

# MOwNiT - Aproksymacja sredniokwadratowa wielomianowa

Jakub Frączek

18 kwietnia 2024

## 1 Funkcja dla której przeprowadzone zostało doświadczenie

$$f(x) = 10 * m + \frac{x^2}{k} - 10 * m * \cos(k * x)$$

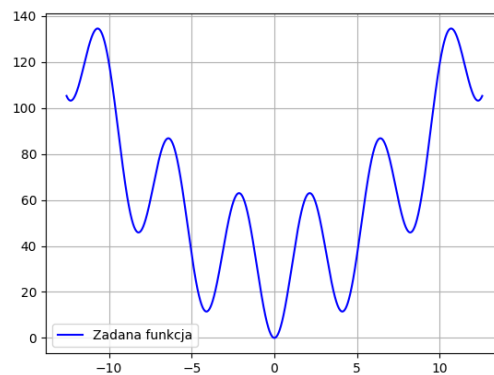
gdzie:

$$k = 1.5$$

$$m = 3.0$$

$$x \in [-4\pi, 4\pi]$$

Wykres funkcji:



## 2 Dane techniczne

### 2.1 Hardware

Laptop z procesorem Intel Core i5-9300H 2.4GHz oraz 32 GB pamięci RAM.

### 2.2 Software

Wykorzystany został system Windows 11 x64 oraz język Python w wersji 3.11.8 wraz z bibliotekami:

- math
- copy
- matplotlib
- numpy

### 3 Aproksymacja

Funkcja bazowe, czyli ciągi jednomianów  $\varphi_j(x) = x^j, j = 0, 1, \dots, m$

Funkcja aproksymująca:  $f(x) = \sum_{j=0}^m a_j \varphi_j(x) = \sum_{j=0}^m a_j x^j$

F(x) - zadana na zbiorze dyskretnym  $\{x_i\}, i = 0, 1, \dots, n$

Szukamy takich współczynników  $a_j$ , że:

$$\min! \sum_{i=0}^n w(x_i) [F(x_i) - f(x_i)]^2$$

Układ normalny:

$$\sum_{i=0}^n w(x_i) [F(x_i) - \sum_{j=0}^m a_j x_i^j] x_i^k \frac{\partial f}{\partial a_k} = 0, k = 0, 1, \dots, m$$

$$\sum_{i=0}^n w(x_i) x_i^k \sum_{j=0}^m a_j x_i^j = \sum_{i=0}^n w(x_i) F(x_i) x_i^k, k = 0, 1, \dots, m$$

$$\sum_{j=0}^m \left( \sum_{i=0}^n w(x_i) x_i^{j+k} \right) a_j = \sum_{i=0}^n w(x_i) F(x_i) x_i^k$$

W postaci macierzowej:

$$\begin{pmatrix} \sum w_i & \sum w_i x_i & \sum w_i x_i^2 & \dots & \sum w_i x_i^m \\ \sum w_i x_i & \sum w_i x_i^2 & \sum w_i x_i^3 & \dots & \sum w_i x_i^{m+1} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ \sum w_i x_i^m & \sum w_i x_i^{m+1} & \sum w_i x_i^{m+2} & \dots & \sum w_i x_i^{2m} \end{pmatrix} \begin{pmatrix} a_0 \\ a_1 \\ \vdots \\ a_m \end{pmatrix} = \begin{pmatrix} \sum w_i F_i \\ \sum w_i F_i x_i \\ \vdots \\ \sum w_i F_i x_i^m \end{pmatrix}$$

$$G \cdot A = B$$

### 4 Funkcja wyliczająca aproksymację

Po dyskusji na zajęciach i weryfikacji przeze mnie rozwiązania, okazało się, że aproksymacja jest wyliczana nie do końca prawidłowo.

Funkcja przed poprawą wyglądała następująco:

```
def calculate_approx(xs, ys, m, w):

    G = np.zeros((m, m))
    B = np.zeros(m)

    for j in range(m):
        for k in range(m):

            G[j][k] = sum([w[i] * (xs[i] ** (j + k)) for i in range(len(xs))])

            B[j] = sum([w[i] * ys[i] * (xs[i] ** j) for i in range(len(xs))])

    A = np.linalg.solve(G, B)

    return lambda x : sum([A[i] * (x ** i) for i in range(m)])
```

Natomiast poprawiona:

```
def calculate_approx(xs, ys, m, w):

    G = np.zeros((m + 1, m + 1))
    B = np.zeros(m + 1)

    for j in range(m + 1):
        for k in range(m + 1):

            G[j][k] = sum([w[i] * (xs[i] ** min(j + k, 2 * m + 1))
                           for i in range(len(xs))])

            B[j] = sum([w[i] * ys[i] * (xs[i] ** j) for i in range(len(xs))])

    A = np.linalg.solve(G, B)

    return lambda x : sum([A[i] * (x ** i) for i in range(m + 1)])
```

Wniosek jest taki, że stopień wielomianu nie był wyliczony poprawnie i zmienna  $m$  wcześniej nie oznaczała stopnia wielomianu.

## 5 Metody szacowania błędu przybliżenia funkcji

Wszystkie błędy zostały policzone z dokładnością do 100 równoodległych punktów.

### 5.1 Największa różnica wartości funkcji

Największa różnica między wartością funkcji aproksymowanej, a funkcji aproksymującej:

$$\max_{x \in [a, b]} |F(x) - P_n(x)|$$

### 5.2 Błąd średniokwadratowy

Suma kwadratów różnic między wartością funkcji aproksymowanej, a funkcji aproksymującej podzielona przez liczbę punktów, w których wykonujemy porównanie:

$$\frac{1}{N} * \sum_{i=1}^N (F(x_i) - P_n(x_i))^2$$

## 6 Analiza

Podczas analizy funkcji aproksymującej, węzły oraz stopnie wielomianów dobrałem zgodnie z poniższym twierdzeniem:

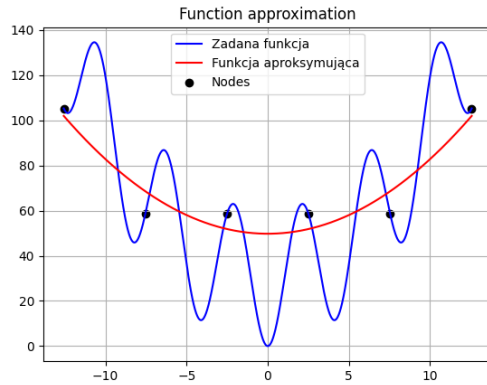
Jeżeli  $x_1, x_2, \dots, x_n$  są różne oraz  $m \leq n$ , to  $G \neq 0$

Ponownie, po dyskusji na zajęciach ilość punktów wykorzystywanych przy liczeniu błędów średniokwadratowego i maksymalnego oraz rysowaniu funkcji została zmieniona ze 100 na 1000.

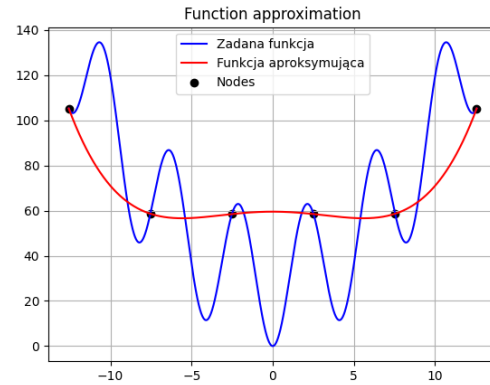
## 7 Przebieg funkcji dla wybranej liczby węzłów

### 7.1 Dla 6 węzłów

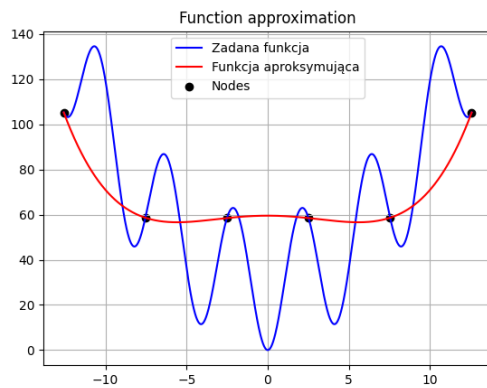
Jak widać na poniższych wykresach (wykres 1, wykres 2, wykres 3, wykres 4), mimo małego błędu średniokwadratowego aproksymacja nie jest zbyt dobra.



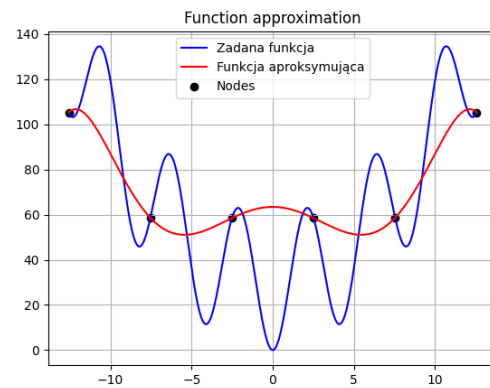
Wykres 1: Wielomian 3 stopnia



Wykres 3: Wielomian 5 stopnia

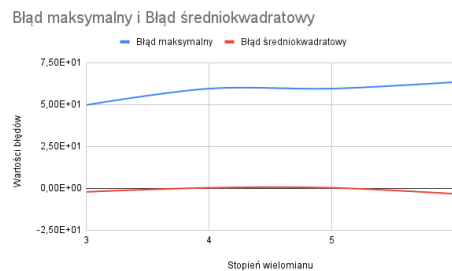


Wykres 2: Wielomian 4 stopnia



Wykres 4: Wielomian 6 stopnia

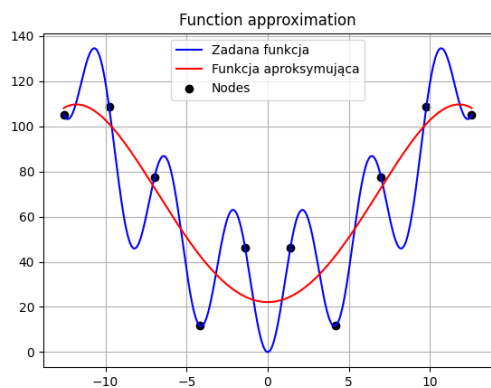
Poniżej, na wykresie 5 przedstawione zostały wartości błędów dla wszystkich możliwych stopni wielomianu.



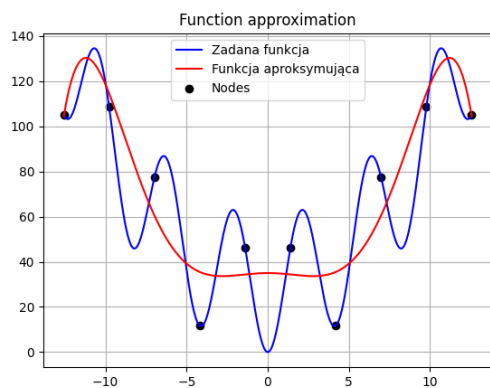
Wykres 5: Wartości błędów

## 7.2 Dla 10 węzłów

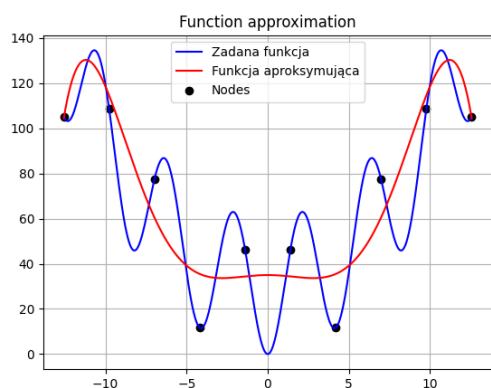
Jak widać na poniższych wykresach (wykres 6, wykres 7, wykres 8, wykres 9), najlepsze przybliżenie otrzymujemy dla wielomianu stopnia 6, a dla wielomianów niższych i wyższych stopni przybliżenie jest coraz gorsze.



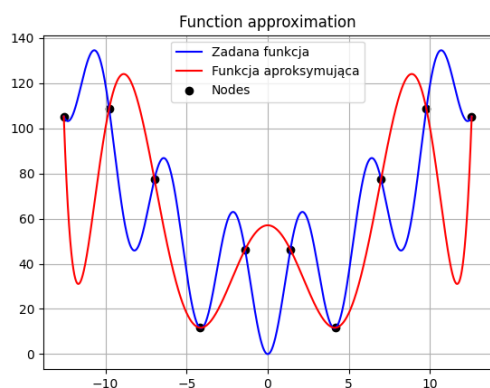
Wykres 6: Wielomian 5 stopnia



Wykres 8: Wielomian 7 stopnia

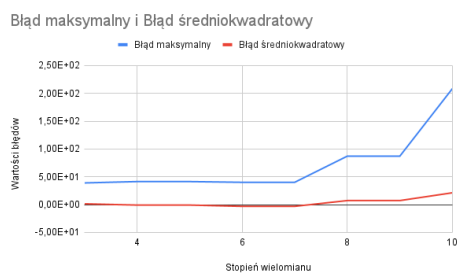


Wykres 7: Wielomian 6 stopnia



Wykres 9: Wielomian 8 stopnia

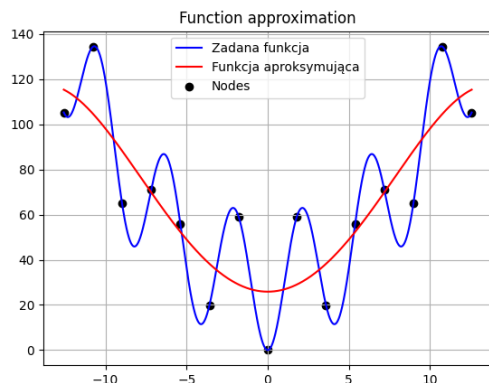
Poniżej, na wykresie 10 przedstawione zostały wartości błędów dla wszystkich możliwych stopni wielomianu.



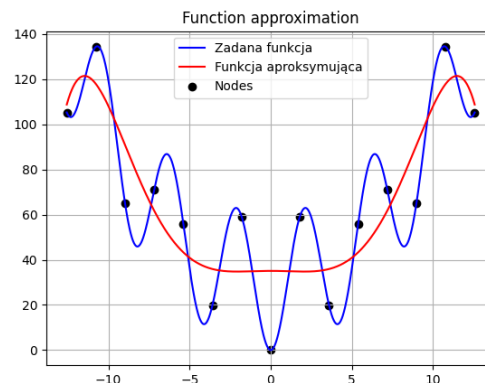
Wykres 10: Wartości błędów

### 7.3 Dla 15 węzłów

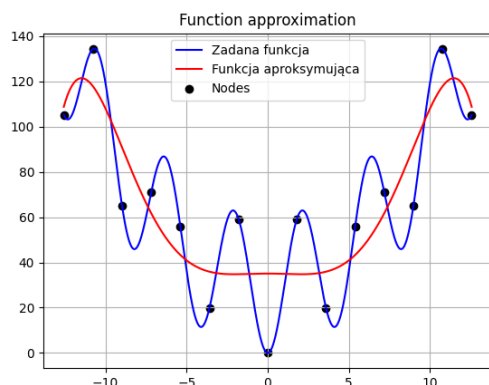
W tym przypadku przybliżenie, dalej nie jest dobre, najmniejsze błędy otrzymałem dla wielomianu 6-go stopnia, jednak, wielomian 8-go stopnia zdaje się lepiej przybliżać zadaną funkcję (wykres 11, wykres 12, wykres 13 i wykres 14).



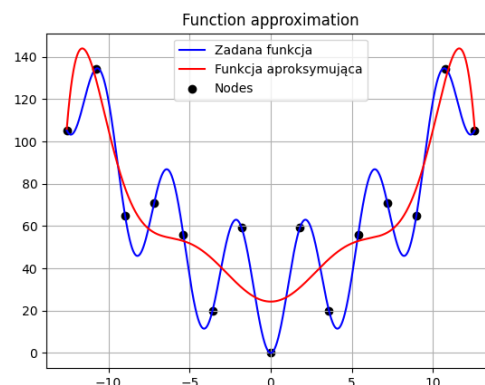
Wykres 11: Wielomian 5 stopnia



Wykres 13: Wielomian 7 stopnia

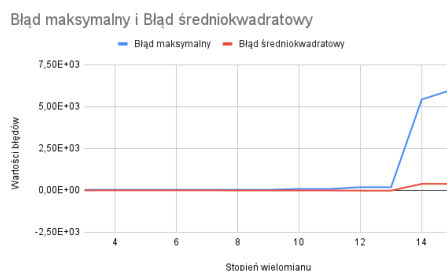


Wykres 12: Wielomian 6 stopnia



Wykres 14: Wielomian 8 stopnia

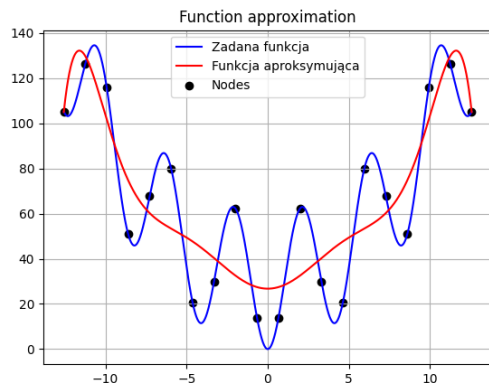
Poniżej, na wykresie 15 przedstawione zostały wartości błędów dla wszystkich możliwych stopni wielomianu.



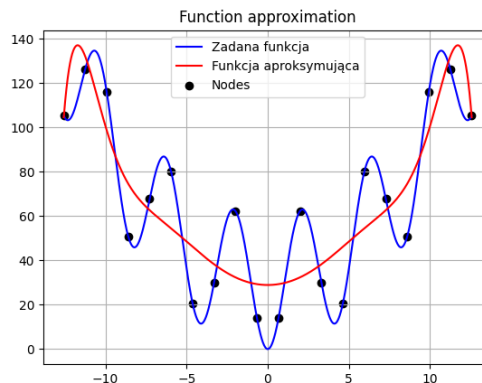
Wykres 15: Wartości błędów

## 7.4 Dla 20 węzłów

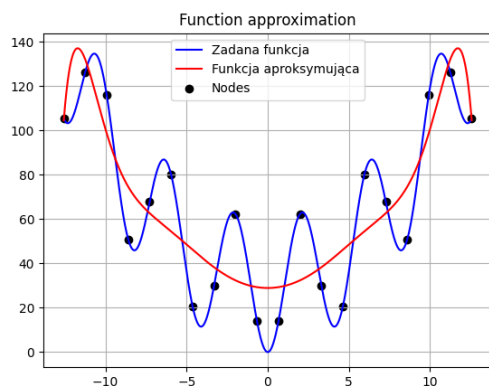
Jak widać na poniższych wykresach (wykres 16, wykres 17, wykres 18 i wykres 19), przybliżenie jest nieco bardziej dokładne, niż poprzednio. Najlepsze przybliżenie otrzymałem dla wielomianu 10-go stopnia, potem mimo, że w środku przedziału dopasowanie jest jeszcze lepsze, to na końcach znacznie zaczyna odbiegać od zadanej funkcji.



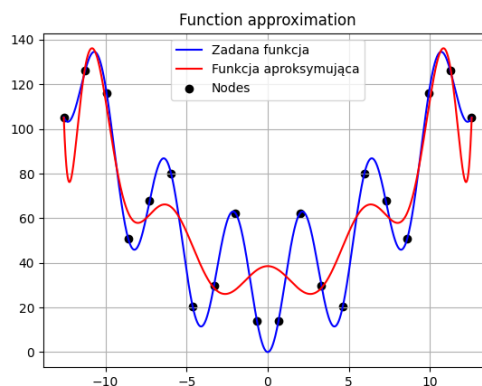
Wykres 16: Wielomian 9 stopnia



Wykres 18: Wielomian 11 stopnia



Wykres 17: Wielomian 10 stopnia



Wykres 19: Wielomian 12 stopnia

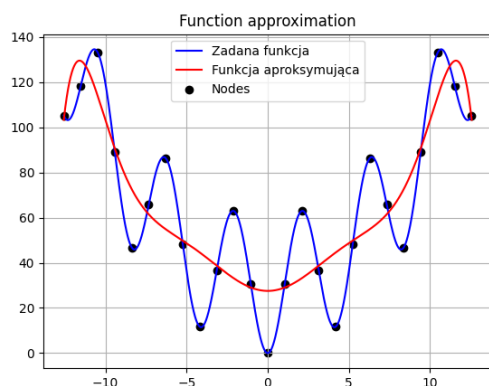
Poniżej, na wykresie 20 przedstawione zostały wartości błędów dla wszystkich możliwych stopni wielomianu.



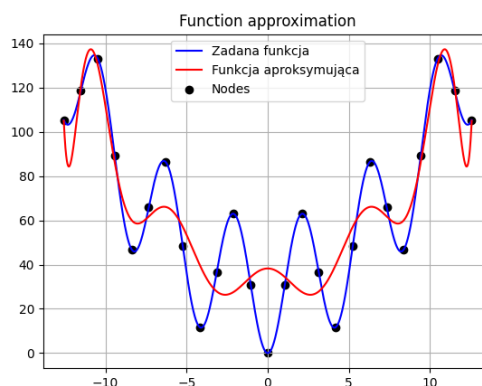
Wykres 20: Wartości błędów

## 7.5 Dla 25 węzłów

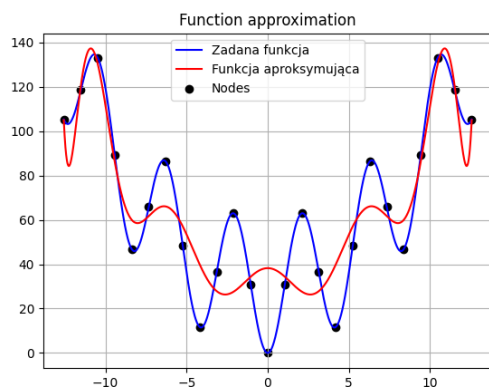
W tym przypadku otrzymałem jeszcze lepsze przybliżenie, natomiast nadal jest ono bardzo złe. Najlepszy okazał się wielomian stopnia 12, a potem błąd zaczął się bardzo szybko zwiększać (wykres 21, wykres 22, wykres 23 i wykres 24)



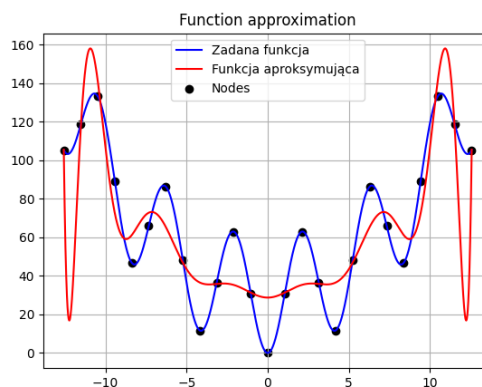
Wykres 21: Wielomian 11 stopnia



Wykres 23: Wielomian 13 stopnia

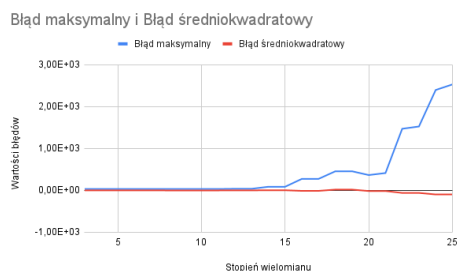


Wykres 22: Wielomian 12 stopnia



Wykres 24: Wielomian 14 stopnia

Poniżej, na wykresie 25 przedstawione zostały wartości błędów dla wszystkich możliwych stopni wielomianu.

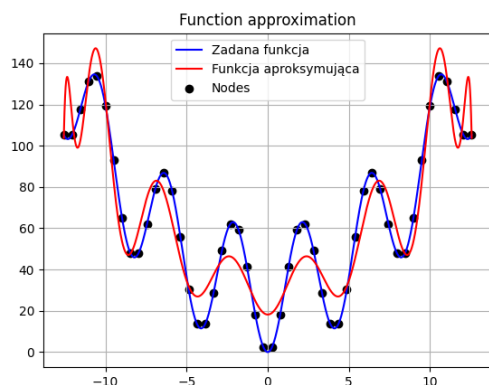


Wykres 25: Wartości błędów

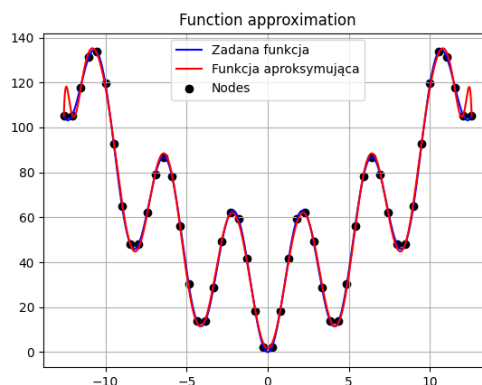


## 7.6 Dla 50 węzłów

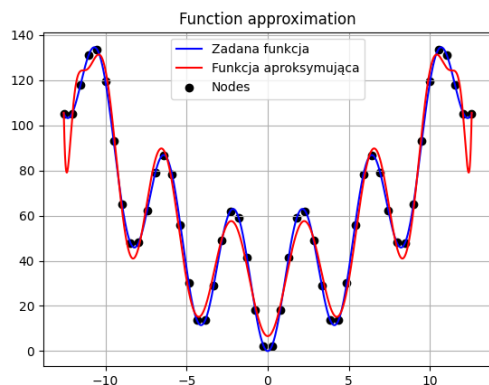
Dla 50 węzłów jesteśmy w stanie otrzymać lepsze przybliżenie, ale nadal za dobre przybliżenie w środku przedziału płacimy dużymi błędami na krańcach (wykres 26, wykres 27, wykres 28, wykres 29).



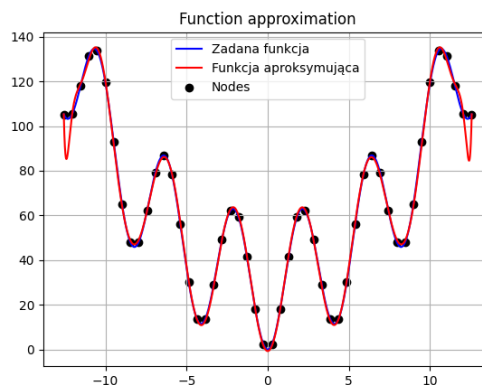
Wykres 26: Wielomian 17 stopnia



Wykres 28: Wielomian 21 stopnia



Wykres 27: Wielomian 18 stopnia



Wykres 29: Wielomian 22 stopnia

Poniżej, na wykresie 30 przedstawione zostały wartości błędów dla wszystkich możliwych stopni wielomianu.

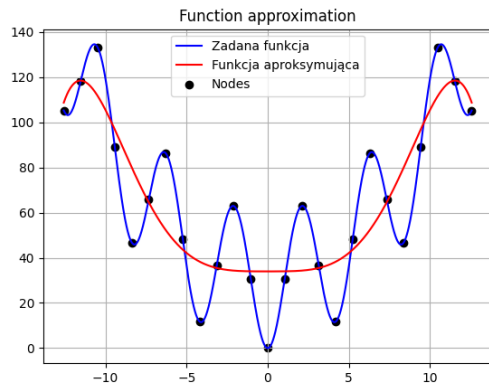


Wykres 30: Wartości błędów

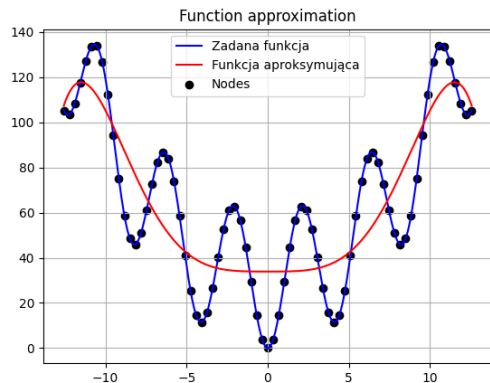
## 8 Przebieg funkcji dla wybranego stopnia wielomianu

### 8.1 Dla 6 stopnia

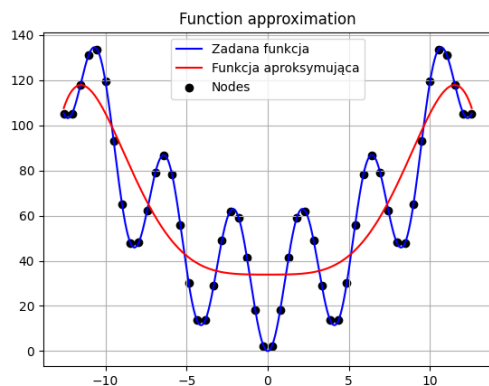
Jak Widać, dla różnej liczby węzłów przebieg funkcji aproksymującej wygląda mniej więcej tak samo, z drobnymi różnicami na krańcach przedziałów (wykres 31, wykres 32, wykres 33, wykres 34)



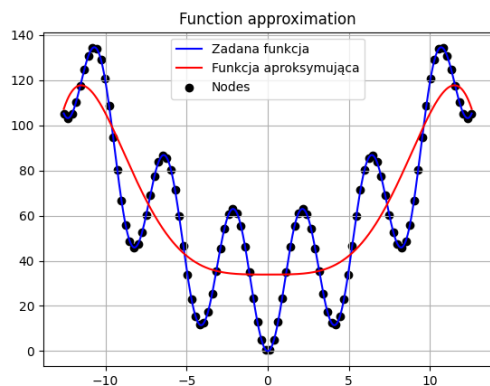
Wykres 31: Dla 25 węzłów



Wykres 33: Dla 75 węzłów

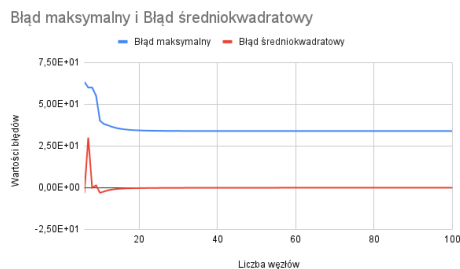


Wykres 32: Dla 50 węzłów



Wykres 34: Dla 100 węzłów

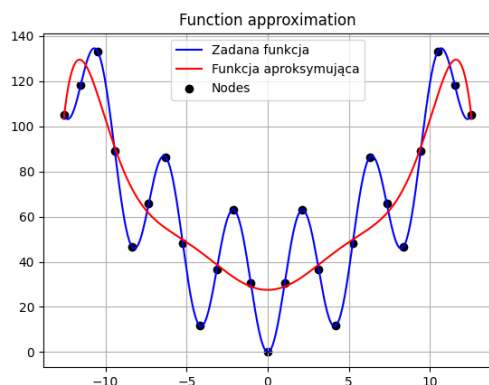
Poniżej, na wykresie 35 przedstawione zostały wartości błędów dla wszystkich możliwych stopni wielomianu.



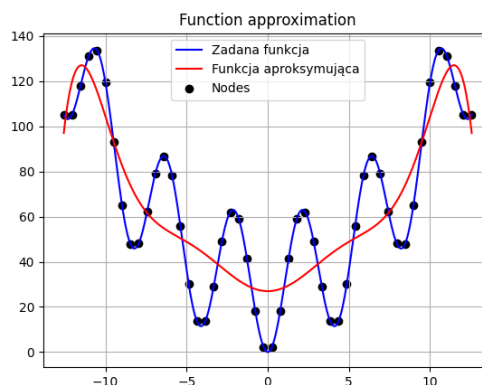
Wykres 35: Wartości błędów

## 8.2 Dla 10 stopnia

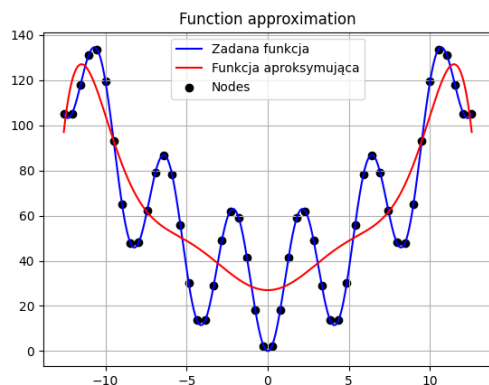
Przybliżenie jest trochę lepsze na krańcach przedziału, jednak dalej funkcja aproksymująca przechodzi przez "środek" zadanej funkcji, co jest nieakceptowalne (wykres 36, wykres 37, wykres 38 i wykres 39).



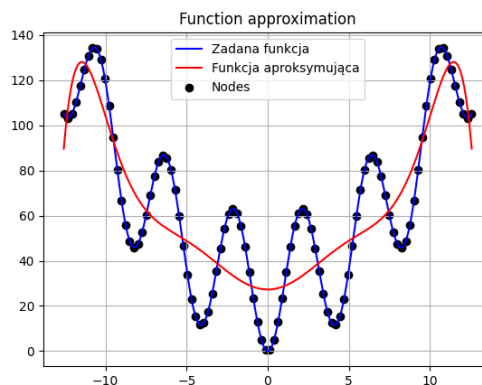
Wykres 36: Dla 25 węzłów



Wykres 38: Dla 75 węzłów

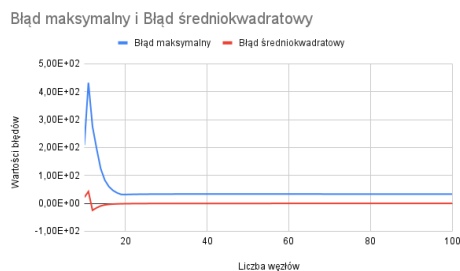


Wykres 37: Dla 50 węzłów



Wykres 39: Dla 100 węzłów

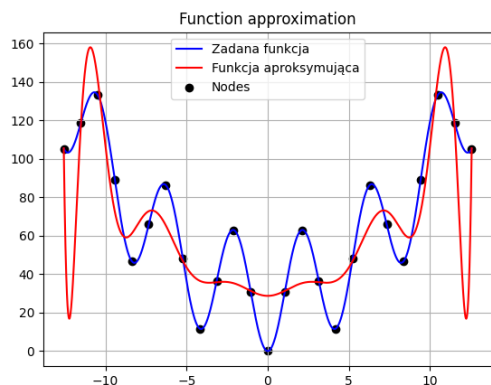
Poniżej, na wykresie 40 przedstawione zostały wartości błędów dla wszystkich możliwych stopni wielomianu.



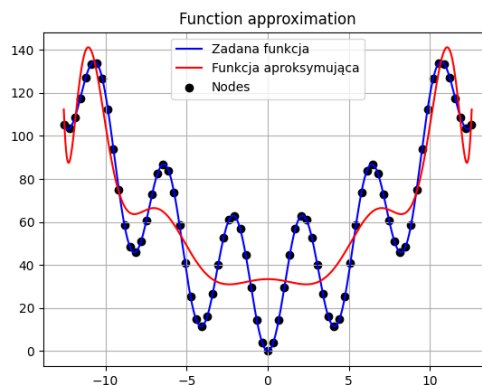
Wykres 40: Wartości błędów

### 8.3 Dla 15 stopnia

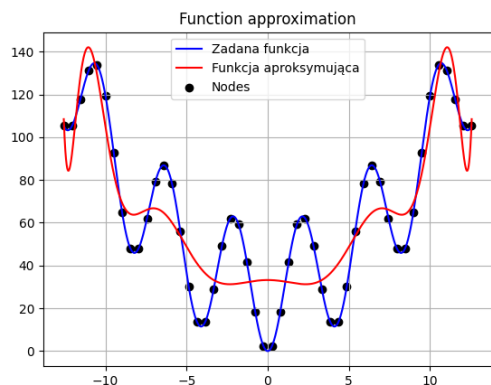
Jak widać na poniższych wykresach (wykres 41, wykres 42, wykres 43, wykres 44) dla 15 stopnia wielomianu otrzymujemy nieco lepsze przybliżenie, jednak dalej nie jest ono zadowalające.



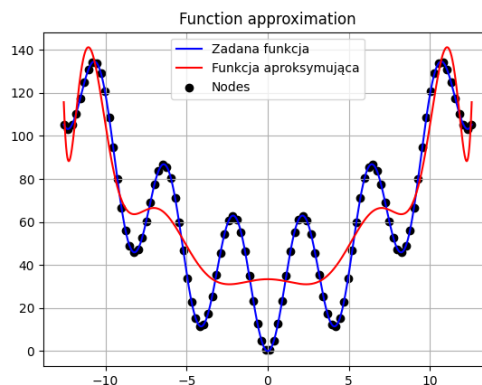
Wykres 41: Dla 25 węzłów



Wykres 43: Dla 75 węzłów

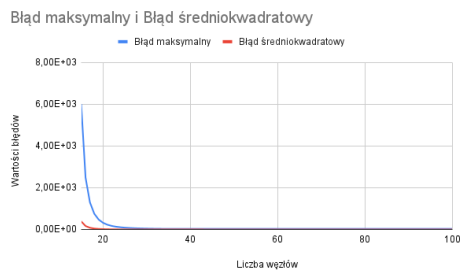


Wykres 42: Dla 50 węzłów



Wykres 44: Dla 100 węzłów

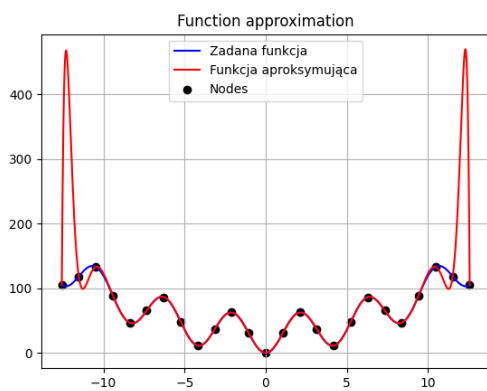
Poniżej, na wykresie 45 przedstawione zostały wartości błędów dla wszystkich możliwych stopni wielomianu.



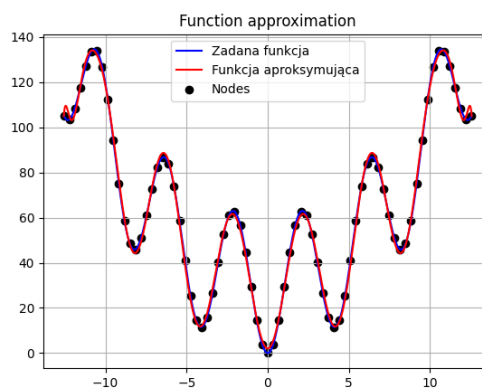
Wykres 45: Wartości błędów

## 8.4 Dla 20 stopnia

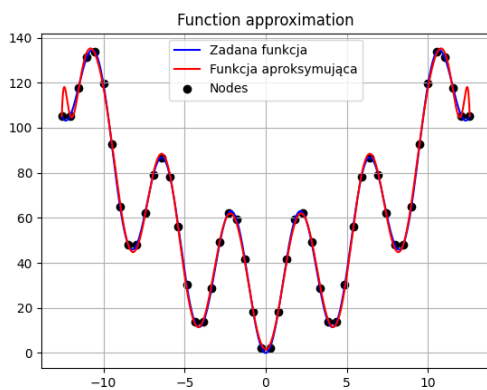
Dla 20 stopnia można już z pewnością powiedzieć, że przybliżenie jest dobre. Szczególnie jest to widoczne dla 100 węzłów, gdzie błędy na krańcach przedziałów są minimalne (wykres 46, wykres 47, wykres 48 i wykres 49).



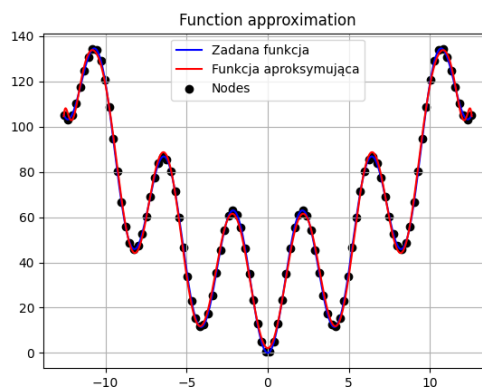
Wykres 46: Dla 25 węzłów



Wykres 48: Dla 75 węzłów



Wykres 47: Dla 50 węzłów



Wykres 49: Dla 100 węzłów

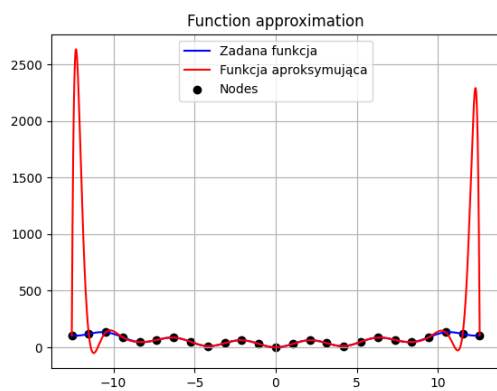
Poniżej, na wykresie 50 przedstawione zostały wartości błędów dla wszystkich możliwych stopni wielomianu.



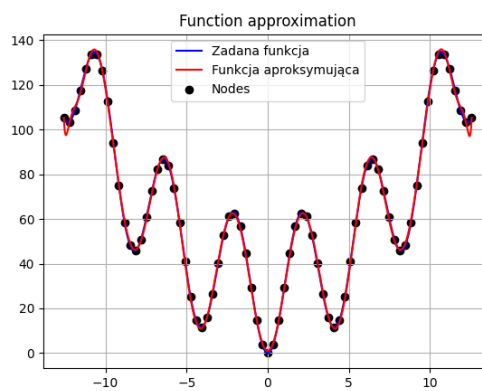
Wykres 50: Wartości błędów

## 8.5 Dla 25 stopnia

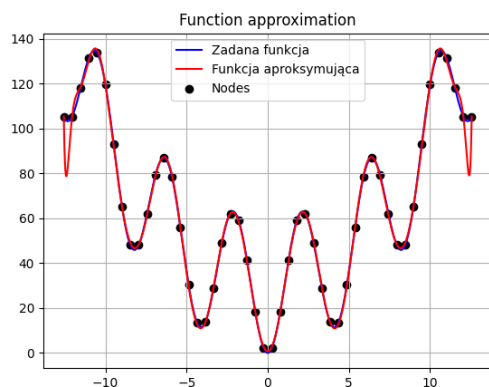
Dla 25 stopnia wielomianu otrzymałem jeszcze lepsze przybliżenie, ponownie wraz z wzrostem liczby węzłów poprawia się przybliżenie, w szczególności maleje błąd na krańcach przedziałów (wykres 51, wykres 52, wykres 53, wykres 54).



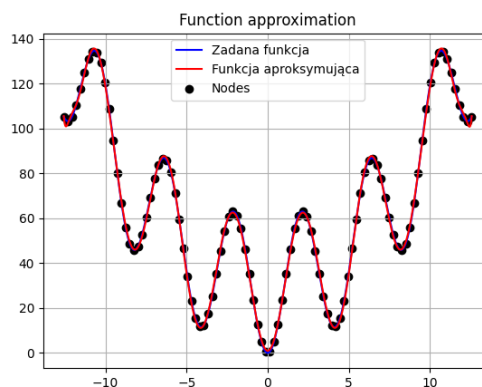
Wykres 51: Dla 25 węzłów



Wykres 53: Dla 75 węzłów



Wykres 52: Dla 50 węzłów



Wykres 54: Dla 100 węzłów

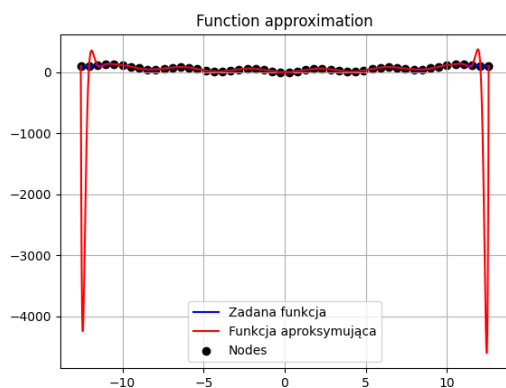
Poniżej, na wykresie 55 przedstawione zostały wartości błędów dla wszystkich możliwych stopni wielomianu.



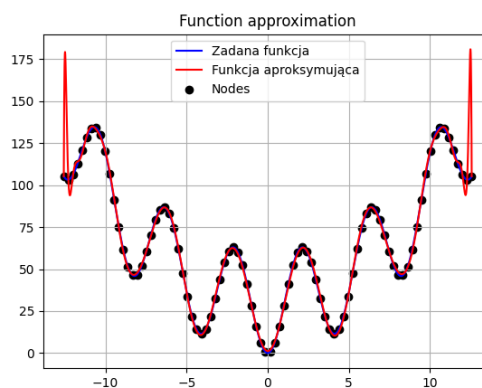
Wykres 55: Wartości błędów

## 8.6 Dla 50 stopnia

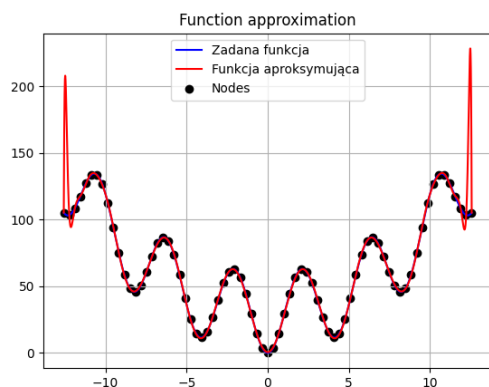
W tym przypadku nadal otrzymałem dobre przybliżenie, jednak z tego powodu, że stopień jest bliski liczbie węzłów występują duże błędy na krańcach przedziałów. Ponownie widać, że wraz z wzrostem liczby węzłów, rośnie dokładność (wykres 56, wykres 57, wykres 58, wykres 59).



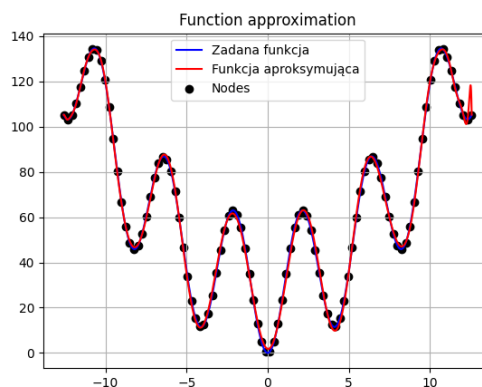
Wykres 56: Dla 50 węzłów



Wykres 58: Dla 90 węzłów

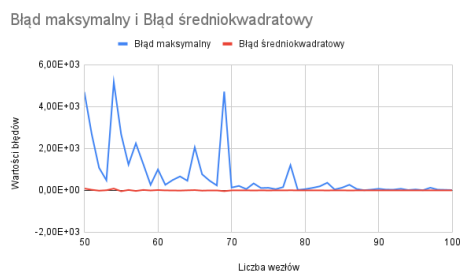


Wykres 57: Dla 75 węzłów



Wykres 59: Dla 100 węzłów

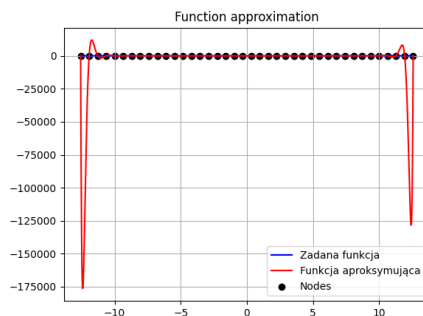
Poniżej, na wykresie 60 przedstawione zostały wartości błędów dla wszystkich możliwych stopni wielomianu.



Wykres 60: Wartości błędów

## 9 Efekt Rungego

Efekt Rungego mógł wystąpić, przy zbliżeniu się do interpolacji, w moim przypadku największy błąd generował dla 40 węzłów i wielomianu 36 stopnia.

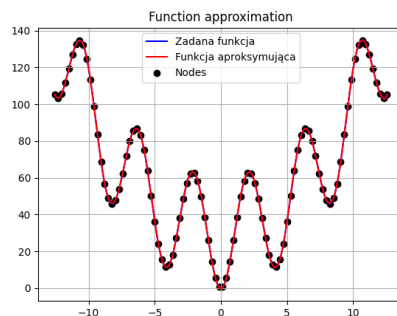


Wykres 61: Najlepsze przybliżenie funkcji

Błąd maksymalny	176413.32885002135
Błąd średniokwadratowy	3832.7665755629732

## 10 Najlepsze przybliżenie funkcji

Najlepsze przybliżenie aproksymowanej funkcji otrzymałem dla 94 węzłów i wielomianu 25 stopnia, został on pokazany na poniższym wykresie, a jego błąd maksymalny i średniokwadratowy poniżej w tabelce. Można powiedzieć, że jest to naprawdę dobre przybliżenie.



Wykres 62: Najlepsze przybliżenie funkcji

Błąd maksymalny	0.5014720569615605
Błąd średniokwadratowy	0.0017181768299556364



## 11 Wnioski

- Zwiększanie stopnia wielomianu dla ustalonej liczby węzłów powoduje zwiększenie dokładności przybliżenia tylko do pewnego stopnia, potem zaczyna się zwiększać efekt Rungego i przybliżenie jest coraz gorsze
- Zwiększanie liczby węzłów dla ustalonego stopnia wielomianu prowadzi do zwiększenia dokładności przybliżenia.
- Wielomiany o dużym stopniu (30, 40, 50, ...) bardzo często stają się coraz mniej dokładne