**Zagadnienie Lagrange’a**

**Krystian Madej, 21.03.2024**

**1. Treść zadania**

Dla funkcji , , wyznacz dla zagadnienia Lagrange’a wielomian interpolujący w postaci Lagrange’a i Newtona, 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, procesor **64-bitowy** Intel Core i5-11400H 2.70GHz, kod kompilowany kompilatorem **MSVC** (wersja 19.38).

**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
* <future> - obiekty future oraz std::async
* <ranges> - operacje na obiektach iterowalnych

**4. Sposób obliczeń**

**4.1. Metoda Lagrange’a**

Metoda Lagrange’a polega na obliczeniu sumy danej wzorem:  
gdzie:

* Baza Lagrange’a
* - kty węzeł

**4.2 Metoda Newtona (ilorazów różnicowych)**

Metoda Newtona polega na obliczeniu sumy danej wzorem:

gdzie:

* iloraz różnicowy
* – ity węzeł

**4.3 Metoda Newtona dla równoodległych węzłów**

Metoda ta wynika z przekształceń zwykłej metody Newtona.

Róznica progresywna:  
Dla węzłów równoodległych zachodzi:  
Podstawiając mamy:

**4.4 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 zaimplementowano funkcje generujące:

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

Następnie zaimplementowano funkcję interpolation::\_lagrange, wykonującą interpolację metodą Lagrange’a. Przyjmuje tablicę wartości funkcji interpolowanej i tablicę z odpowiadającymi im węzłami, a zwraca obiekt wywoływalny, odpowiadający wielomianowi interpolacyjnemu. Istnieje też funkcja interpolation::lagrange, która przyjmuje funkcję interpolowaną, funkcję generującą węzły, ilość węzłów oraz przedział do interpolacji. Generuje ona dane na podstawie przekazanych argumentów, przekazując te dane do interpolation::\_lagrange.

Analogicznie zaimplementowano metodę Newtona. Istnieją funkcje interpolation::newton i interpolation::\_newton, realizujące przypadek ogólny, oraz interpolation::\_newton\_uniform i interpolation::newton\_uniform, realizujące przypadek z równoodległymi węzłami.  
Dodatkowo, jeżeli do funkcji interpolation::newton przekażemy nodes::uniform, tak naprawdę wywoła się interpolation::newton\_uniform.

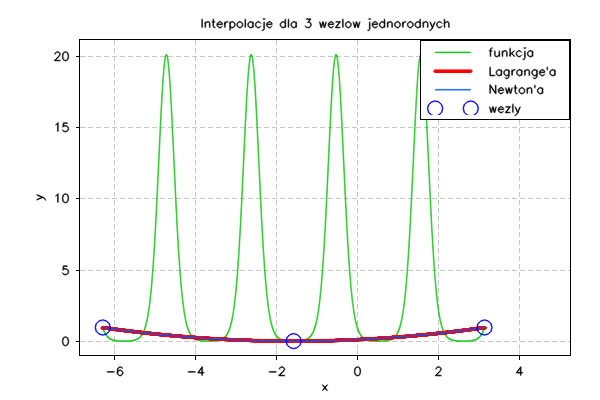
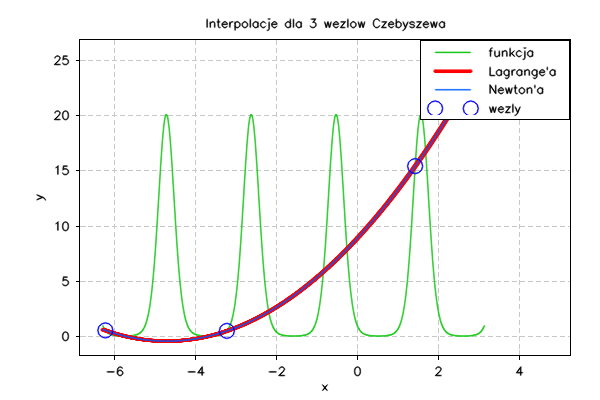
Zaimplementowano też funkcje obliczające błędy interpolacji:

* 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

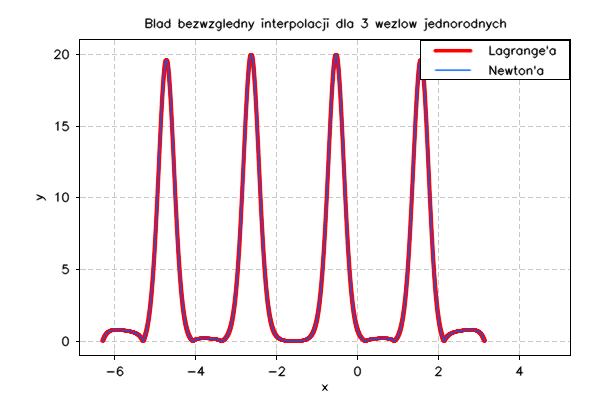
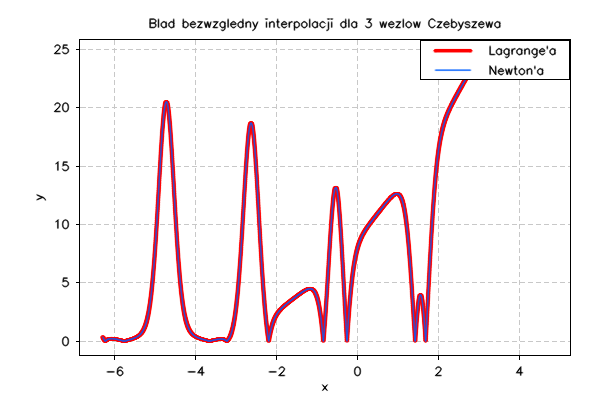
W folderze interpolation\_images/ są zapisywane wykresy funkcji interpolowanej, interpolacji i wartości błędów.

**6. Wyniki obliczeń**

**6.1 Dla 3 węzłów**

Wykres 2.

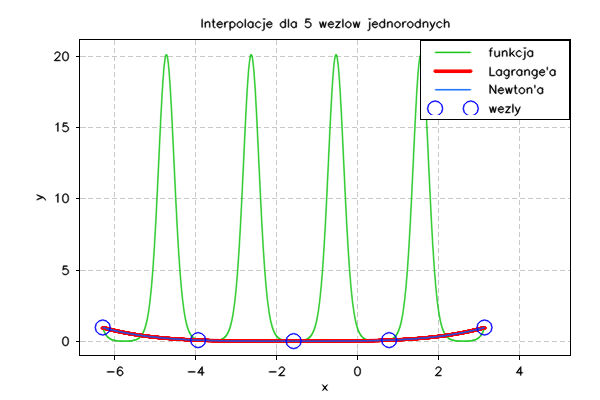
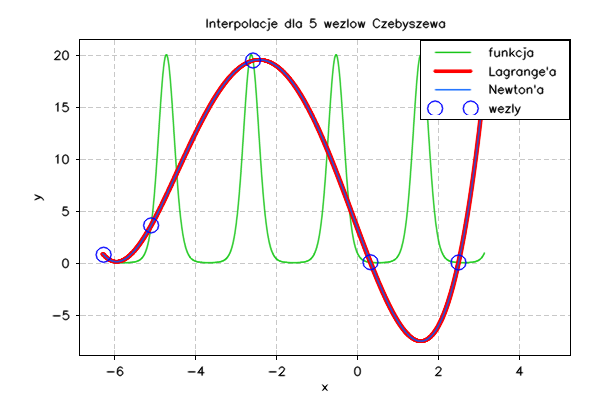
Wykres 1.

****

Wykres 4.

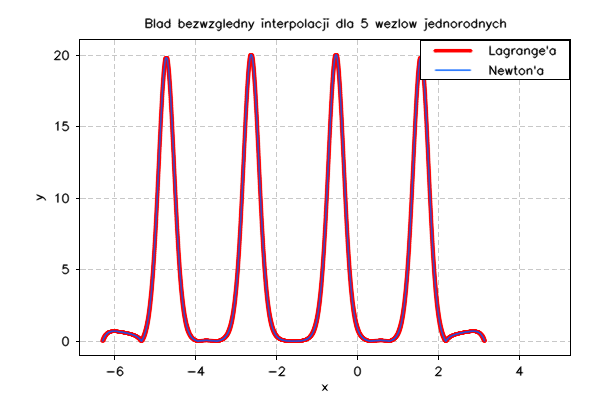
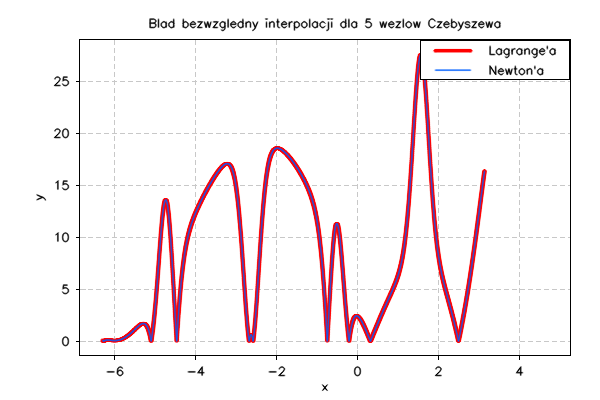
Wykres 3.

Tabela 1.  
Błędy interpolacji dla 3 węzłów

**6.2 Dla 5 węzłów**

Wykres 6.

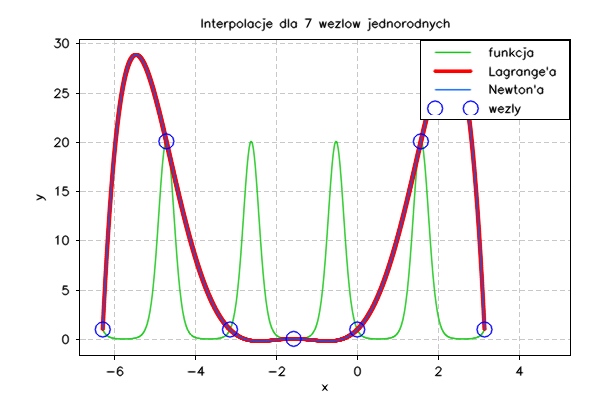
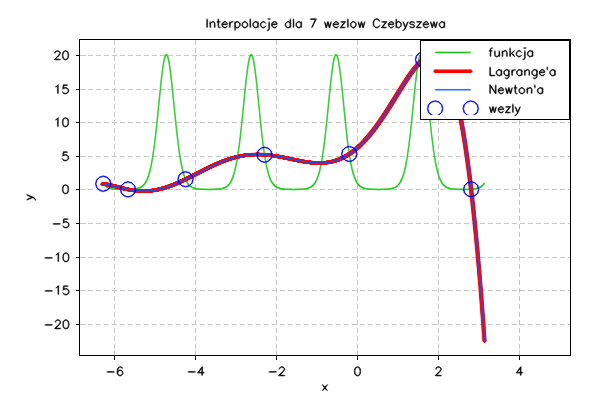
Wykres 5.

****

Wykres 8.

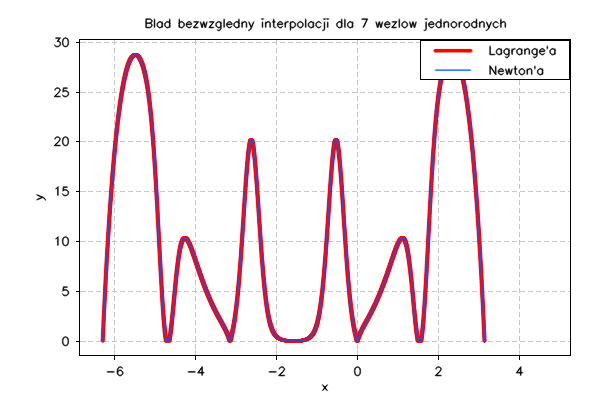
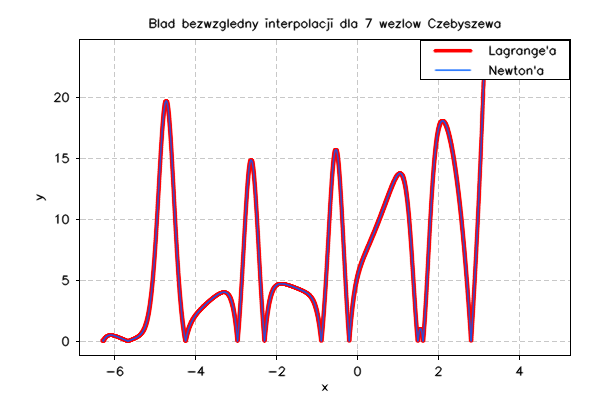
Wykres 7.

Tabela 2.  
Błędy interpolacji dla 5 węzłów

**6.3 Dla 7 węzłów**

Wykres 10.

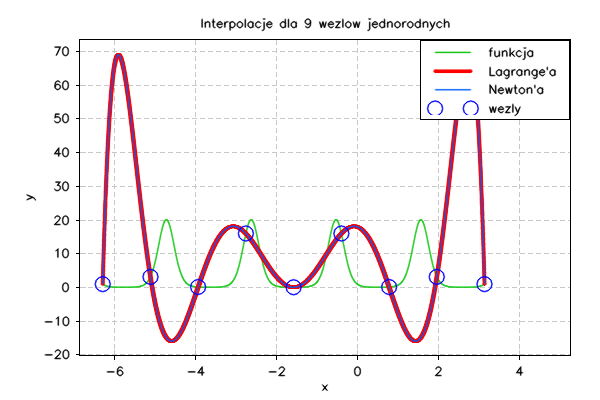
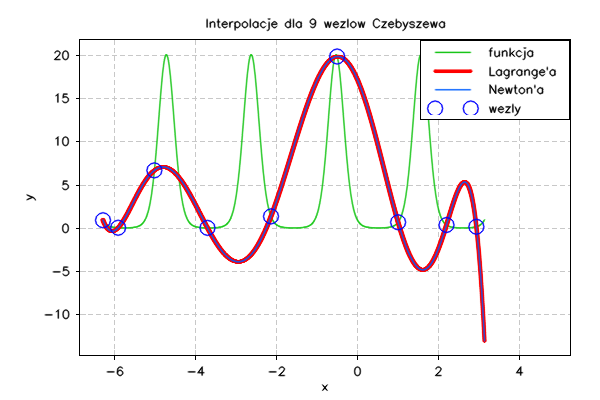
Wykres 9.

****

Wykres 12.

Wykres 11.

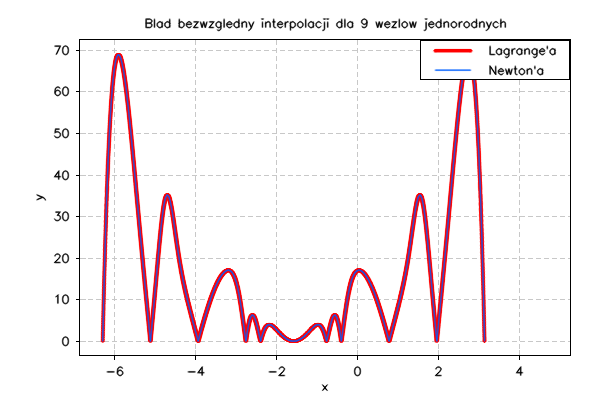
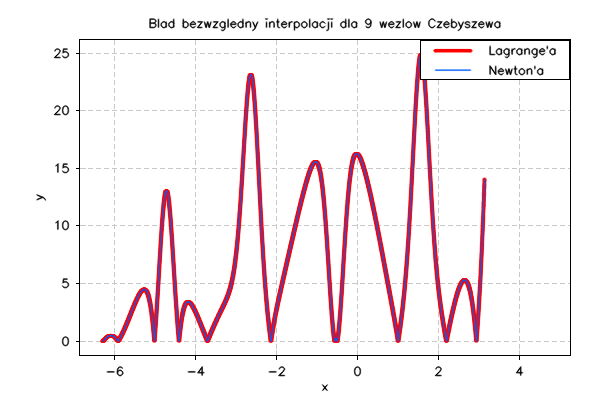
Tabela 3.  
Błędy interpolacji dla 7 węzłów

**6.4 Dla 9 węzłów**

Wykres 14.

Wykres 13.

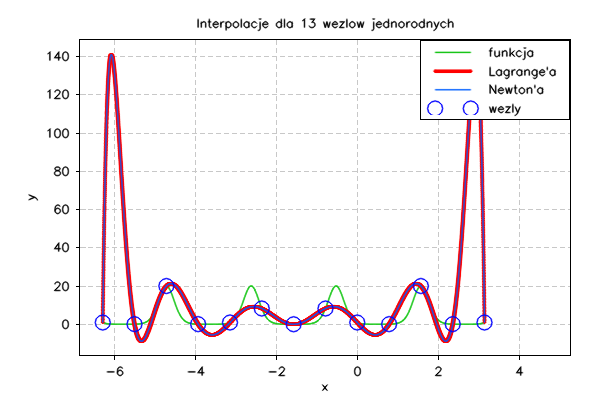
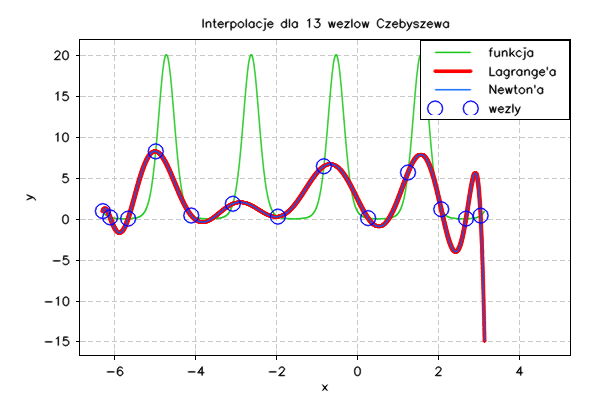
Dla 9 węzłów zaczyna być widoczny efekt Runge’go na krańcach przedziału interpolacji, tylko dla równoodległych węzłów.

****

Wykres 16.

Wykres 15.

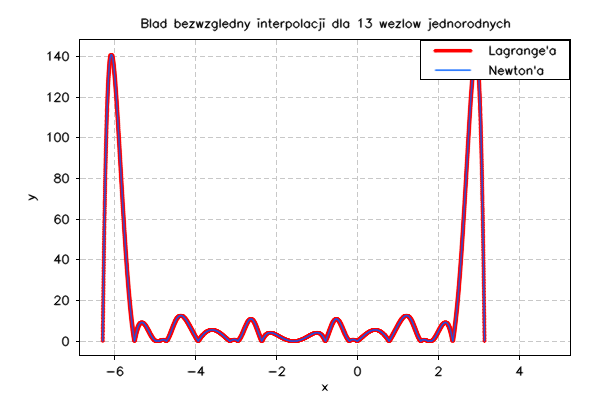
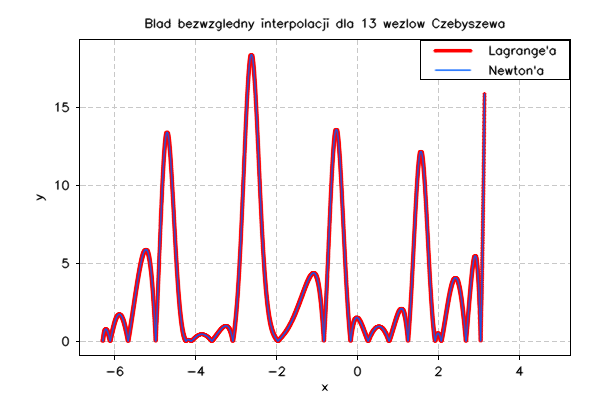
Tabela 4.  
Błędy interpolacji dla 9 węzłów

**6.5 Dla 13 węzłów**

Wykres 18.

Wykres 17.

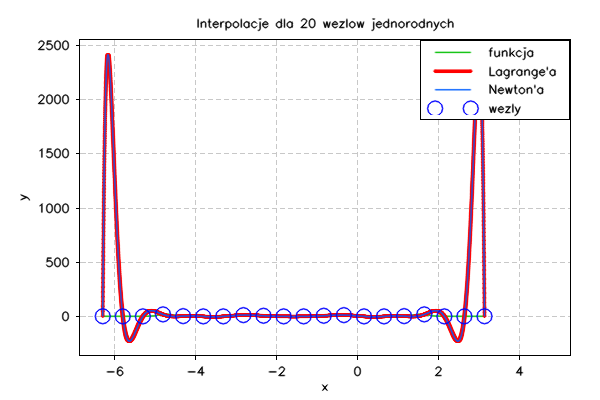
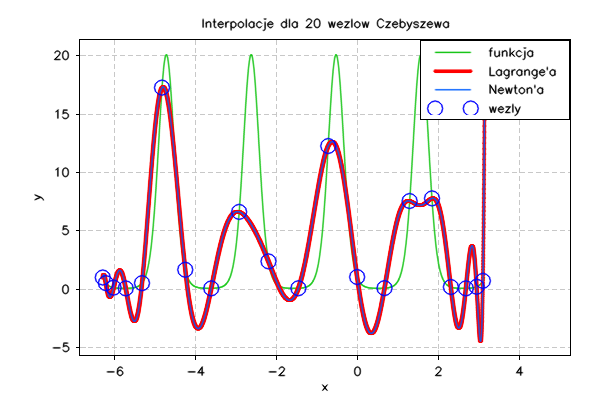
Dla 13 widoczny większy efekt Runge’go na krańcach przedziału interpolacji, tylko dla równoodległych węzłów.

****

Wykres 20.

Wykres 19.

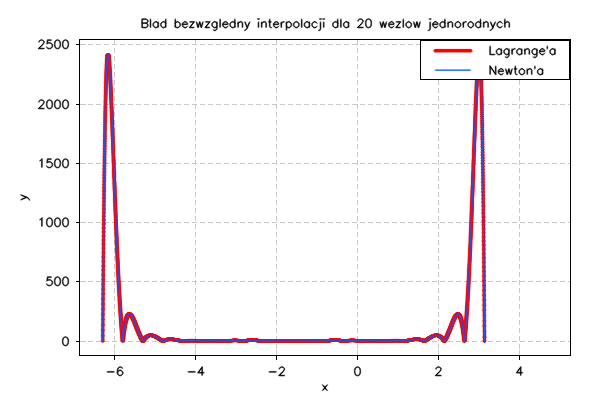
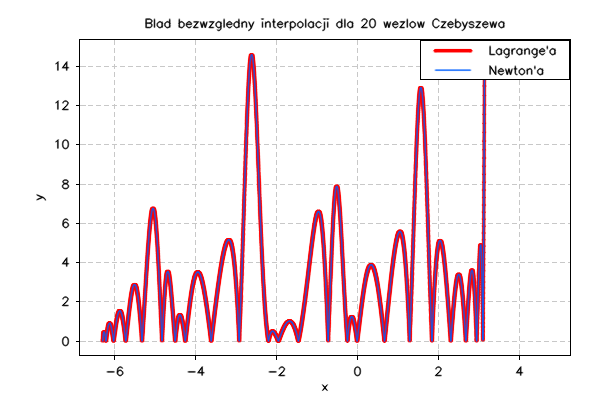
Tabela 5.  
Błędy interpolacji dla 13 węzłów

**6.6 Dla 20 węzłów**

Wykres 22.

Wykres 21.

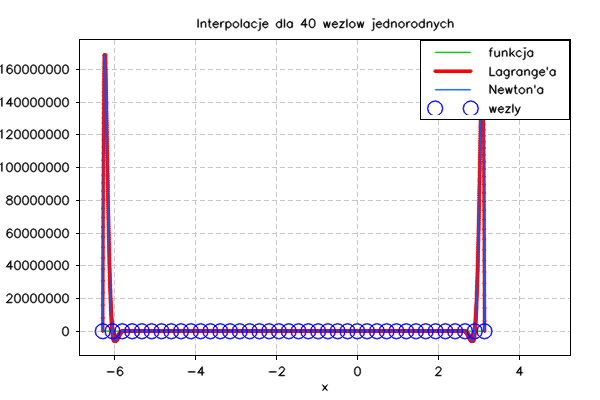
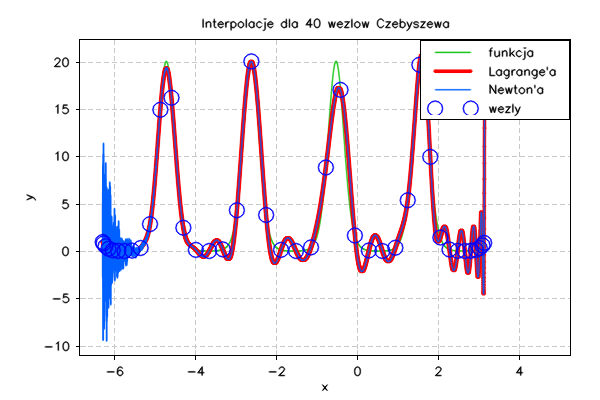
Dla 20 węzłów widoczny duży efekt Runge’go na krańcach przedziału interpolacji, tylko dla równoodległych węzłów.

****

Wykres 24.

Wykres 23.

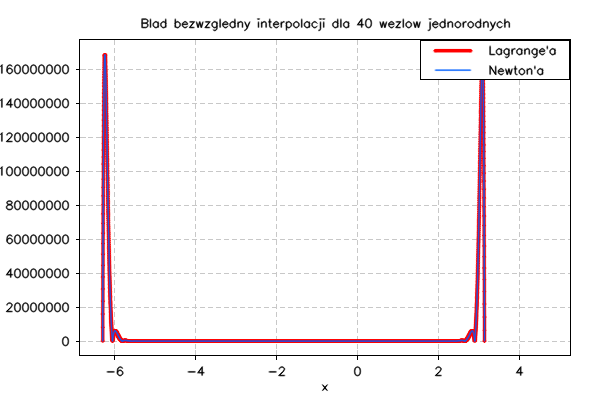
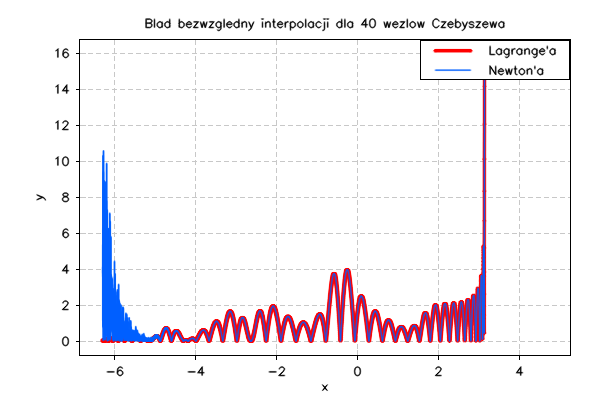
Tabela 6.  
Błędy interpolacji dla 20 węzłów

**6.7 Dla 40 węzłów**

Wykres 26.

Wykres 25.

Dla 40 węzłów widoczny ogromny efekt Runge’go na krańcach przedziału interpolacji, tylko dla równoodległych węzłów. Interpolacja Newtona dla węzłów Czebyszewa ma duże błędy (zapewne numeryczne) na początku przedziału interpolacji.

****

Wykres 28.

Wykres 27.

Tabela 7.  
Błędy interpolacji dla 40 węzłów

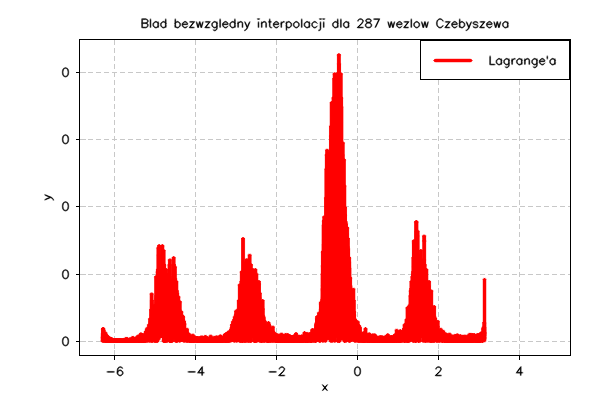
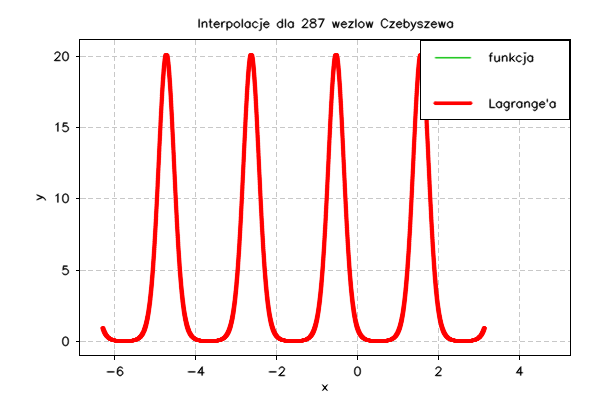
**6.8. Najlepiej przybliżający wielomian**

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, dla większej ilości węzłów tylko wielomian Lagrange’a z węzłami Czebyszewa daje lepsze przybliżenie. W celu zaoszczędzenia czasu, dalsze obliczenia będą prowadzone tylko dla tego wielomianu.  
Błędy liczono dla ilości węzłów od 2 do 500 włącznie.

Najmniejszą wartość maksymalnego błędu uzyskał wielomian z 287 węzłami , uzyskując równocześnie jedną z najmniejszych sum kwadratów błędów

**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.  
Można się od niego uchronić, zagęszczając węzły na krańcach przedziału interpolacji, np. przyjmując za nie zera wielomianów Czebyszewa.

Pomimo użycia węzłów Czebyszewa, interpolacja Newtona napotyka błędy numeryczne, gdy ich liczba jest w okolicach 40. Wielomian Lagrange’a dość długo pozostaje bez większych błędów.

W zadanym przypadku nie opłacałoby się używać wielomianu  
o liczbie węzłów >300, gdyż dokładność nie wzrasta dla większej ich ilości, przeciwnie, zaczyna spadać.