

Zagadnienie Hermite'a

Krystian Madej, 03.04.2024

1. Treść zadania

Dla funkcji $e^{-k \sin(mx)}$, $k = m = 3$, wyznacz dla zagadnienia Hermitea'a wielomian interpolujący, na przedziale $[a = -2\pi; b = \pi]$.

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 $k + 1$ węzłów x_0, x_1, \dots, x_k i ich kolejne krotności

$$m_0, m_1, \dots, m_k; \sum_{i=0}^k m_i = n + 1$$

szukamy wielomianu $H_n(x)$, stopnia $\leq n$, takiego że:

$$H_n^{(j)}(x_i) = f^{(j)}(x_i)$$

Krotność węzła mówi, ile kolejnych pochodnych ma być równych.

$$s(i) = \begin{cases} 0, & i = 0 \\ m_0 + m_1 + \dots + m_{i-1}, & i > 0 \end{cases}$$

Każda liczba $0 \leq l \leq n$ da się przedstawić jako $l = s(i) + j$, gdzie $0 \leq i \leq k$ oraz $0 \leq j \leq m_i - 1$

Zdefiniujmy wielomiany:

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

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

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

Wtedy wielomian $H_n(x)$ ma postać:

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

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

Przykład dla $m_1 = 3$ (b_l są zaznaczone na czerwono):

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	$f(x_1)$	$f'(x_1)$	$\frac{f''(x_1)}{2!}$	$f[x_0, x_1, x_1, x_1]$	
...
x_n	$f(x_n)$	$f[x_{n-1}, x_n]$	$f[x_0, \dots, x_n]$

4.2 Ocena dokładności

Dokładność interpolacji można ocenić porównując następujące wartości:

- Błąd bezwzględny
 $|f(x) - H_n(x)|$
- Błąd maksymalny
 $\max_i \{|f(x_i) - H_n(x_i)|\}$
- Suma kwadratów

$$\sum_{i=0}^n (f(x_i) - H_n(x_i))^2$$

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 $x_i = a + \frac{b-a}{n-1}i$
- `nodes::chebyshev` (węzły czebyszewa), korzystającą ze wzoru $x_i = \frac{a+b}{2} + \frac{b-a}{2} \cos\left(\frac{2j-1}{2k}\pi\right)$

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 i -ty zawiera wartości $f(x_i), f'(x_1), f''(x_1), \dots, f^{(j)}(x_i)$).

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 b_i , tablicę krotności m_i oraz sumę $\sum_{i=0}^k m_i$.

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`

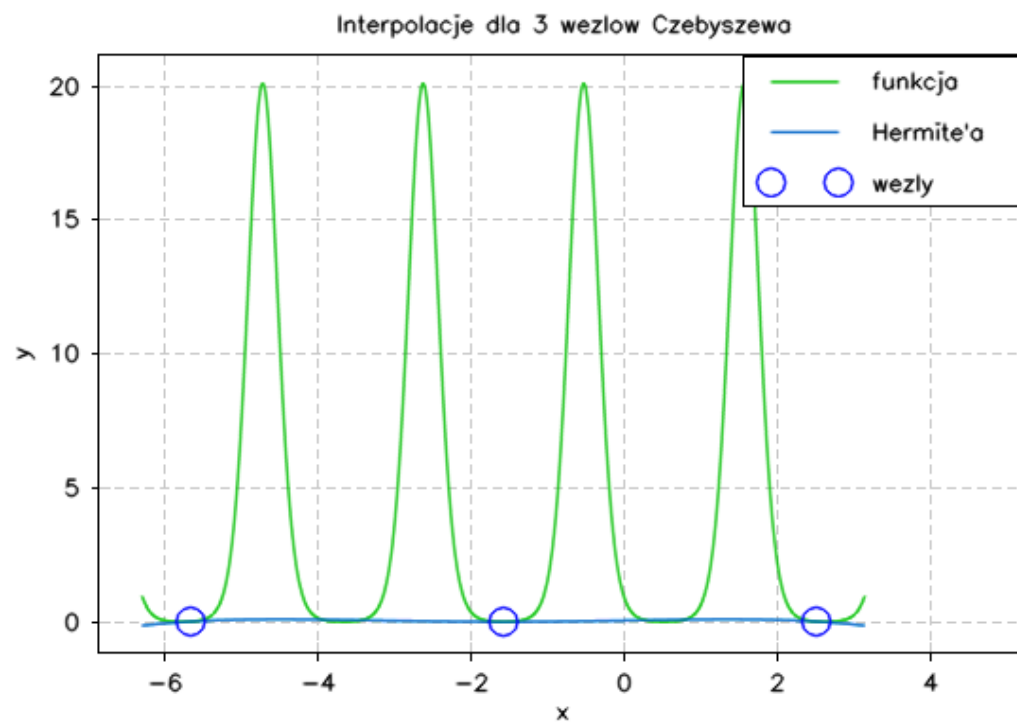
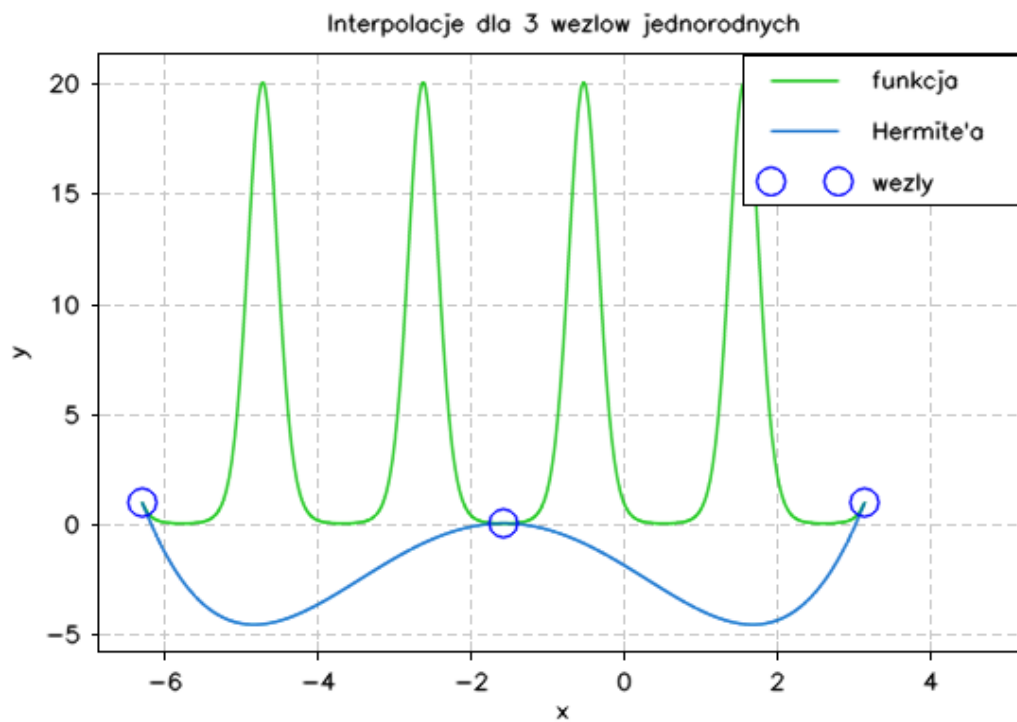
Sumy kwadratów błędów bezwzględnych w pliku

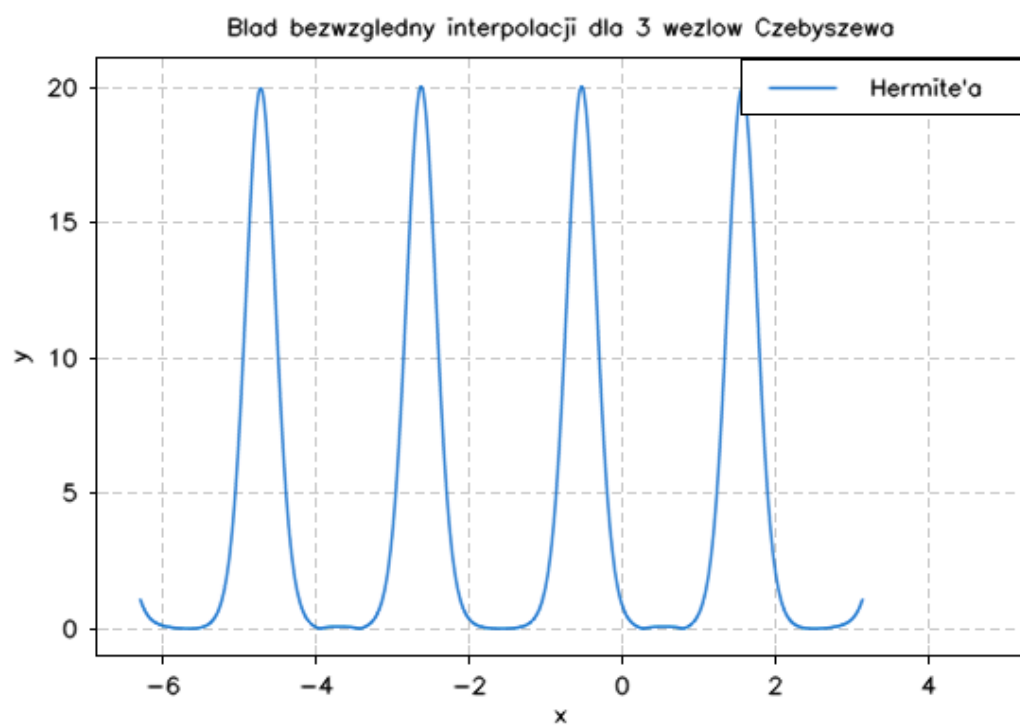
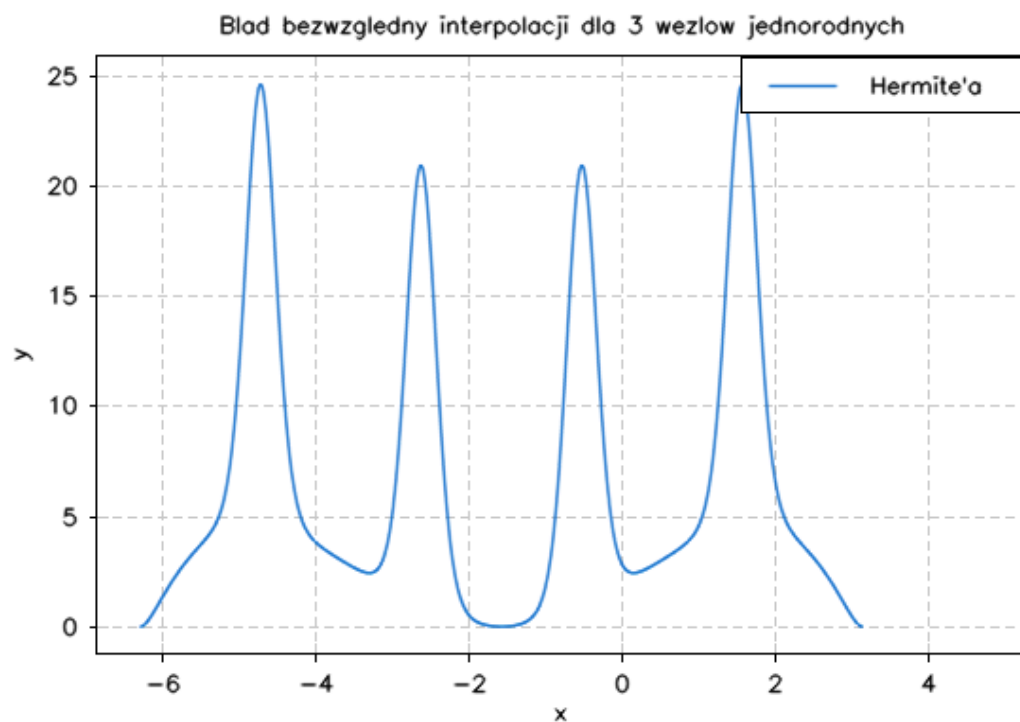
`interpolation_results/sum_squared.txt`

W folderze `interpolation_images/` są zapisywane wykresy funkcji interpolowanej, interpolacji i wartości błędów.

6. Wyniki obliczeń

6.1 Dla 3 węzłów

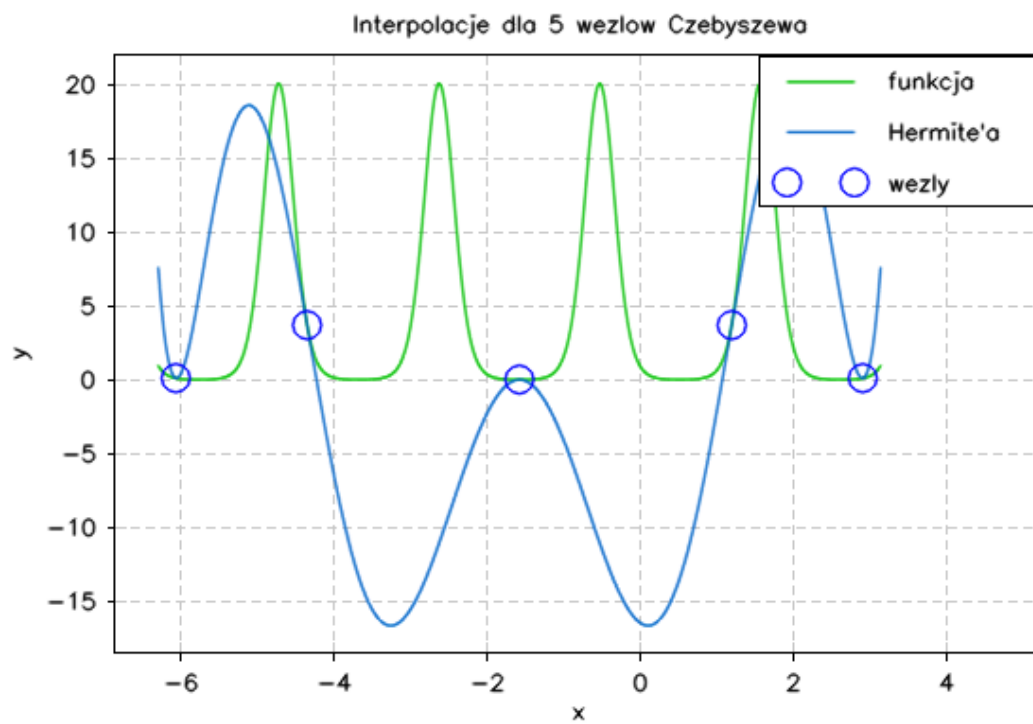
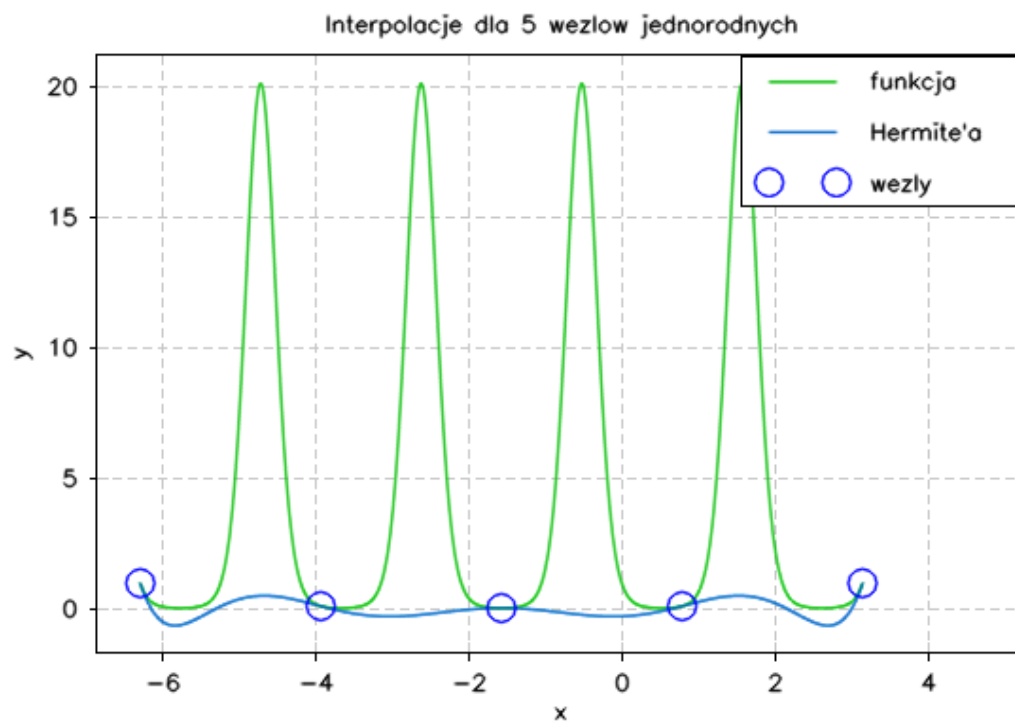


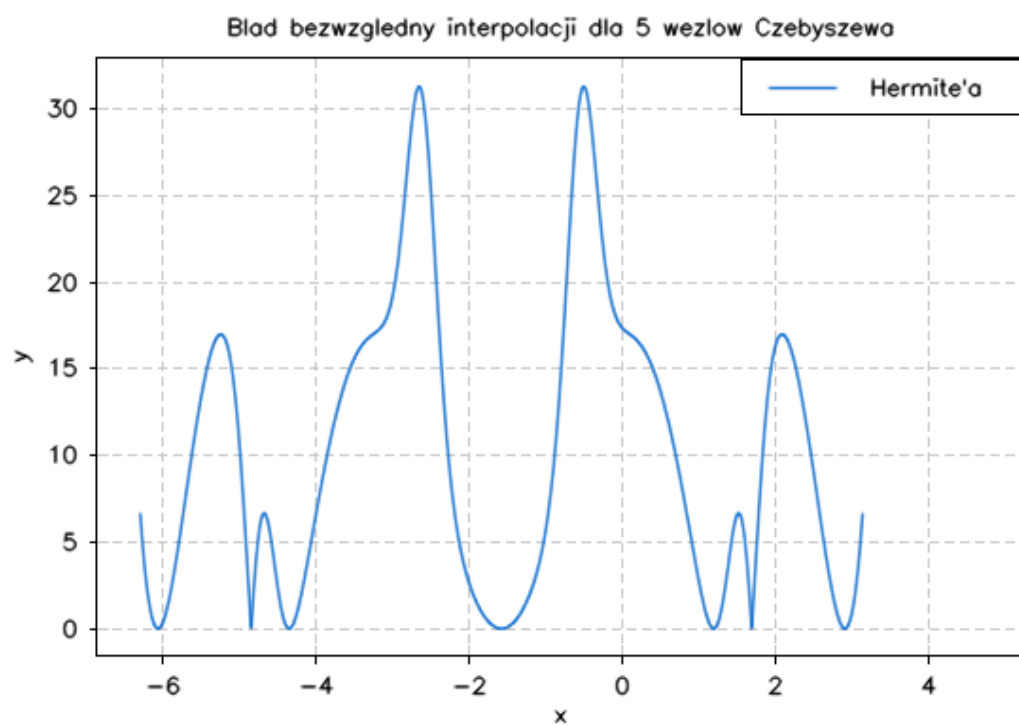
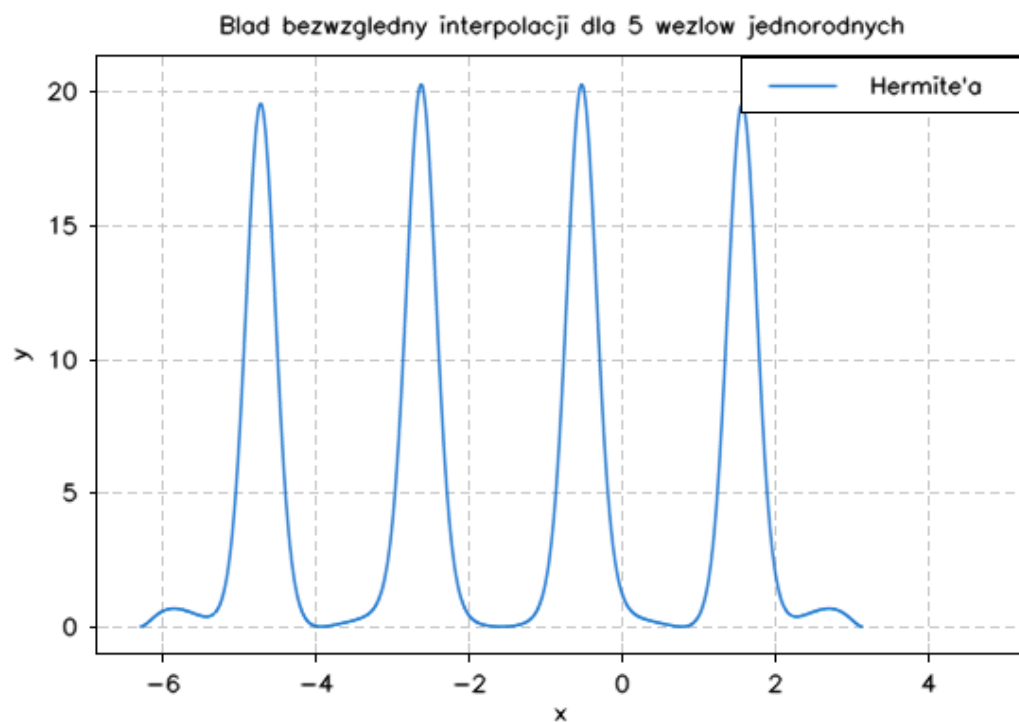


	Węzły	
Błąd względny	Równoodległe	Czebyszewa
Max	24.61492034	20.0138018
Suma	63131.73915	40442.5372

Tabela 1.
Błędy interpolacji dla 3 węzłów

6.2 Dla 5 węzłów

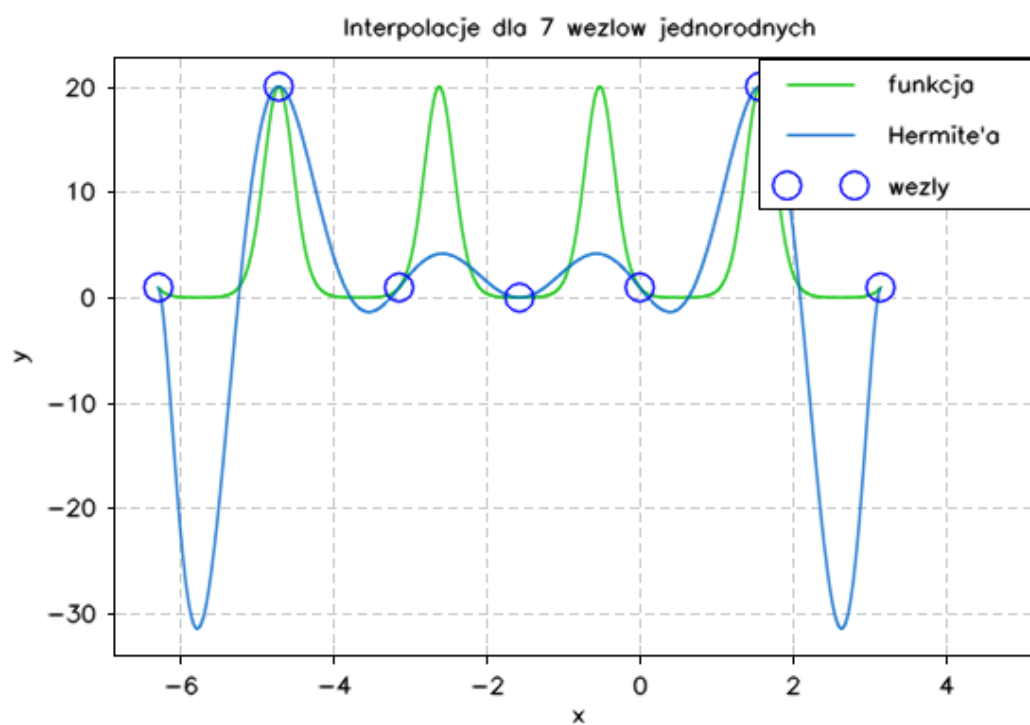




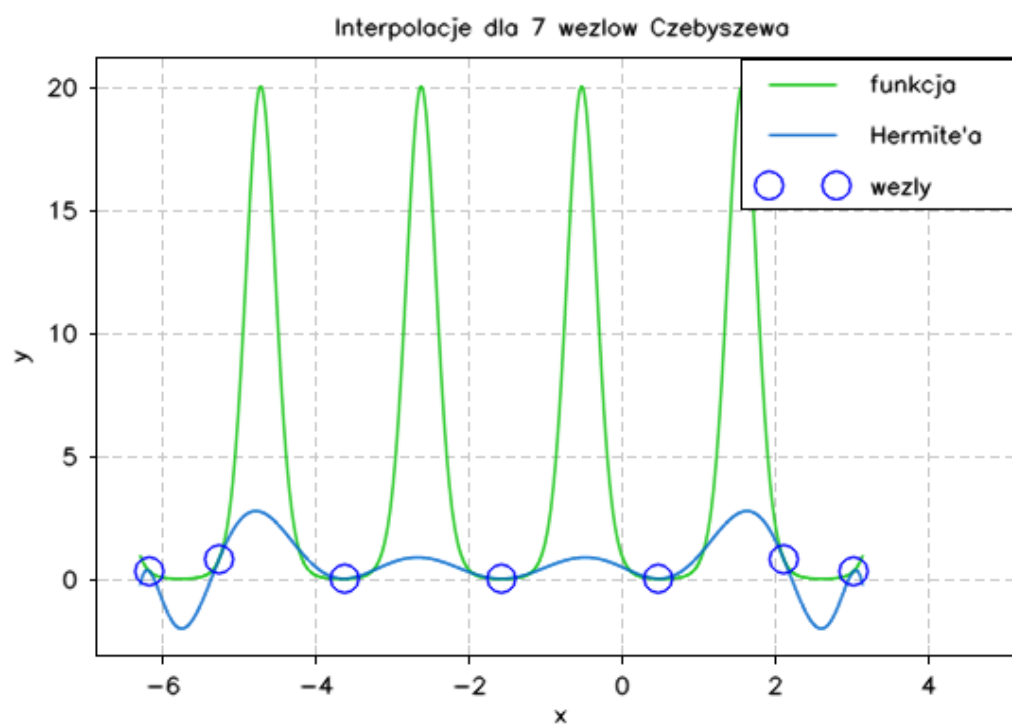
Błąd względny	Węzły	
	Równoodległe	Czebyszewa
Max	20.29266626	31.2863715
Suma	41218.223	96723.8046

Tabela 2.
Błędy interpolacji dla 5 węzłów

6.3 Dla 7 węzłów

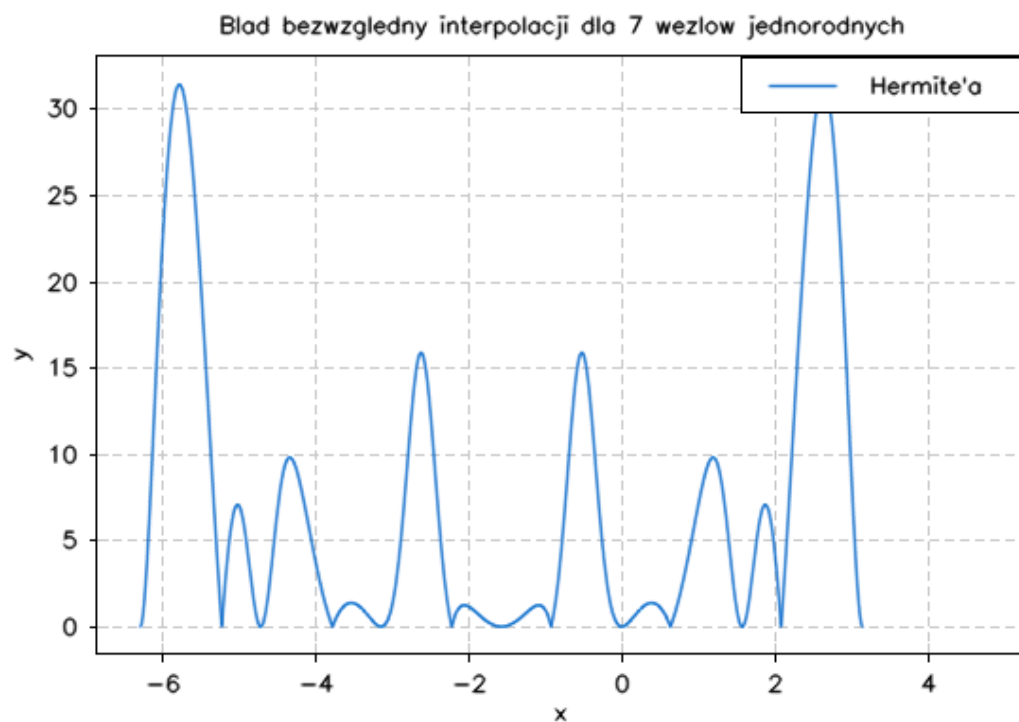


Wykres 9.

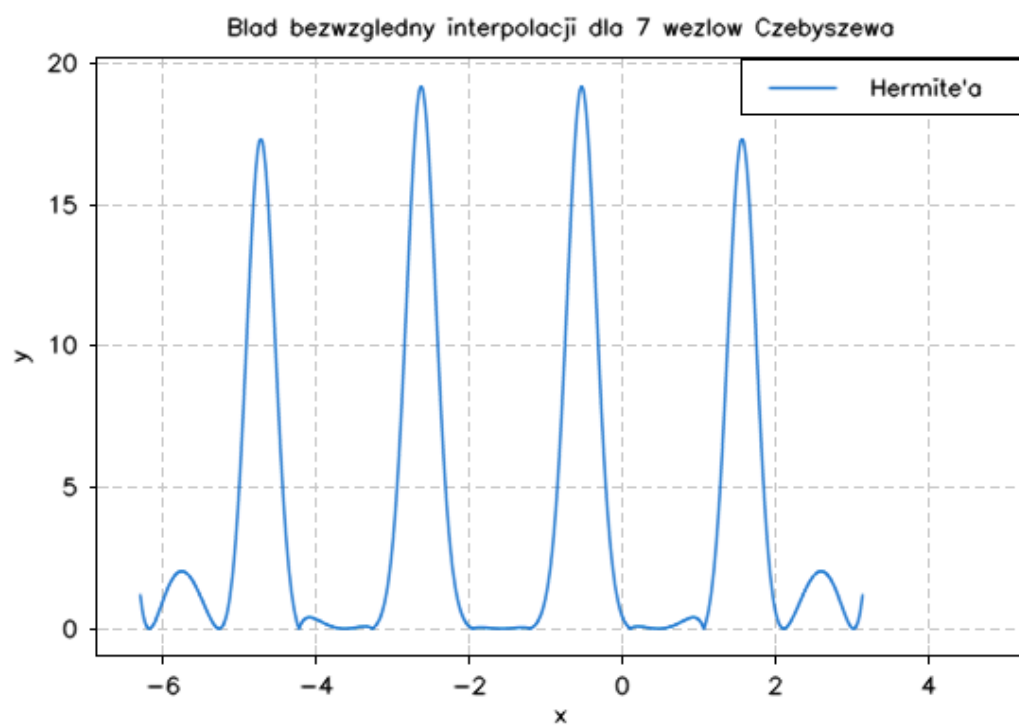


Wykres 10.

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



Wykres 11.

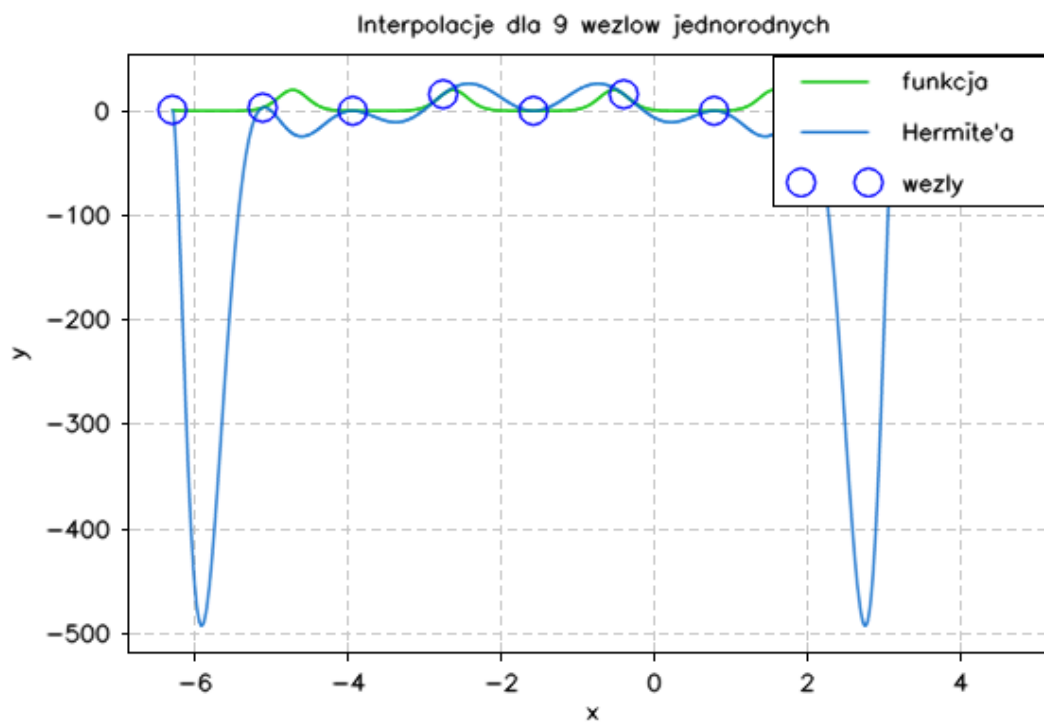


Wykres 12.

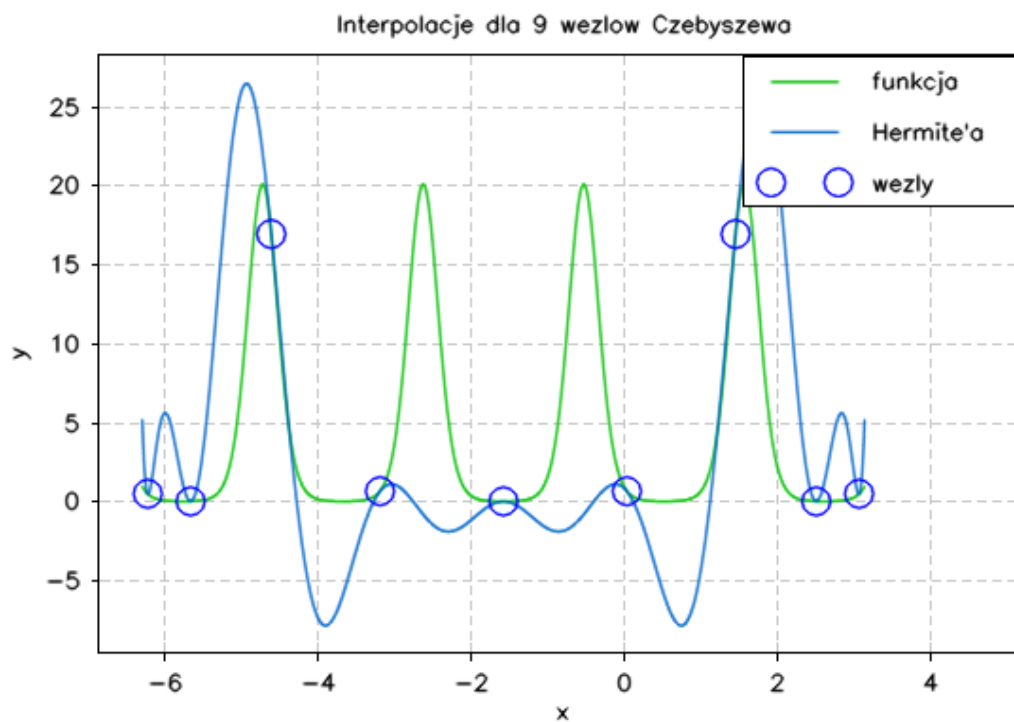
	Węzły	
Błąd względny	Równoodległe	Czebyszewa
Max	31.44664324	19.1695261
Suma	67537.84604	36252.883

Tabela 3.
Błędy interpolacji dla 7 węzłów

6.4 Dla 9 węzłów

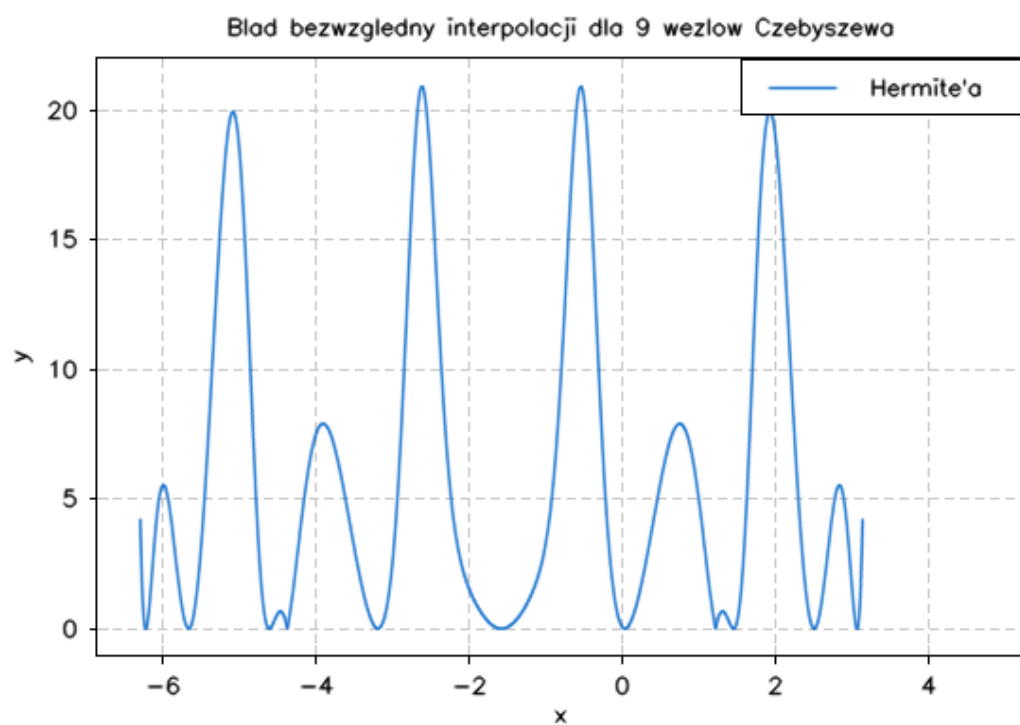
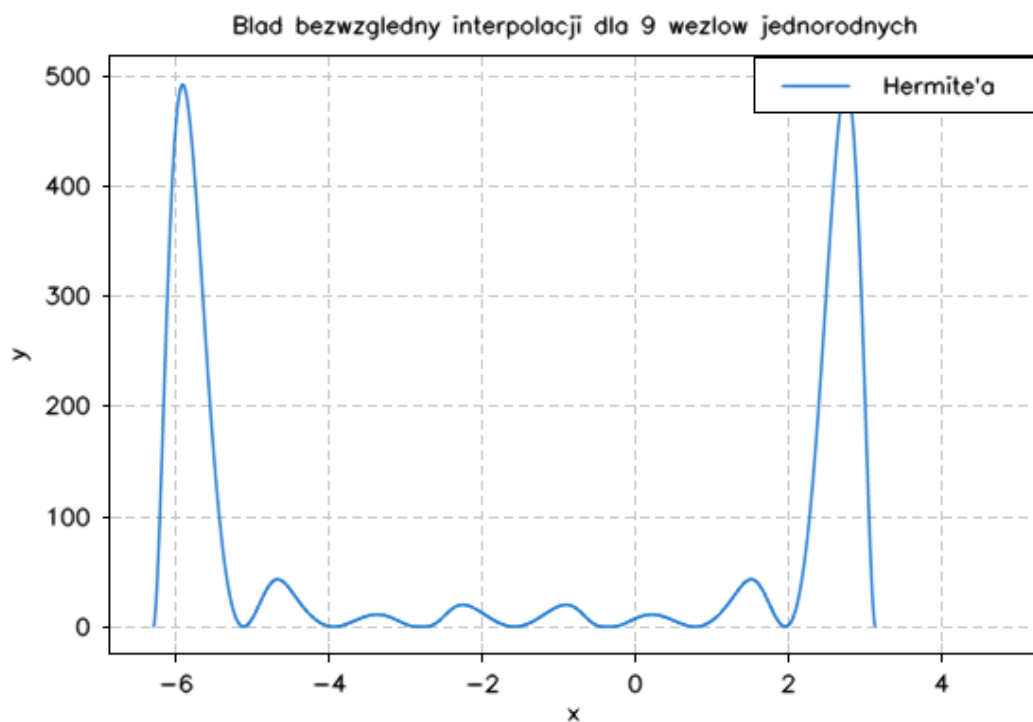


Wykres 13.



Wykres 14.

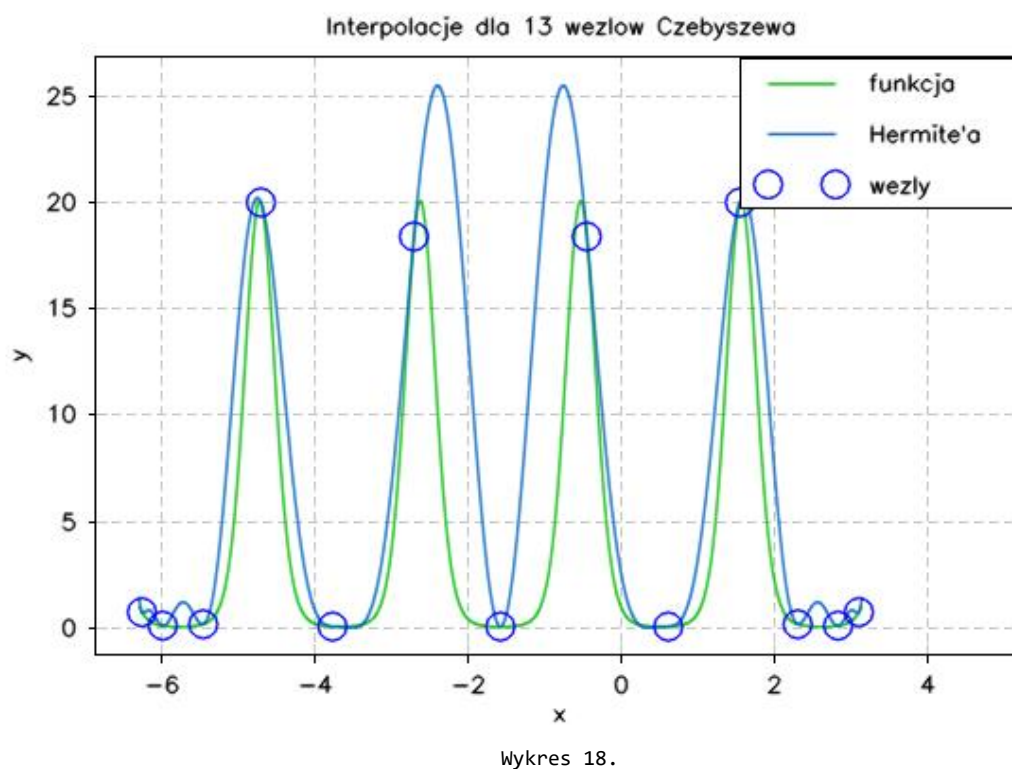
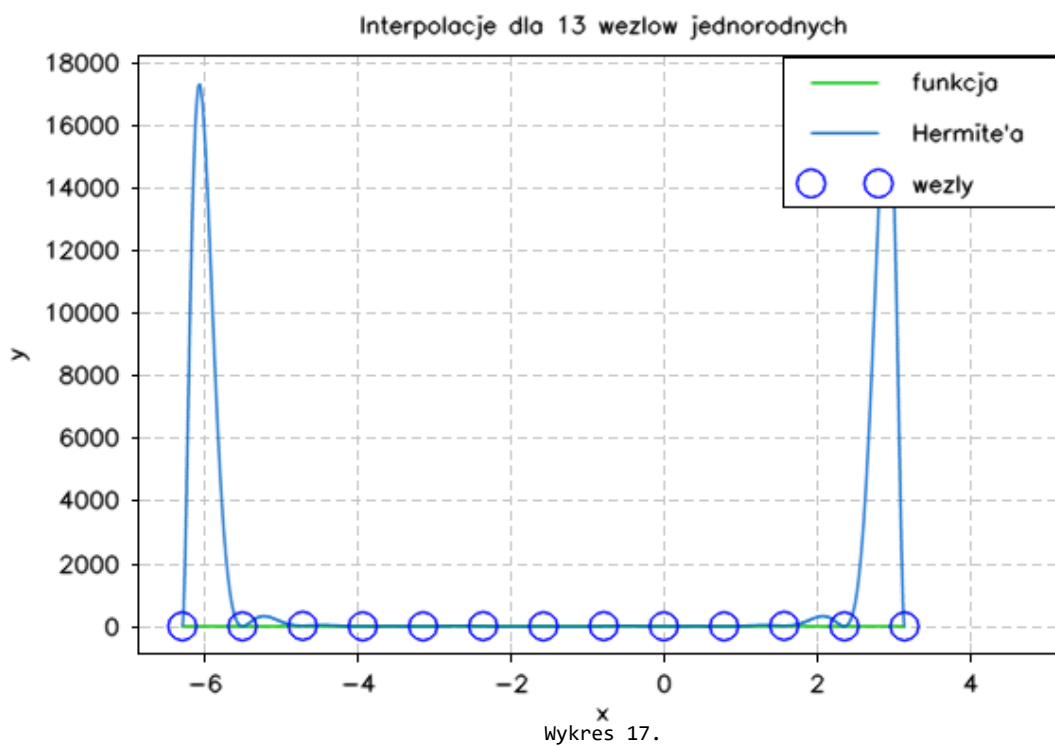
Dla 9 węzłów zaczyna być widoczny duży efekt Runge'go na krańcach przedziału interpolacji, tylko dla równoodległych węzłów.



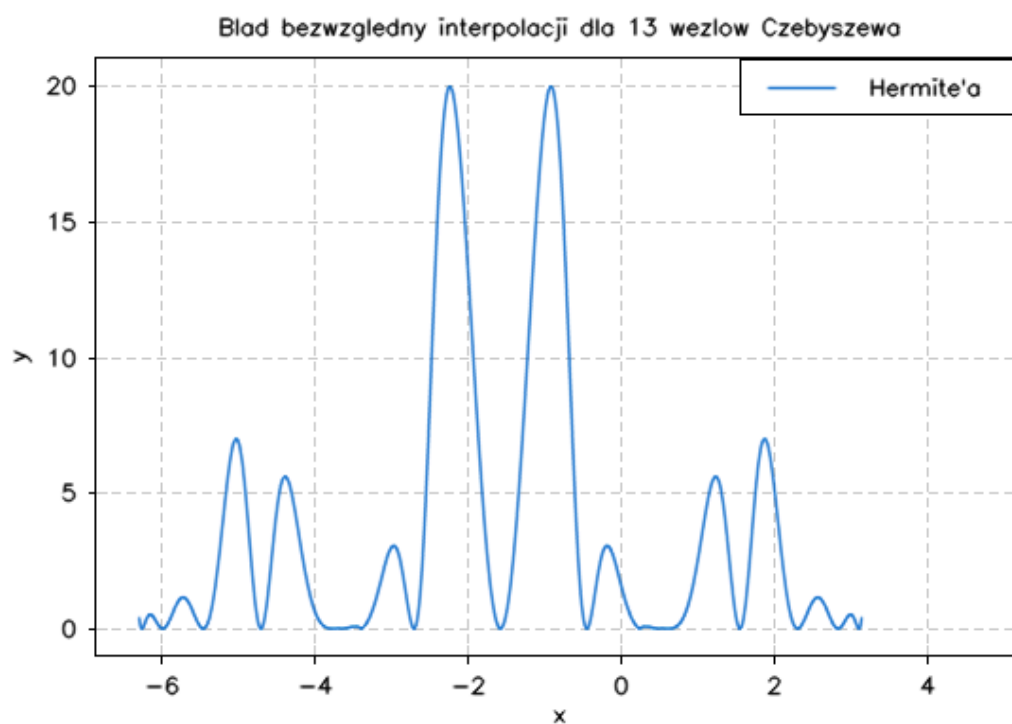
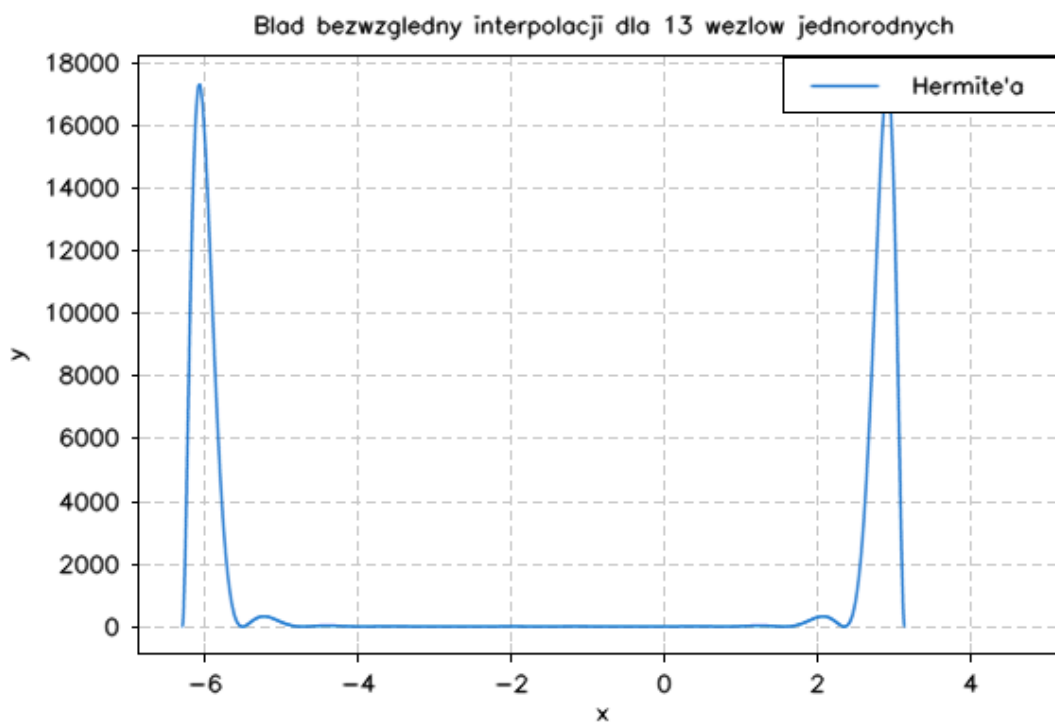
	Węzły	
Błąd względny	Równoodległe	Czebyszewa
Max	492.5958018	20.8921088
Suma	620202.836	55679.0905

Tabela 4.
Błędy interpolacji dla 9 węzłów

6.5 Dla 13 węzłów



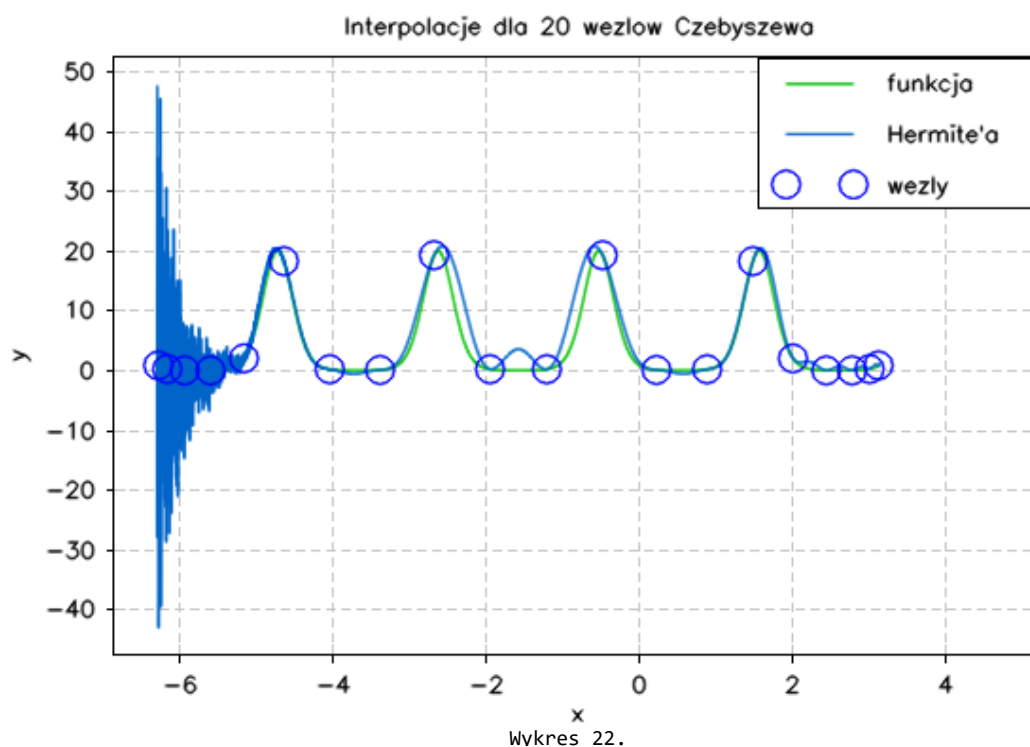
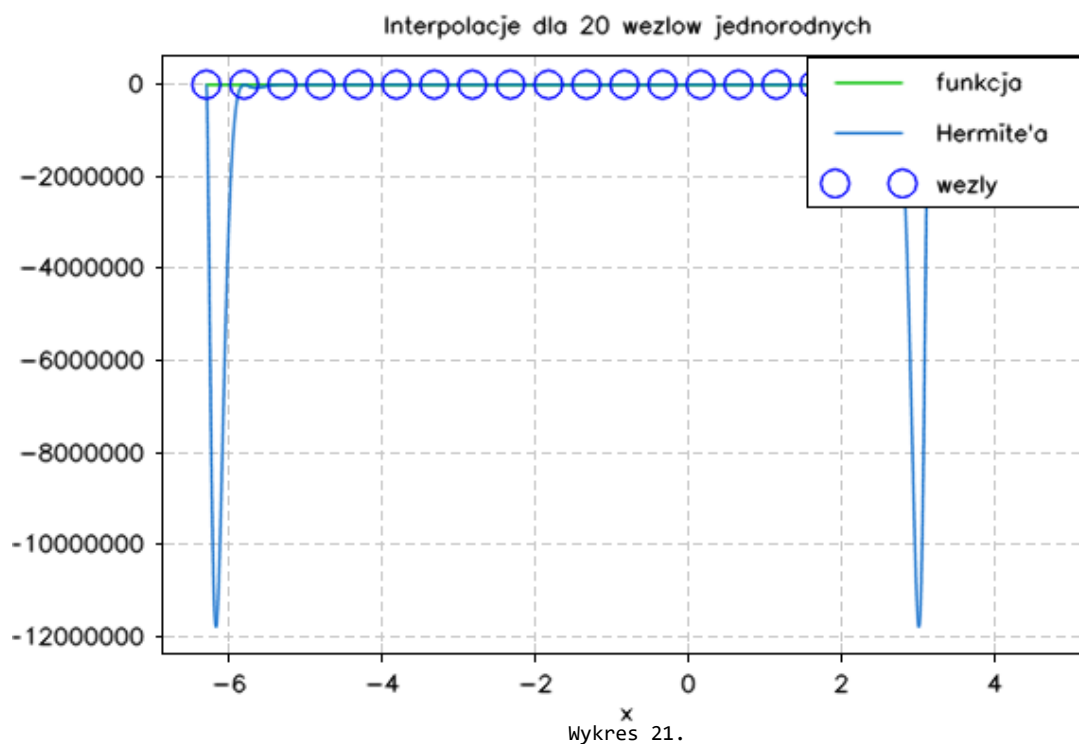
Dla 13 widoczny jeszcze większy efekt Runge'go na krańcach przedziału interpolacji, tylko dla równoodległych węzłów.



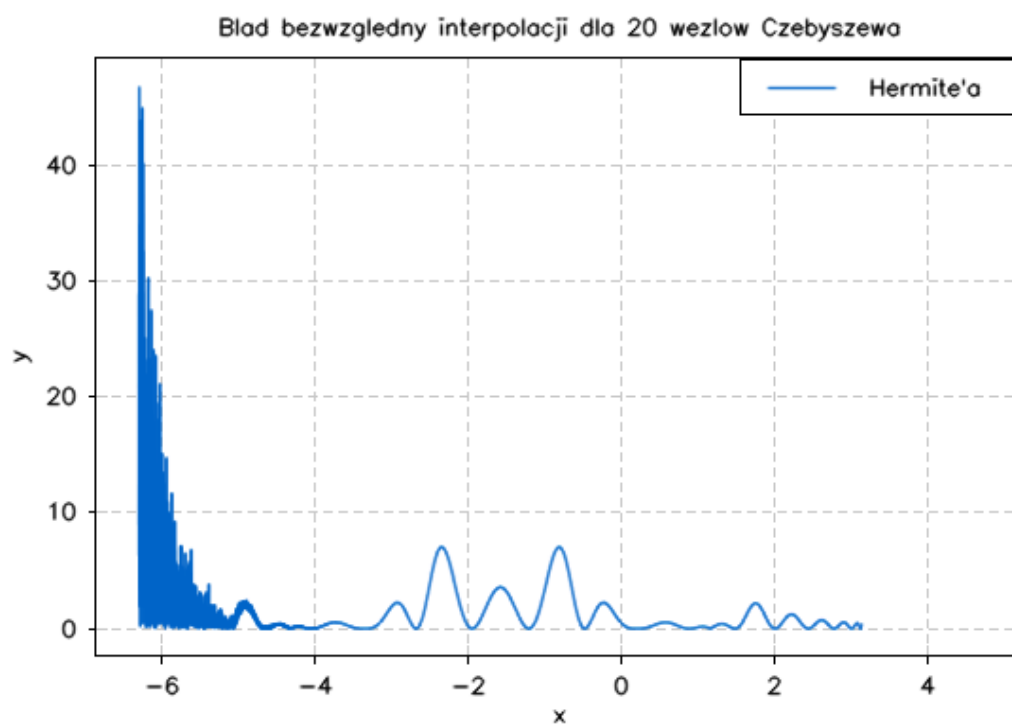
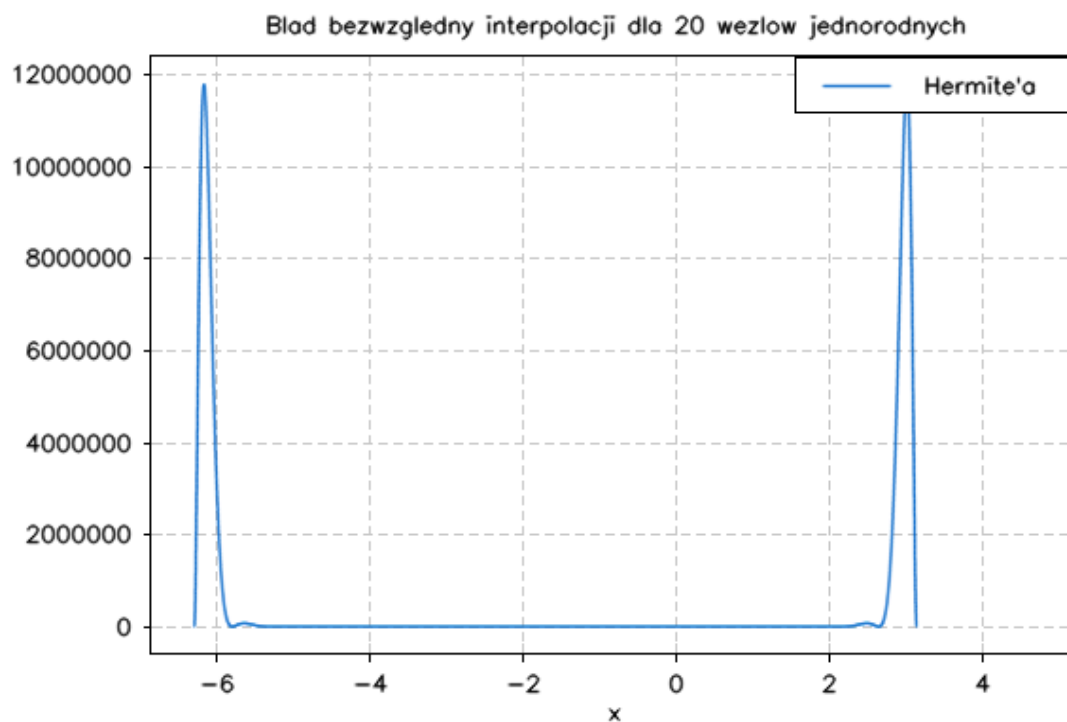
	Węzły	
Błąd względny	Równoodległe	Czebyszewa
Max	17315.99249	20.0039609
Suma	11979256.97	34311.8271

Tabela 5.
Błędy interpolacji dla 13 węzłów

6.6 Dla 20 węzłów



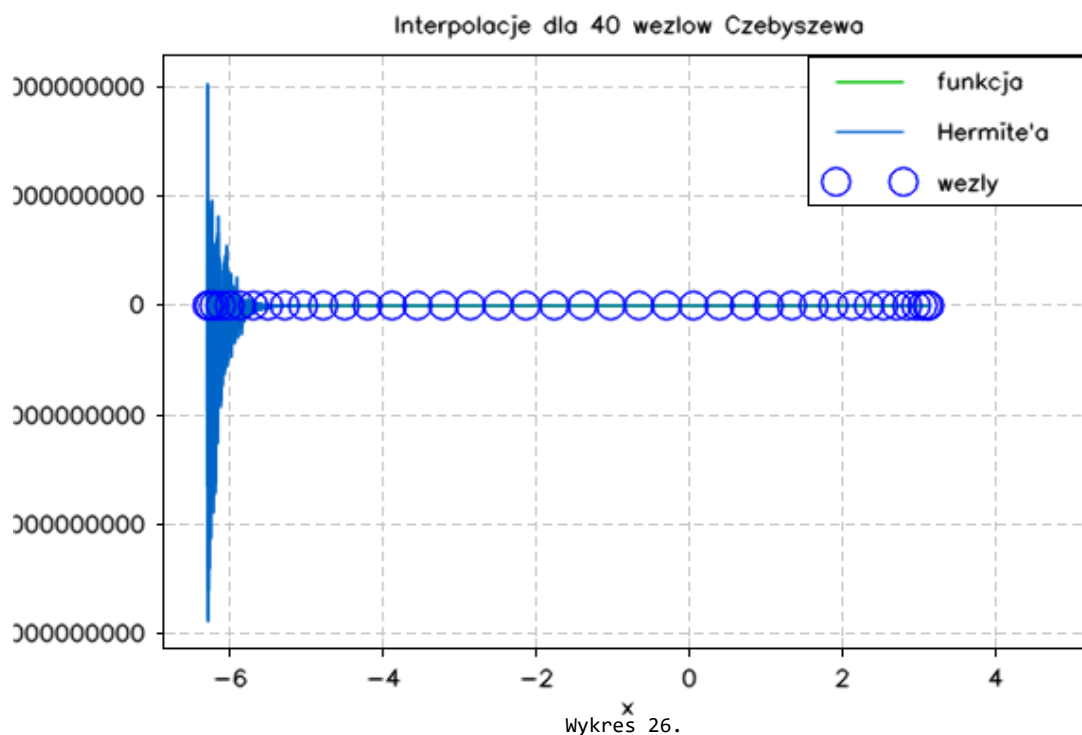
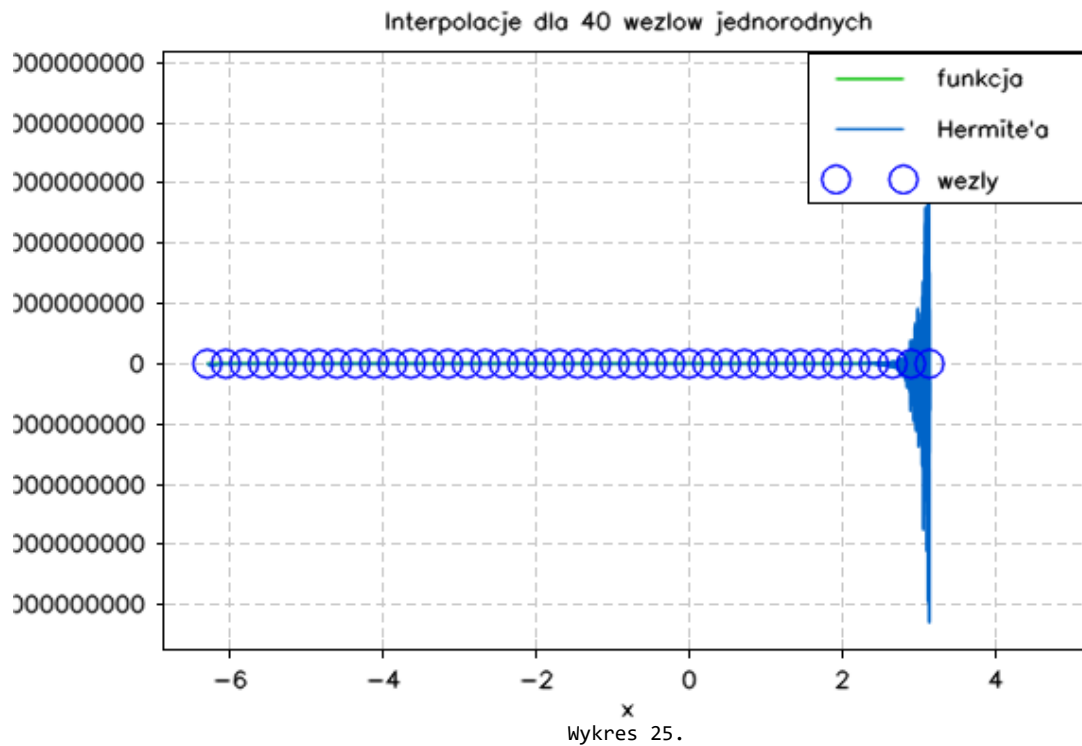
Dla 20 węzłów widoczny ogromny efekt Runge'go na krańcach przedziału interpolacji, tylko dla równoodległych węzłów. Pomijając błędy numeryczne na początku przedziału, interpolacja węzłami Czebyszewa jest dość dokładna.



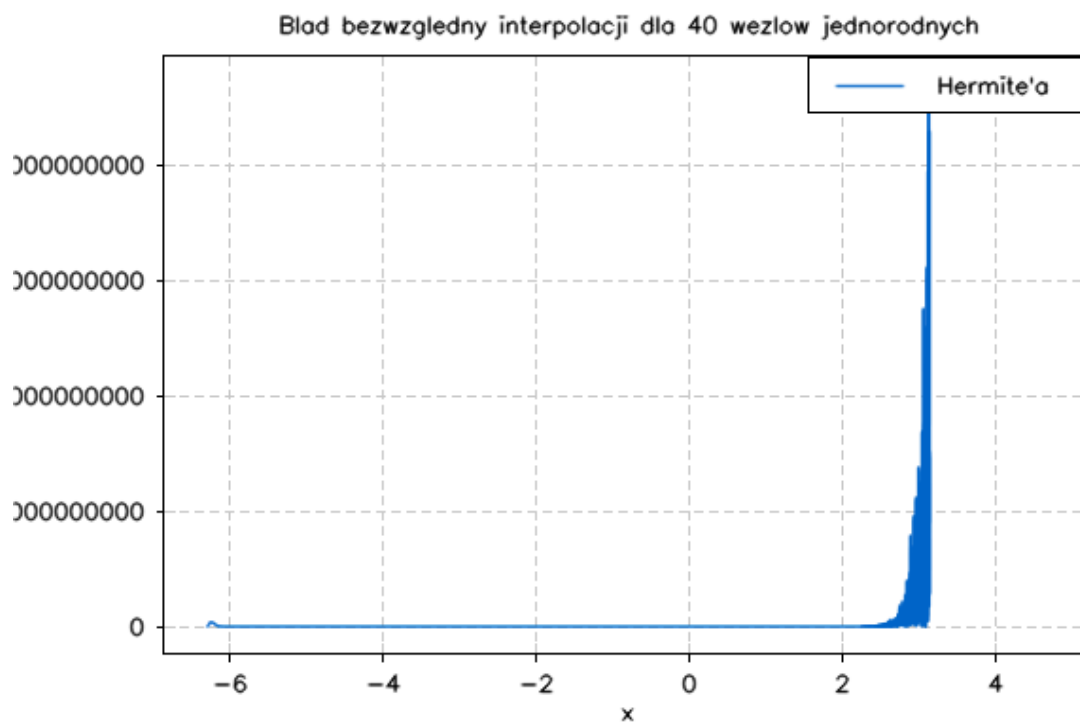
Błąd względny	Węzły	
	Równoodległe	Czebyszewa
Max	11789410.71	46.7717509
Suma	4709853330	14282.8814

Tabela 6.
Błędy interpolacji dla 20 węzłów

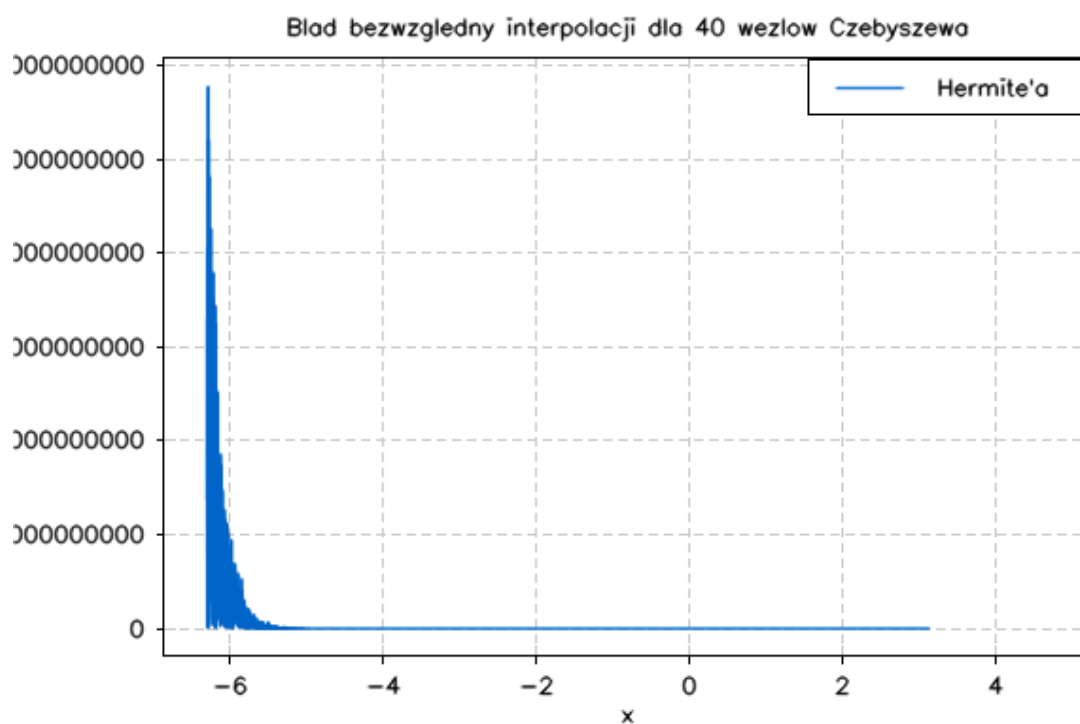
6.7 Dla 40 węzłów



Dla 40 węzłów w obu przypadkach 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 10^{17} .



Wykres 27.



Wykres 28.

	Węzły	
Błąd względny	Równoodległe	Czebyszewa
Max	4.70045E+17	5.7692E+17
Suma	1.9454E+19	3.5831E+19

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.

Ilość węzłów	Równoodległe	Czebyszewa	Ilość węzłów	Równoodległe	Czebyszewa
2	39.24776298	113.5471425	2	164947.1755	596322.7168
3	24.61492034	20.01380179	3	63131.73915	40442.53717
4	29.0260187	20.64143568	4	76910.38627	41642.367
5	20.29266626	31.28637154	5	41218.223	96723.80457
6	105.845553	80.77905644	6	270567.0822	178502.3482
7	31.44664324	19.16952606	7	67537.84604	36252.88303
8	15.82839029	27.7939095	8	54355.38135	75872.39658
9	492.5958018	20.89210877	9	620202.836	55679.09049
10	16.08352194	17.01339068	10	32519.67438	55598.74927
11	3187.587407	17.32202029	11	2807696.143	45981.56732
12	7620.16696	17.09701183	12	5865536.509	36062.5123
13	17315.99249	20.00396092	13	11979256.97	34311.82712
14	28883.79889	12.97746689	14	18131086.69	20415.18791
15	46755.87612	11.28238081	15	26962466.47	17568.30225
16	423763.6821	7.964421343	16	223960320	18775.14759
17	1364754.347	7.473871957	17	666991978	20170.91171
18	2848571.373	8.631264748	18	1294693835	15735.04884
19	2342921.654	6.939107175	19	992640396.2	12564.12585
20	11789410.71	46.77175088	20	4709853330	14282.88144
21	68692138.7	78.31799297	21	25799058337	17080.82621
22	208233933.5	2636.000302	22	73810160534	206315.2764
23	396123328.1	4319.545301	23	1.32873E+11	336440.9871
24	191576095.2	114281.317	24	60711229272	9444766.232
25	2051330388	112591.9217	25	6.22666E+11	11994787.68
25+	wartości rosnące		25+	wartości rosnące	

Tabela 8.

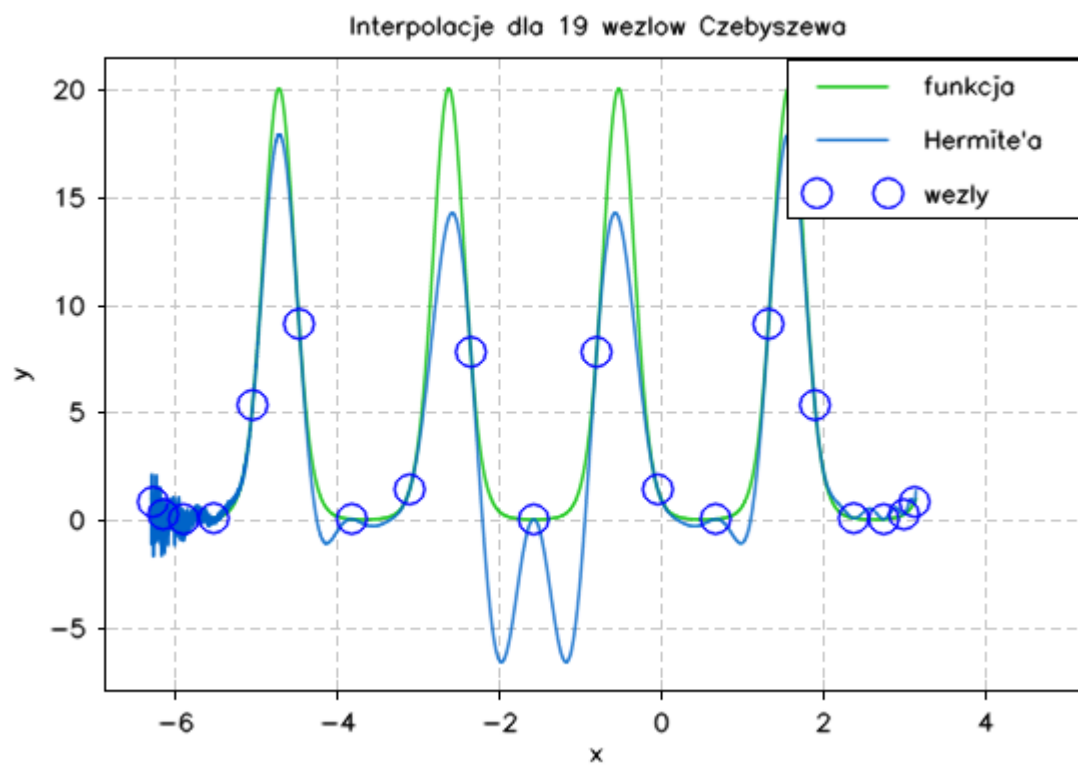
Maksymalne błędy interpolacji

Tabela 9.

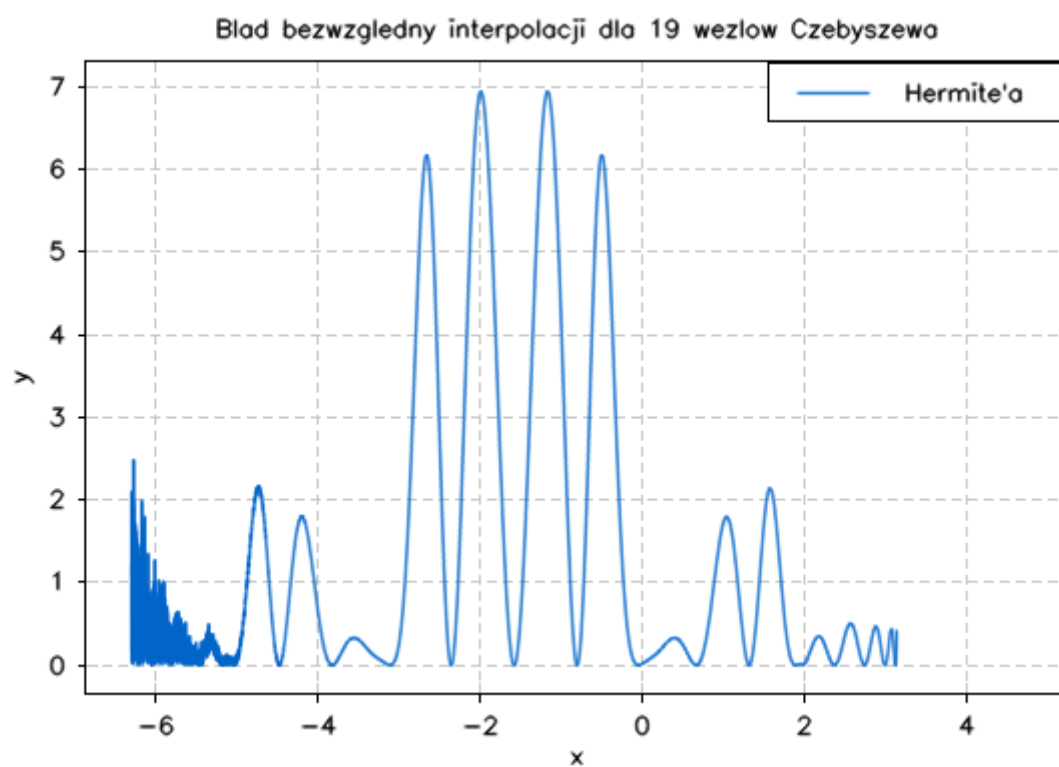
Sumy błędów interpolacji

Jak łatwo zauważyć, najmniejszą wartość maksymalnego błędu bezwzględnego jak i najmniejszą sumę błędów bezwzględnych, równe kolejno (6.93910717478166) i (12564.1258504046), uzyskano przy wykorzystaniu 19 węzłów Czebyszewa. W powyższych tabelach pominąłem wartości dla liczby węzłów > 25 , 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.



Wykres 29.



Wykres 30.

7. Wnioski

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.