**Zagadnienie Hermite’a**

**Krystian Madej, 03.04.2024**

**1. Treść zadania**

Dla funkcji , , wyznacz dla zagadnienia Hermitea’a wielomian interpolujący,  
na przedziale [].  
Interpolację przeprowadź dla różnej liczby węzłów. Dla każdego przypadku interpolacji porównaj wyniki otrzymane dla różnego rozmieszczenia węzłów:  
równoodległe oraz Czebyszewa (zera wielomianu Czebyszewa).  
Oceń dokładność, z jaką wielomian przybliża zadaną funkcję.  
Poszukaj wielomianu, który najlepiej przybliża zadaną funkcję  
Wyszukaj stopień wielomianu, dla którego można zauważyć efekt Runge’go (dla równomiernego rozmieszczenia węzłów). Porównaj z wyznaczonym wielomianem dla węzłów Czebyszewa.

**2. Środowisko obliczeń**

**­**Obliczenia zostały wykonane przy pomocy języka **C++20** na systemie **Windows 11**, kompilacja 22631.3296, procesorze  
**64-bitowym** Intel Core i5-11400H 2.70GHz, kod kompilowany kompilatorem **MSVC** (wersja 19.39).

**3. Użyte biblioteki i programy pomocnicze**

Do instalacji bibliotek **C++** użyto programu **conan**, wersja 2.1.

Najważniejsze użyte biblioteki:

* <format> - łatwe formatowanie
* <numbers> - stałe matematyczne
* CvPlot – tworzenie wykresów
* SymEngine – obliczenia symboliczne
* <future> - obiekty std::future oraz std::async
* <ranges> - operacje na obiektach iterowalnych

**4. Sposób obliczeń**

**4.1. Metoda Hermite’a**

Mając węzłów i ich kolejne krotności  
szukamy wielomianu , stopnia , takiego że:

Krotność węzła mówi, ile kolejnych pochodnych ma być równych.

Każda liczba da się przedstawić jako , gdzie oraz

Zdefiniujmy wielomiany:

,

gdziie   
Wtedy wielomian ma postać:

Współczynniki znajdujemy budując tablicę ilorazów różnicowych, podobną do tej w metodzie Newtona. Tam gdzie nie da się utworzyć ilorazu, korzystamy z pochodnych.

Przykład dla ( są zaznaczone na czerwono):

**4.2 Ocena dokładności**

Dokładność interpolacji można ocenić porównując następujące wartości:

* Błąd bezwględny
* Błąd maksymalny
* Suma kwadratów

**5.1. Implementacja obliczeń**

Wszystkie obliczenia były wykonywane przy użyciu  
64-bitowego typu zmiennoprzecinkowego **double** (w kodzie zaliasowany jako **flt**)

W celu wygenerowania węzłów użyto funkcji generujących, zaimplementowanych w poprzednim ćwiczeniu:

* nodes::uniform (węzły równoodległe), korzystającą ze wzoru
* nodes::chebyshev (węzły czebyszewa), korzystającą ze wzoru

Następnie zaimplementowano funkcję interpolation::\_hermite, wykonującą interpolację metodą Hermite’a. Przyjmuje dwuwymiarową tablicę wartości funkcji interpolowanej i tablicę z odpowiadającymi im węzłami, a zwraca obiekt wywoływalny, odpowiadający wielomianowi interpolacyjnemu. W tablicy dwuwymiarowej wiersze odpowiadają kolejnym węzłom, natomiast kolumny kolejnym pochodnym  
(wiersz -ty zawiera wartości ). Istnieje też funkcja interpolation::hermite, która przyjmuje funkcję interpolowaną w postaci SymEngine::Expression, liczbę pochodnych do wykorzystania, funkcję generującą węzły, ilość węzłów oraz przedział do interpolacji. Generuje ona dane na podstawie przekazanych argumentów, przekazując te dane do interpolation::\_hermite.  
Funkcja interpolation::\_hermite korzysta z funkcji interpolation::precompute\_hermite, przyjmującą dwuwymiarową tablicę wartości funkcji interpolowanej i tablicę z odpowiadającymi im węzłami, zwracającą krotkę zawierającą tablicę współczynników , tablicę krotności   
oraz sumę .

Pochodne są obliczane iteracyjnie, metodą SymEngine::Expression::diff. Zgodnie z poleceniem ustnym obliczam tylko pochodną pierwszego stopnia.

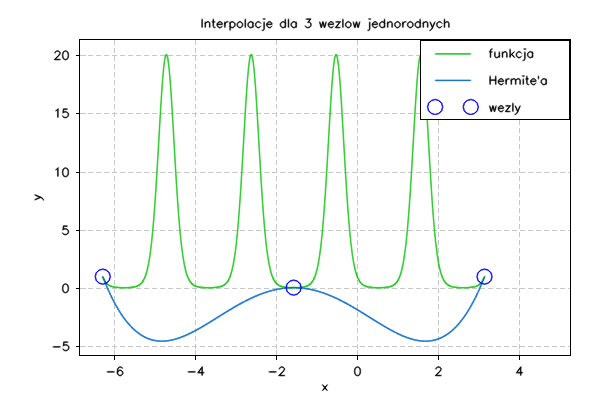
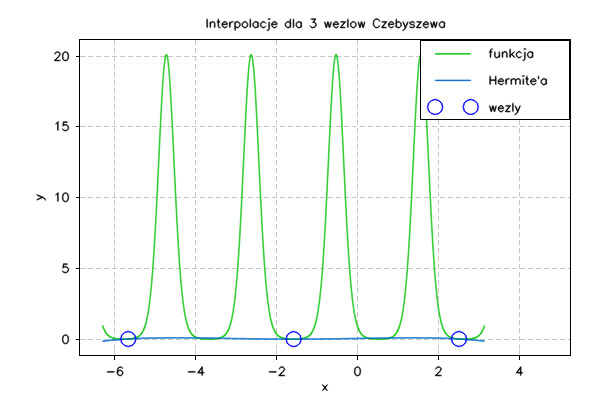
Użyto też funkcje obliczające błędy interpolacji, zaimplementowanych w poprzednim ćwiczeniu:

* error::abs – przyjmująca funkcję interpolowaną, wielomian interpolujący i tablicę węzłów, zwracająca tablicę błędów bezwzględnych
* error::max – w 2 wersjach: pierwsza, przyjmująca funkcję interpolowaną, wielomian interpolujący i tablicę węzłów i obliczająca wcześniej błąd bezwzględny, i druga, przyjmująca tablicę błędów bezwzględnych, obie zwracają wartość maksymalną z tablicy błędów bezwzględnych
* error::sum\_squared – w 2 wersjach: pierwsza, przyjmująca funkcję interpolowaną, wielomian interpolujący i tablicę węzłów i obliczająca wcześniej błąd bezwzględny, i druga, przyjmująca tablicę błędów bezwzględnych, obie zwracają sumę kwadratów błędów bezwzględnych

W funkcji main prowadzone są obliczenia dla kolejnych ilości węzłów.   
Wartości funkcji interpolowanej, jak i interpolacji są zapisywane w plikach  
interpolation\_results/result\_<ilość węzłów>.txt  
Wartości błędów bezwzględnych w plikach interpolation\_results/error\_<ilość węzłów>.txt   
Wartości maksymalne błędów bezwzględnych w pliku interpolation\_results/max\_abs.txt   
Sumy kwadratów błędów bezwzględnych w pliku interpolation\_results/sum\_squared.txt

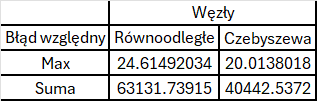
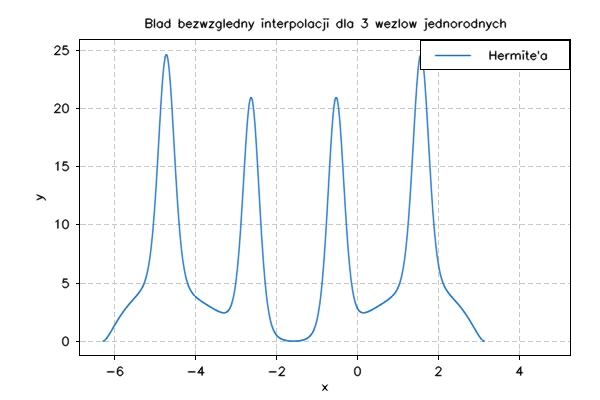
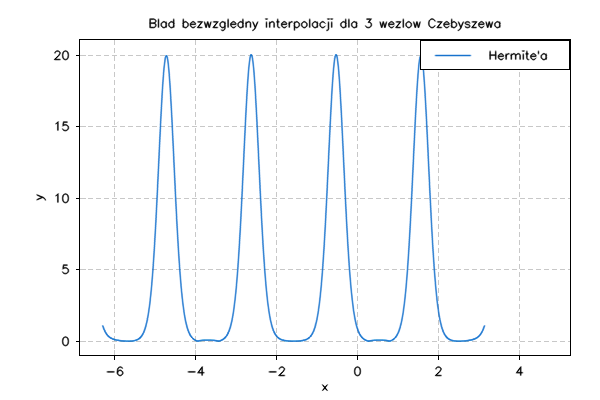
W folderze interpolation\_images/ są zapisywane wykresy funkcji interpolowanej, interpolacji i wartości błędów.

**6. Wyniki obliczeń**

**6.1 Dla 3 węzłów**

Wykres 2.

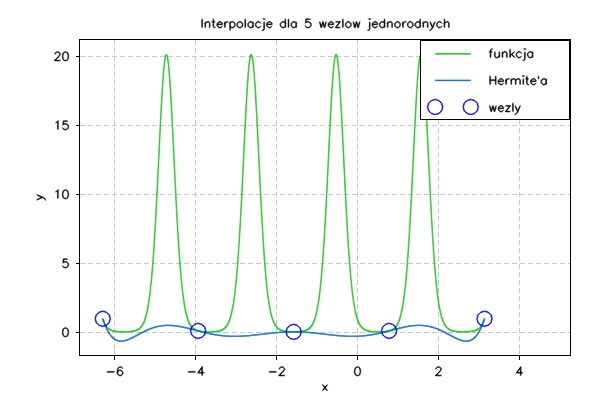
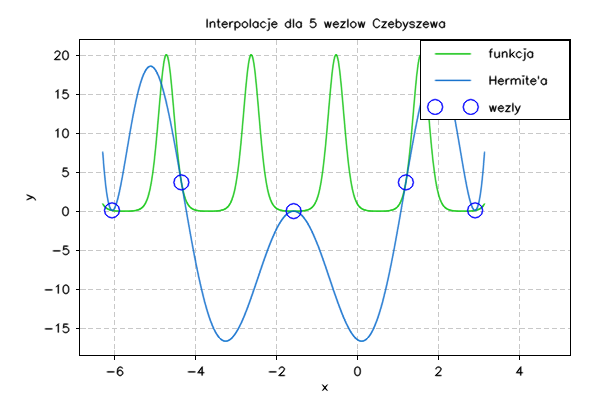
Wykres 1.

****

Wykres 4.

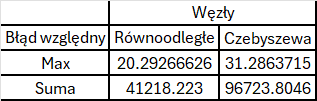
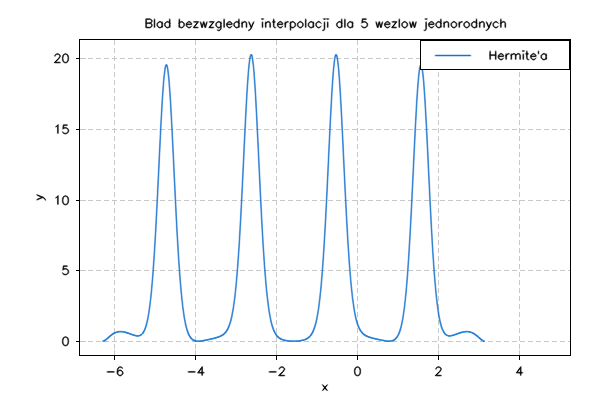
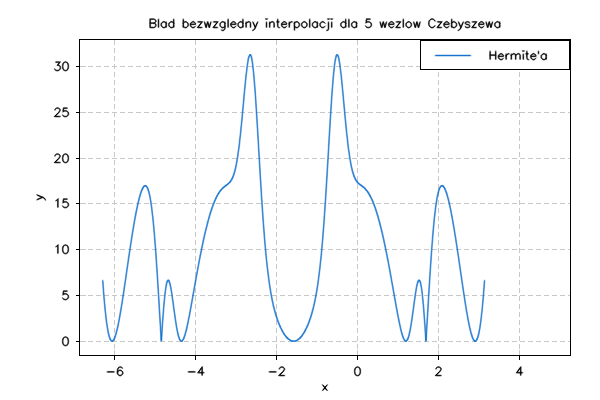
Wykres 3.

Tabela 1.  
Błędy interpolacji dla 3 węzłów

**6.2 Dla 5 węzłów**

Wykres 6.

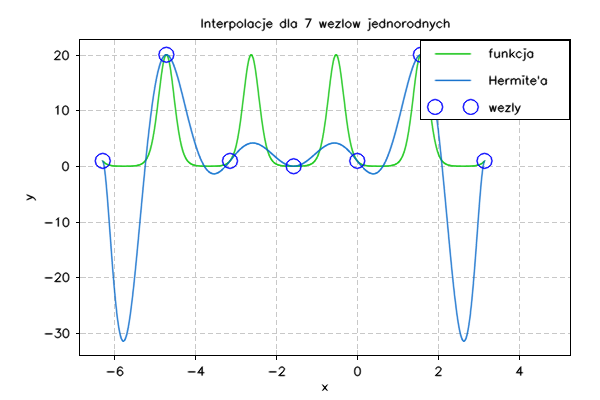
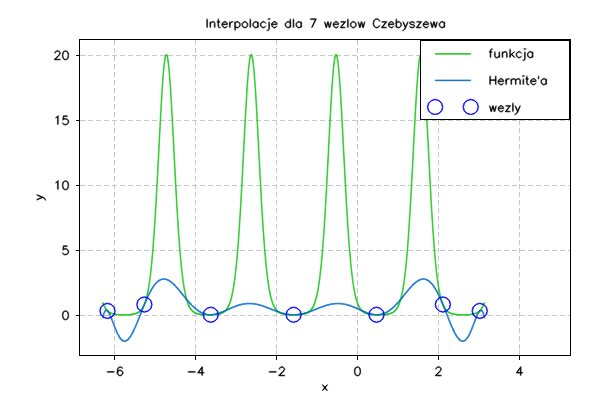
Wykres 5.

****

Wykres 8.

Wykres 7.

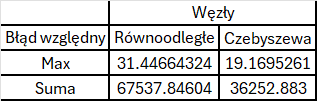
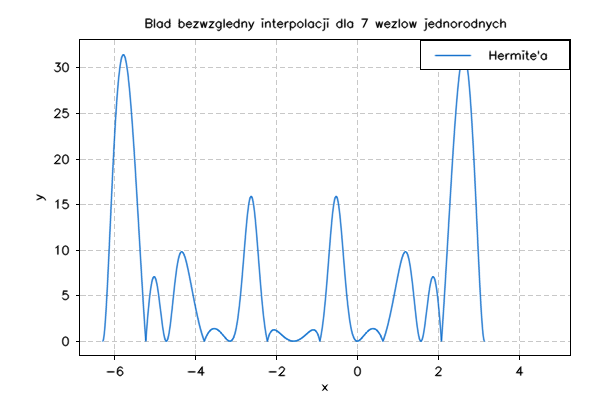
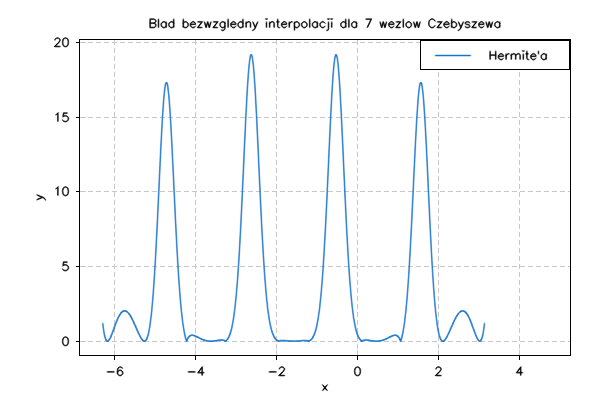
Tabela 2.  
Błędy interpolacji dla 5 węzłów

**6.3 Dla 7 węzłów**

Wykres 10.

Wykres 9.

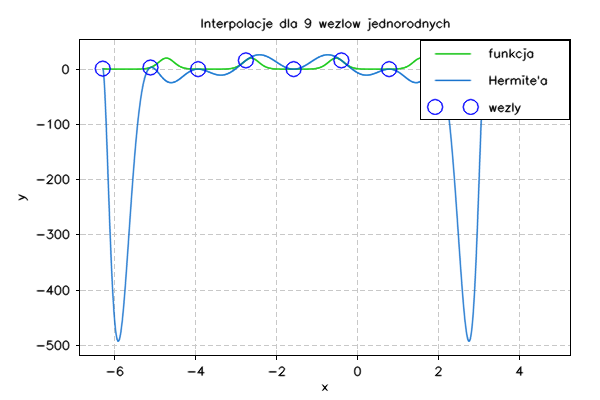
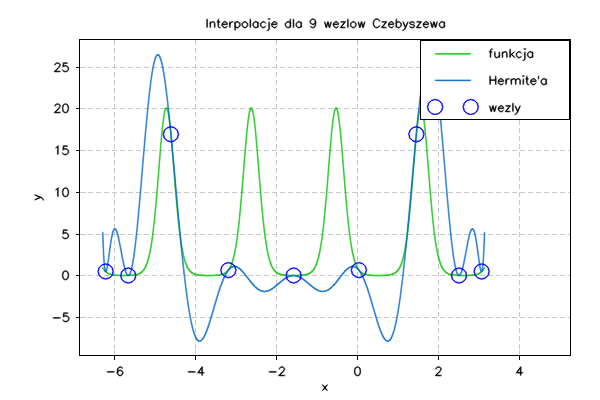
Dla 7 równoodległych węzłów zaczyna pojawiać się względnie niewielki efekt Runge’go.

****

Wykres 12.

Wykres 11.

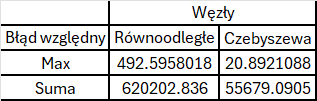
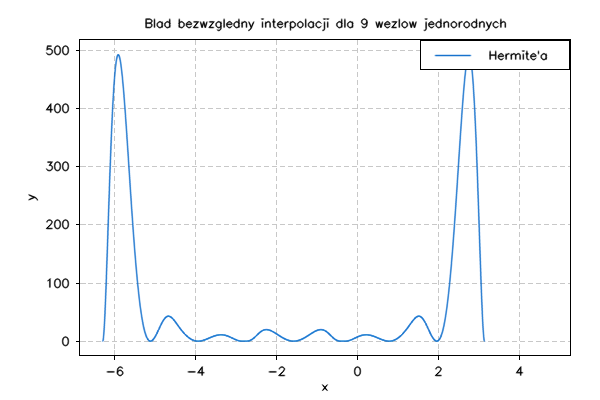
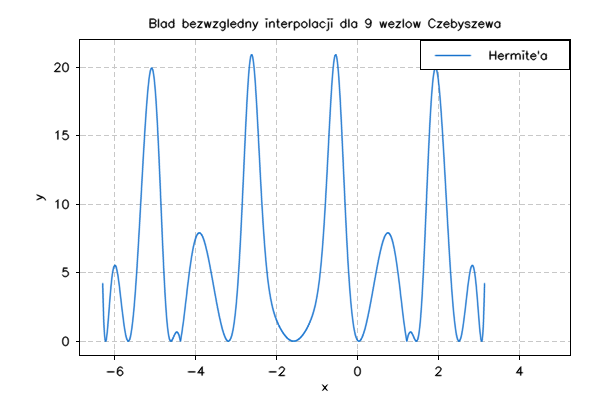
Tabela 3.  
Błędy interpolacji dla 7 węzłów

**6.4 Dla 9 węzłów**

Wykres 14.

Wykres 13.

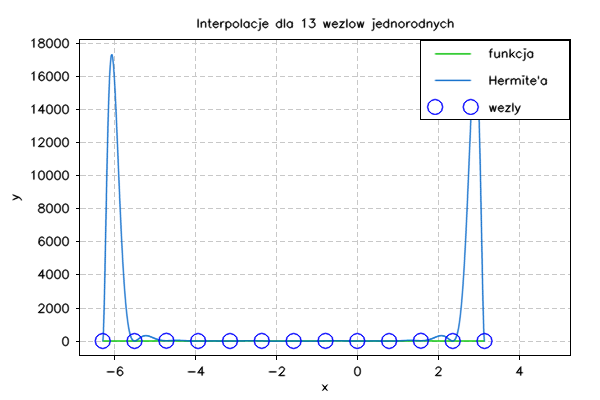
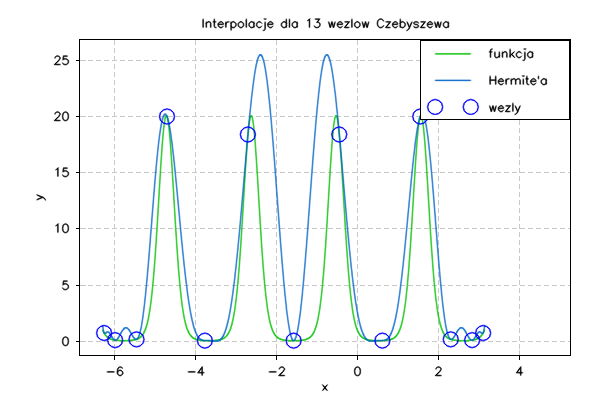
Dla 9 węzłów zaczyna być widoczny duży efekt Runge’go na krańcach przedziału interpolacji, tylko dla równoodległych węzłów.

****

Wykres 16.

Wykres 15.

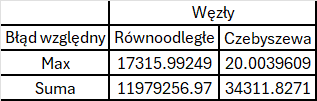
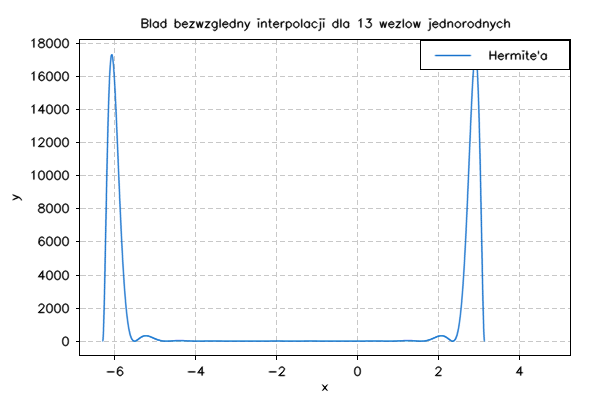
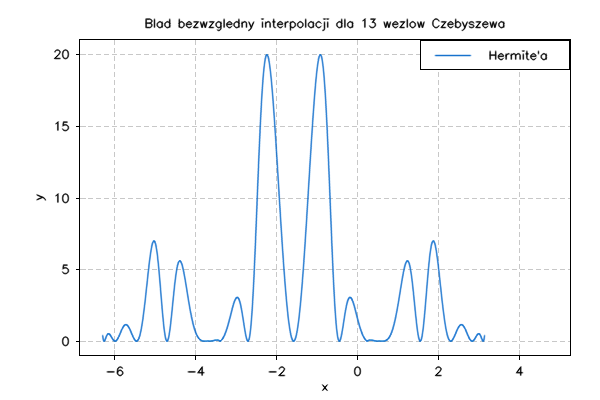
Tabela 4.  
Błędy interpolacji dla 9 węzłów

**6.5 Dla 13 węzłów**

Wykres 18.

Wykres 17.

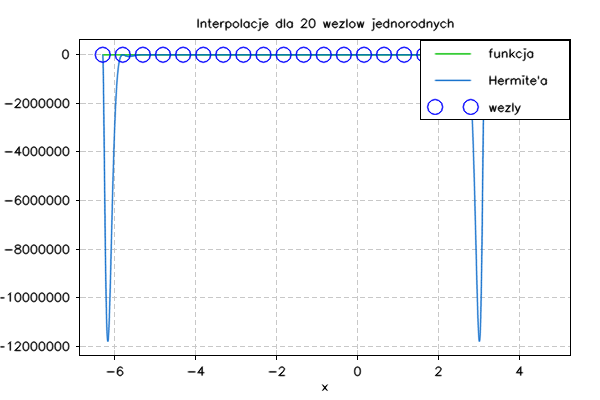
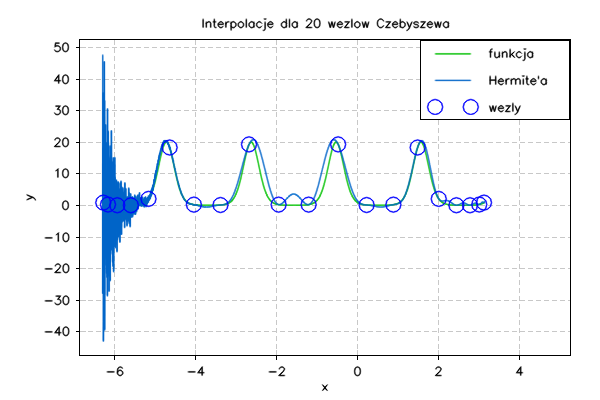
Dla 13 widoczny jeszcze większy efekt Runge’go na krańcach przedziału interpolacji, tylko dla równoodległych węzłów.

****

Wykres 20.

Wykres 19.

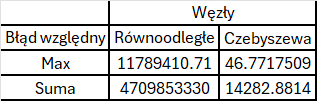
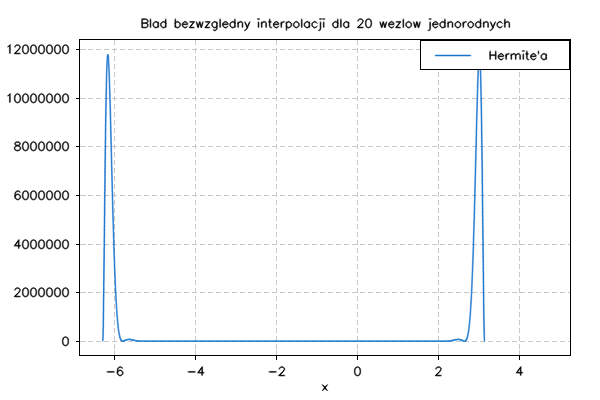
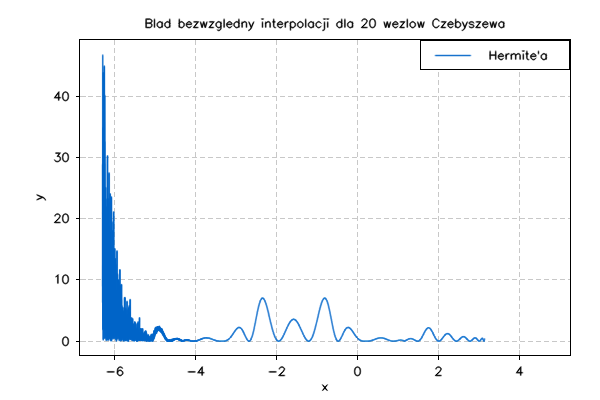
Tabela 5.  
Błędy interpolacji dla 13 węzłów

**6.6 Dla 20 węzłów**

Wykres 22.

Wykres 21.

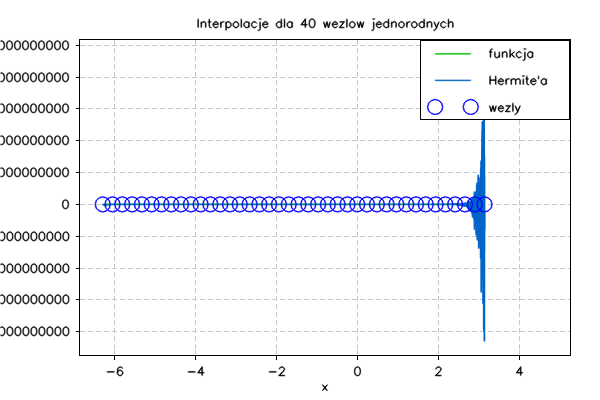
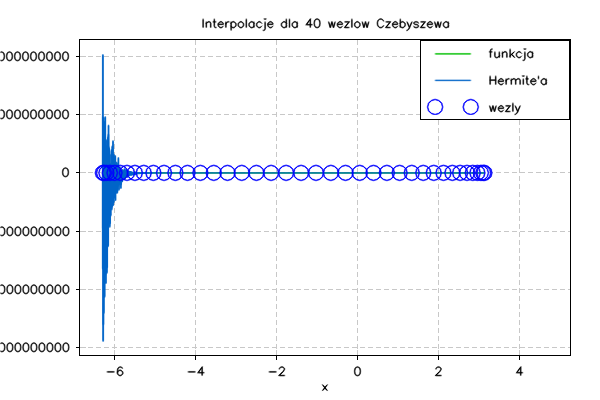
Dla 20 węzłów widoczny ogromny efekt Runge’go na krańcach przedziału interpolacji, tylko dla równoodległych węzłów. Pomijając błędy numeryczne na początku przedziału, interpolacja węzłami Czebyszewa jest dość dokładna.

****

Wykres 24.

Wykres 23.

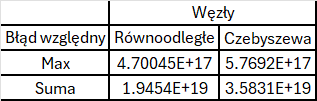
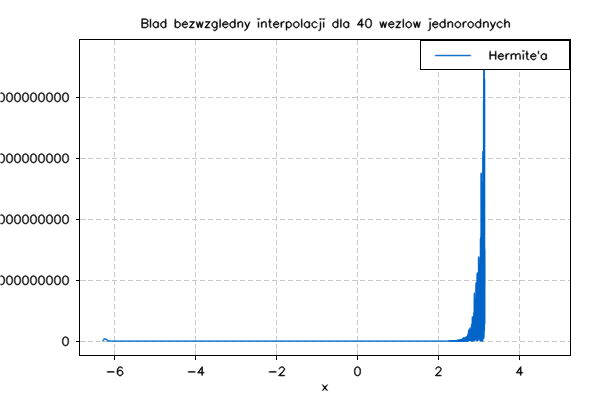
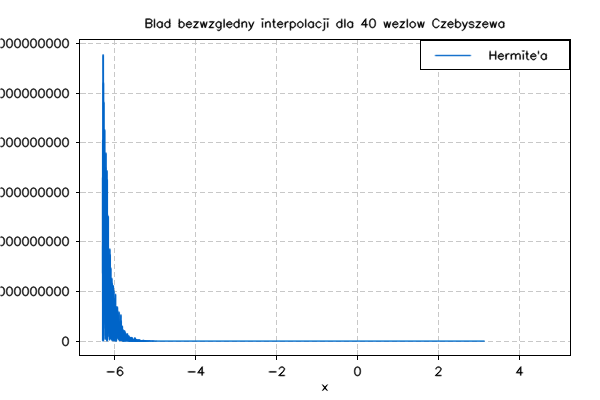
Tabela 6.  
Błędy interpolacji dla 20 węzłów

**6.7 Dla 40 węzłów**

Wykres 26.

Wykres 25.

Dla 40 węzłów w obu przypadkacyh widoczne ogromne błędy numeryczne. Ze względu na ograniczenia biblioteki CvPlot, nie udało się wyrenderować wartości na osi Y. Z pliku interpolation\_results/max\_abs.txt można jednak odczytać, że te wartości są rzędu **.**

****

Wykres 28.

Wykres 27.

Tabela 7.  
Błędy interpolacji dla 40 węzłów

**6.8. Najlepiej przybliżający wielomian**

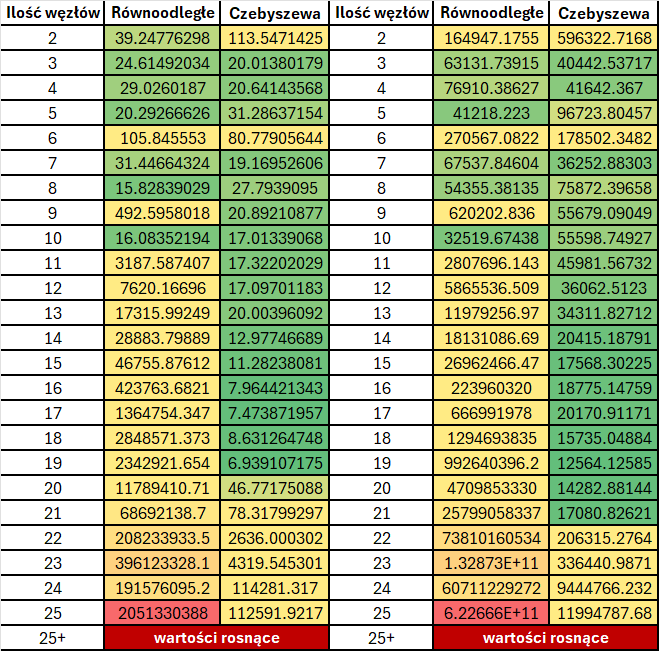
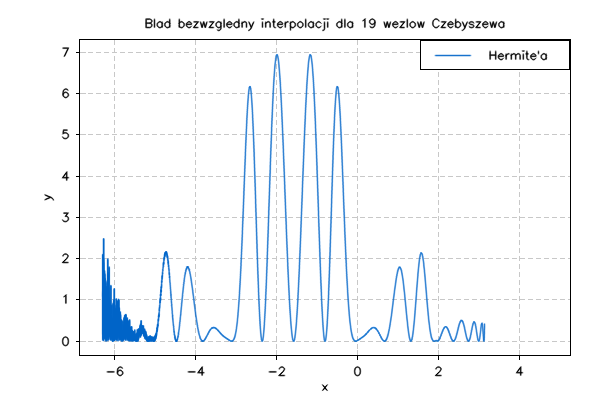
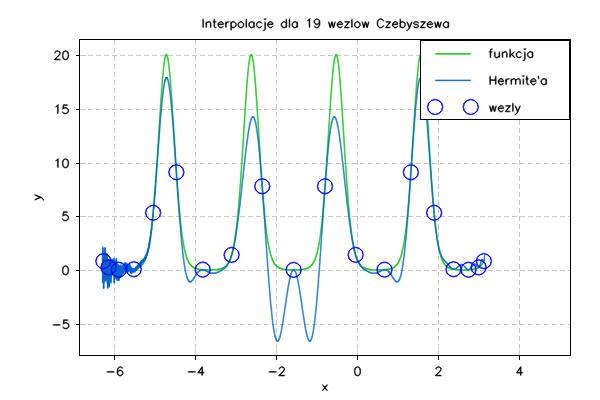
Aby znaleźć najlepiej przybliżający wielomian, należy znaleźć wielomian którego błąd maksymalny i suma błędów są najmniejsze.

Tabela 8. Tabela 9.  
 Maksymalne błędy interpolacji Sumy błędów interpolacji

Jak łatwo zauważyć, najmniejszą wartość maksymalnego błędu bezwzględnego jak i najmniejszą sumę błędów bezwzględnych, równe kolejno i , uzyskano przy wykorzystaniu 19 węzłów Czebyszewa. W powyższych tabelach pominąłem wartości dla liczby węzłów  
, gdyż były one większe i rosnące od widocznych tabeli. Powodem w obu przypadkach są błędy numeryczne, które w przypadku węzłów równoodległych były większe od błędów związanych z efektem Runge’go.

Niestety i ów najlepszy wielomian nie jest wolny od błędów numerycznych, są one jednak niewielkie w porównaniu z innymi wielomianami.

**7. Wnioski**

Wykres 30.

Wykres 29.

Oczywistym jest, że wielomiany interpolacyjne z równoodległymi węzłami nie mogą być dokładne. Nie powinno się ich używać, ponieważ występuje efekt Runge’go, a dla wyższych stopni efekty numeryczne.  
Można się uchronić od efektu Runge’go, zagęszczając węzły na krańcach przedziału interpolacji, np. przyjmując za nie zera wielomianów Czebyszewa.

Pomimo użycia węzłów Czebyszewa, od ilości 18 węzłów pojawiają się błędy numeryczne na początku przedziału interpolacji i zwiększają się wraz z ilością węzłów.

Najmniejsze błędy występowały dla 19 węzłów Czebyszewa.