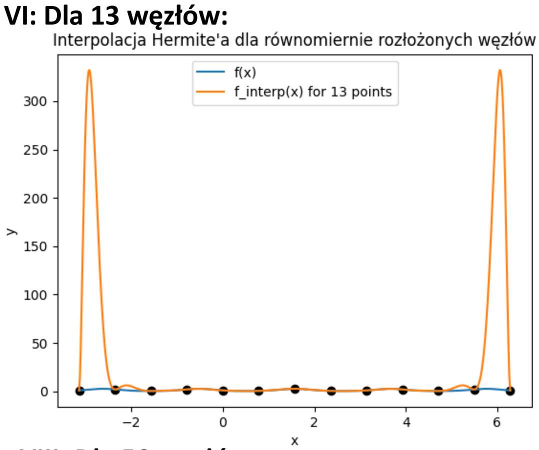
Wykres IV: Dla 9 węzłów:



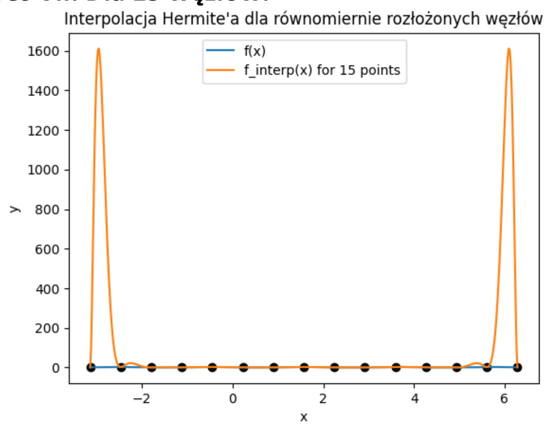
Wykres V: Dla 11 węzłów:



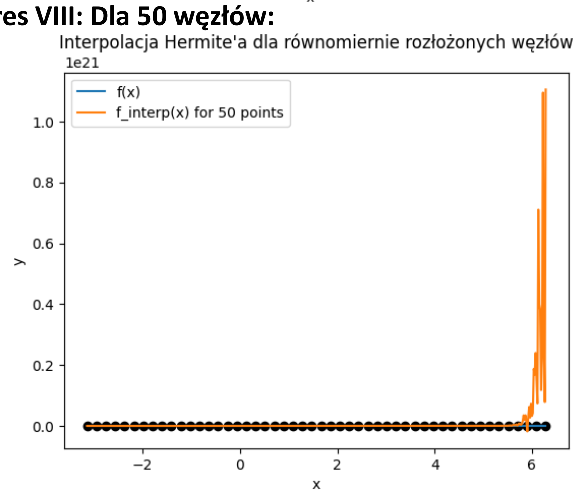
Wykres VI: Dla 13 węzłów:



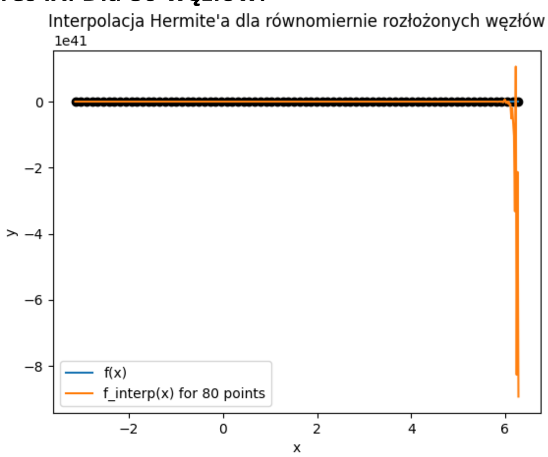
Wykres VII: Dla 15 węzłów:



Wykres VIII: Dla 50 węzłów:

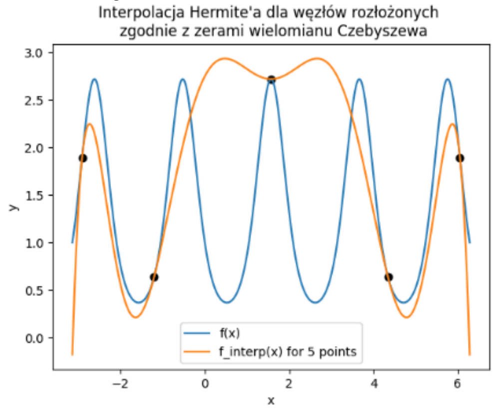


Wykres IX: Dla 80 węzłów:

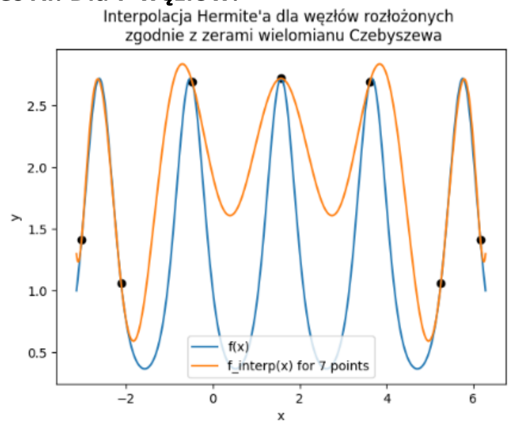


b) Węzły rozłożone zgodnie z zerami wielomianu Czebyszewa:

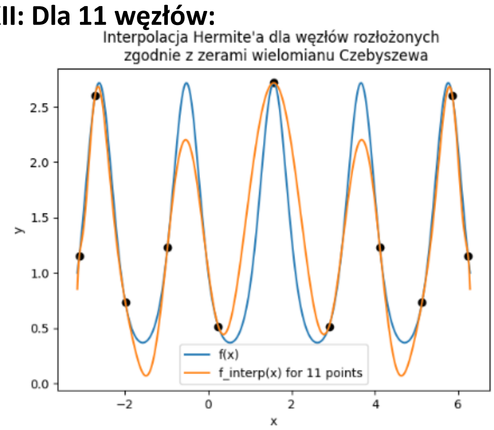
Wykres X: Dla 5 węzłów:



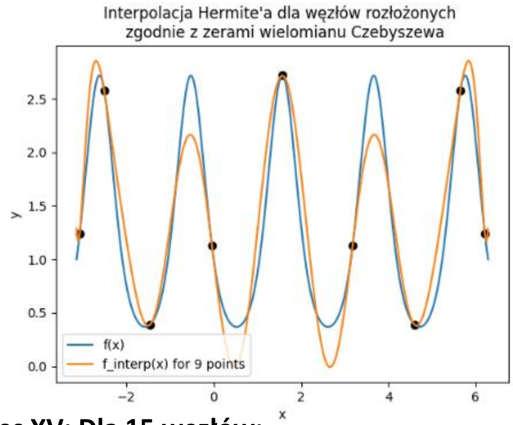
Wykres XI: Dla 7 węzłów:



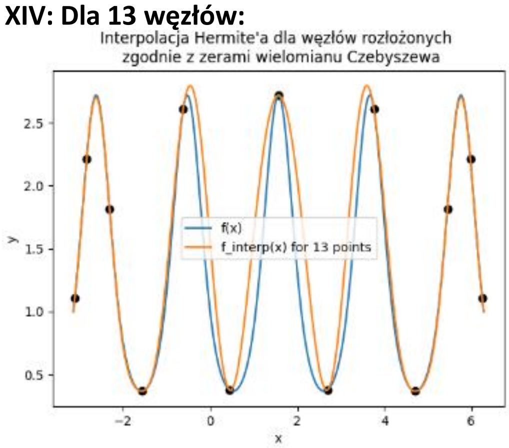
Wykres XII: Dla 11 węzłów:



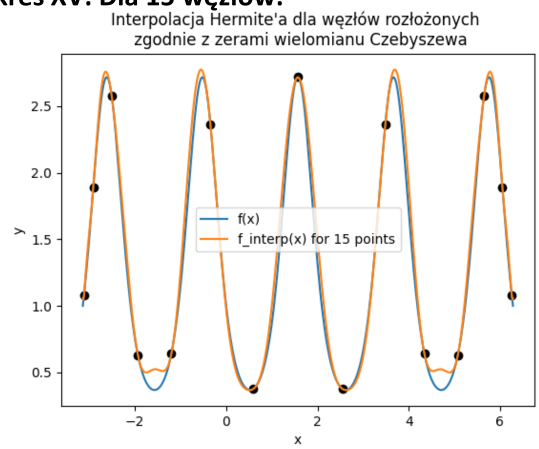
Wykres XIII: Dla 9 węzłów:



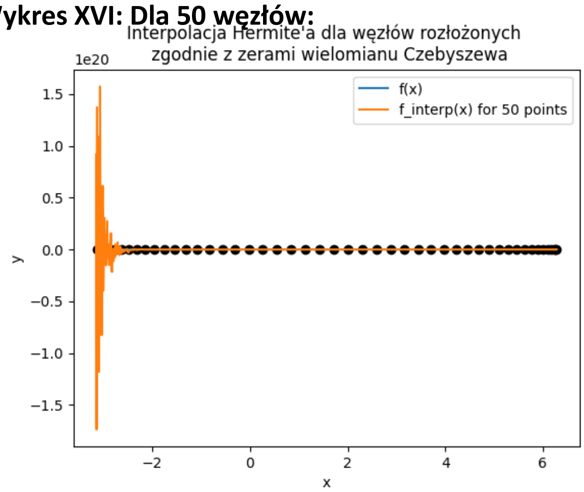
Wykres XIV: Dla 13 węzłów:



Wykres XV: Dla 15 węzłów:



Wykres XVI: Dla 50 węzłów:



Wykres XVII: Dla 80 węzłów:

Liczba węzłów	max(|f(x_interep)-y_interep|)	wzór IV
5	3,965718121	0,107007
7	6,834511312	0,110174
9	15,58387938	0,208182
11	146,1762675	1,674404
13	330,2503907	3,365531
15	1609,653798	15,04516
50	1,1061E+21	3,77E+18
80	8,9281E+41	2,58E+39

Tabelka I: Wartości obliczone dla węzłów rozłożonych równomiernie.

*wyniki zaokrąglone do 9 i 6 cyfr po przecinku

Liczba węzłów	max(|f(x_interep)-y_interep|)	wzór IV
5	2,564390867	0,046923
7	1,460782594	0,031125
9	0,553330911	0,01146
11	0,900019594	0,012478
13	0,76507954	0,010645
15	0,212927279	0,003551
50	1,73752E+20	9,44E+17
80	2,50412E+46	1,05E+44

Tabelka II: Wartości obliczone dla węzłów wyznaczonych zgodnie z zerami wielomianu Czebyszewa

*wyniki zaokrąglone do 9 i 6 cyfr po przecinku

7. Wnioski:

Wniosek dla interpolacji Hermite'a dla równoodległych węzłów:

- Interpolacja Hermite'a z równoodległymi węzłami może prowadzić do nieefektywnej interpolacji, ponieważ interpolacja ta zazwyczaj prowadzi do efektu Rungego, czyli oscylacji w okolicach końców przedziału interpolacji (co można zauważyć na wykresach od II do IX) .
- Interpolacja Hermite'a dla równoodległych węzłów jest bardzo dokładną metodą interpolacji funkcji gładkich, ale może wymagać znacznie więcej obliczeń niż inne metody interpolacji, takie jak metoda Lagrange'a.
- Z tabelki 1 można zauważyć, że raz ze wzrostem liczby węzłów do około 15/20 maksymalna wartość odstająca wzrasta.

Wniosek dla interpolacji Hermite'a dla węzłów Czebyszewa:

- Interpolacja Hermite'a z węzłami Czebyszewa jest preferowana do interpolacji funkcji w porównaniu do równoodległych węzłów Czebyszewa, ponieważ węzły te są lepiej rozłożone na przedziale interpolacji i minimalizują błąd interpolacji.
- Im większa liczba węzłów Czebyszewa, tym lepiej interpolowana funkcja, ponieważ węzły są bardziej gęsto rozłożone, co prowadzi do mniejszych błędów interpolacji.
- Interpolacja Hermite'a z węzłami Czebyszewa jest szczególnie przydatna dla interpolacji funkcji, które wykazują duże zmiany wartości na krańcach przedziału interpolacji.

Ogólne wnioski dla interpolacji Hermite'a:

- Interpolacja Hermite'a jest skuteczna w przypadkach, gdy znane są pochodne interpolowanej funkcji w wybranych punktach.

- Interpolacja Hermite'a pozwala na dokładniejszą interpolację niż interpolacja Lagrange'a, ponieważ interpolacja Hermite'a uwzględnia nie tylko wartości funkcji w punktach interpolacji, ale także ich pochodne.
- Jednakże, interpolacja Hermite'a może prowadzić do oscylacji w okolicach krańców przedziału interpolacji, co wymaga zastosowania dodatkowych technik interpolacji, takich jak Czebyszewa.
- W przypadku węzłów Czebyszewa, interpolacja Hermite'a jest bardziej efektywna numerycznie, ponieważ te węzły są rozmieszczone w sposób, który minimalizuje błędy interpolacji.
- Interpolacja Hermite'a wykorzystuje pochodne funkcji interpolowanej, co pozwala uzyskać bardziej gładką funkcję interpolującą.

8. Bibliografia:

- https://e.kul.pl/files/10382/public/aan_w4_1920.pdf
- Wykłady
- https://home.agh.edu.pl/~byrska/src/MN_2020/3_Interpolacja.pdf
- https://pythonnumericalmethods.berkeley.edu/notebooks/chapter17.05-NewtonsPolynomial-Interpolation.html