Metody numeryczne

Nikodem Kocjan

16.04.2024

Interpolacja Lagrange'a funkcji niesymetrycznej z optymalizacją położeń węzłów.

Spis treści

1 Cel ćwiczenia

Celem ćwiczenia jest wyznaczenie współczynników wielomianów interpolacyjnych, zależnie od ilości zadanych punktów, węzłów.

2 Wstęp teorytyczny

2.1 Interpolacja Lagrange'a

Interpolacja Lagrange'a jest klasyczną metodą numeryczną używaną do znajdowania wielomianu interpolacyjnego, który przechodzi przez zadany zestaw punktów danych. Metoda ta jest szczególnie użyteczna, gdy mamy do czynienia z dyskretnymi danymi, które chcemy reprezentować jako ciągłą funkcję. W matematycznej notacji, interpolacja Lagrange'a opisana jest następująco:

$$W_n(x) = \sum_{i=0}^{n} y_i \prod_{\substack{j=0 \\ j \neq i}}^{n} \frac{x - x_j}{x_i - x_j} \tag{1}$$

gdzie x_i to węzły interpolacji, a $y_i = y(x_i)$ to wartości funkcji w tych węzłach.Często jednak, równomierne rozmieszczenie węzłów interpolacji nie jest najbardziej efektywne, szczególnie w przypadkach, kiedy funkcja badana wykazuje skomplikowane zachowanie.

2.2 Zera wielomianów Czebyszewa

W celu zwiększenia efektywności rozważa się alternatywny rozkład węzłów, taki jak zera wielomianów Czebyszewa, które minimalizują maksymalny błąd interpolacji w sensie normy nieskończonej na przedziale. Są one istotnym narzędziem w numerycznej analizie, zwłaszcza w kontekście aproksymacji i interpolacji funkcji.

W poniższym sprawozdaniu została opisana metoda Czebyszewa, w której zera wielomianów wyznaczono na podstawie poniższego wzoru:

$$x_i = \frac{1}{2}\left[(x_b - x_a)\cos\left(\frac{\pi(2i+1)}{2(n+2)}\right) + (x_a + x_b)\right] \tag{2}$$

gdzie x_a i x_b to odpowiednio lewy i prawy brzeg przedziału interpolacji.

2.3 Zastosowanie metody w praktyce

Wielomiany interpolacji wyznaczono na dwa sposoby.

- Równomierne rozłożone węzłów
- Rozłożenie węzłów metodą Czebyszewa

Oba przypadki rozwiązano dla funkcji

$$y(x) = \frac{x}{1+x^2} \tag{3}$$

na przedziale

$$x \in [-3, 8]$$

Problem rozwiązano dla różnych ilości węzłów, $n = 5$, $n = 10$, $n = 15$.

3 Deklaracja zmiennych oraz implementacja bibliotek

Opisane zagadnienie rozwiązano za pomocą programu Visual Studio Code w java script. Wykorzystanie tego języka umożliwiło bezpośrednie generowanie wykresów, bez konieczności zapisywania danych do plików.

Wykorzystano zewnętrzną bibliotekę plotly, do generowania wykresów.

Zadeklarowane stałe, macierze oraz wektory użyte w programie to:

```
    let n = 5;//ilosc wezlow

    let x = []; //wartosci x z krokiem 0.01
    let y = []; //wartosci f(x)

    let x_a = -3; //poczatek przedzialu
    let x_b = 8;  //koniec przedzialu


    //wezly dla rozlozenia regularnego
    let x_r = [];
    let y_r = [];

    //odleglosc miedzy wezlami
    let deltaX = (x_b-x_a) / n;

    //wezly dla metody Czebyszewa
    let x_o = [];
    let y_o = [];

    //wielomian rozlozenia regularnego
    let wielomian = [];
    //wielomian rozlozenia Czebyszewa
    let wielomian_opt = [];

    let L; //licznik
    let M; //mianownik
```

Fragment kodu 1: Deklaracja stalych

4 Opis działania

4.1 Krótki opis programu

Program podzielony jest na 5 pętli for :

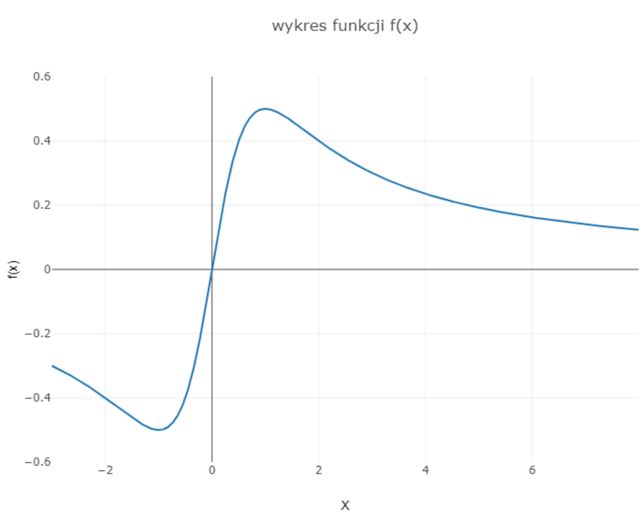
- wyznaczenie x, oraz y dla wygenerowania wykresu
- wyznaczenie węzłów rozłożonych równomiernie
- wyznaczenie wielomianu dla równomiernego rozłożenia węzłów
- wyznaczenie węzłów metodą Czebyszewa
- wyznaczenie wielomianu dla węzłów wyznaczonych metodą Czebyszewa

4.2 Wygenerowanie wykresu podstawowego

```
for(let i =0; i < 1100; i++){
    x[i] = x_a + i / 100;
    y[i] = x[i] / (1 + x[i] * x[i]);
}
```

Fragment kodu 2: Punkty wykresu podstawowego

Pętla uzupełnia tablicę x, zaczynając od wartości początkowej zadanego przedziału. Następnie znając wykres funkcji f(x), wyznaczane są wartości y, dla każdego x. Rezultatem jest funkcja, której wykres przedstawiono poniżej.



Rysunek 1: Wykres funkcji podstawowej w przedziale $[-3, 8]$

4.3 Równomierne rozłożenie węzłów

```
for(let i =0; i <= n; i++){
    x_r[i] = x_a + i * deltaX;
    y_r[i] = x_r[i] / (1 + (x_r[i] * x_r[i]));
}
```

Fragment kodu 3: Wyznacznie węzłów rozłożonych równomiernie

Na podstawie n wyznaczamy zadaną ilość węzłów.

```
      for(let k = 0; k < 1100; k++){
        wielomian[k] = 0;
        for(let i = 0; i <=n; i++){
          L = 1;
          M = 1;
          for(let j =0; j <=n;j++){
            if(i!=j){
              L = L * (x[k]-x_r[j]);
              M = M * (x_r[i] -x_r[j]);
            }
          }

          wielomian[k] = wielomian[k] + ( y_r[i] * (L/M));
        }
      }
```

Fragment kodu 4: Wyznacznie wielomianu

Zewnętrzna pętla "k"odpowiada za wyznaczenie wartości wielomianu dla zadanej wartości x. Aby wykres podstawowy, zgadzał się z wyznaczanym wielomianem, zastosowano taki sam krok, tj. 0,01. Na poczatku każdej pętli ustawiamy wartości wielomian[k] na 0. Jest to przygotowanie do gromadzenia sumy wartości wielomianu.

Środkowa pętla for "i"iteruje od 0 do n. Ta pętla odpowiada za obliczenie i dodanie do sumy wartości i-tego składnika wielomianu Lagrange'a dla punktu x[k]. Wewnątrz zmienne L i M są inicjalizowane na 1. Te zmienne są wykorzystywane do obliczania licznika i mianownika i-tego składnika wielomianu Lagrange'a.

Najbardziej wewnętrzna pętla for "j"iteruje także od 0 do n i odpowiada za obliczanie iloczynów w liczniku (L) i mianowniku (M) składnika wielomianu Lagrange'a, ale pomija przypadki, gdzie i równa się j, ponieważ to odpowiadało by mnożeniu przez zero.

Po wyjściu z najbardziej wewnętrznej pętli, obliczony składnik wielomianu Lagrange'a jest dodawany do sumy w wielomian[k], co jest akumulacją wyniku interpolacji.

4.4 Rozłożenie węzłów metodą Czebyszewa

```
for(let i = 0; i<=n; i++){
    x_o[i] = 0.5 *((x_b-x_a) * Math.cos(Math.PI * (2*i + 1)/ (2*n +
     2)) + (x_a + x_b));
    y_o[i] = x_o[i] / (1 + (x_o[i] * x_o[i]));
}
```

Fragment kodu 5: Wyznacznie węzłów rozłożonych metodą Czebyszewa

W tym przypadku węzły wyznacza się również tylko na podstawie zmiennej n (brzegi liczymy jako wartości stałe) na podstawie wzoru (2).

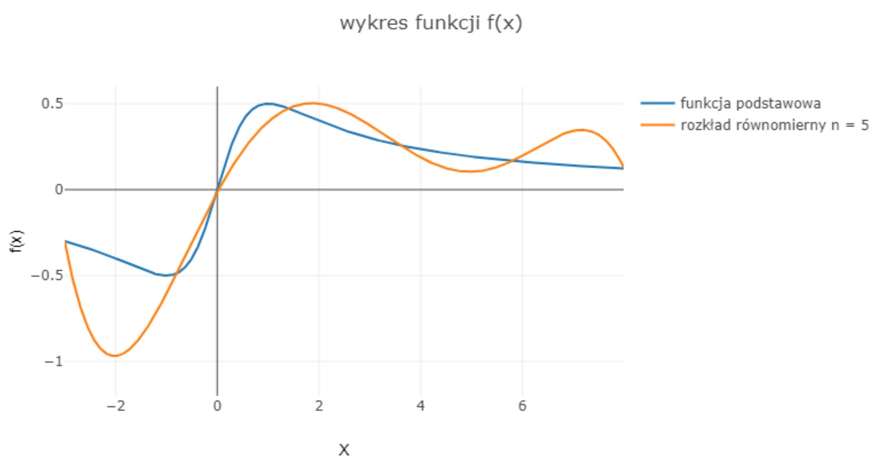
```
      for(let k = 0; k < 1100; k++){
        wielomian_opt[k] = 0;
        for(let i = 0; i <=n; i++){
          L = 1;
          M = 1;
          for(let j =0; j <=n;j++){
            if(i!=j){
              L = L * (x[k]-x_o[j]);
              M = M * (x_o[i] -x_o[j]);
            }
          }

          wielomian_opt[k] = wielomian_opt[k] + ( y_o[i] * (L/M));
        }
      }
```

Fragment kodu 6: Wyznacznie wielomianu dla metody Czebyszewa

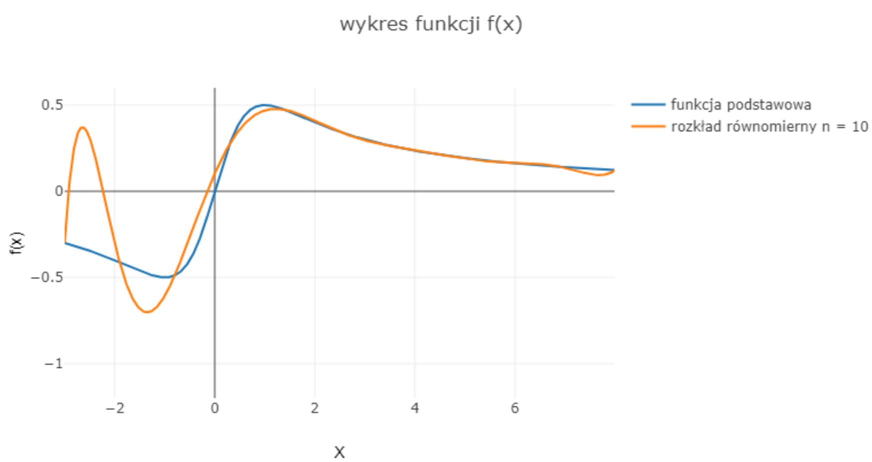
Metoda rozwiązania jest analogiczna jak dla przypadku równomiernego rozłożonia, jedyną zmianą jest zapis do wielomian_opt[k].

5 Analiza wyników

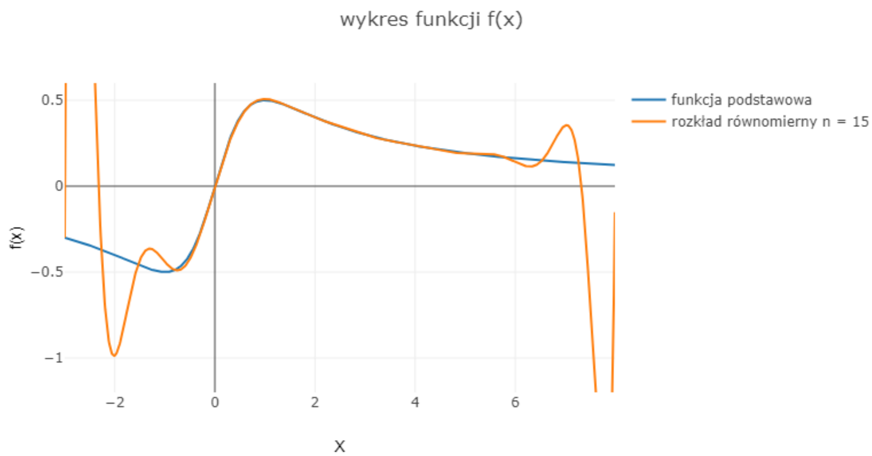
5.1 Równomierne rozłożenie węzłów, dla różnych ilości węzłów



Rysunek 2: Wielomian interpolacji w porównaniu z funkcją podstawową, rozkładu równomiernego oraz ilości węzłów $n = 5$

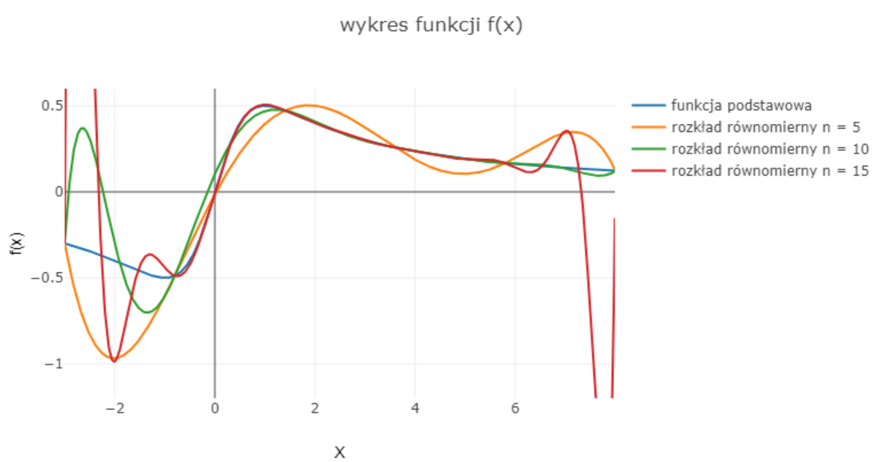


Rysunek 3: Wielomian interpolacji w porównaniu z funkcją podstawową, rozkładu równomiernego oraz ilości węzłów $n = 10$



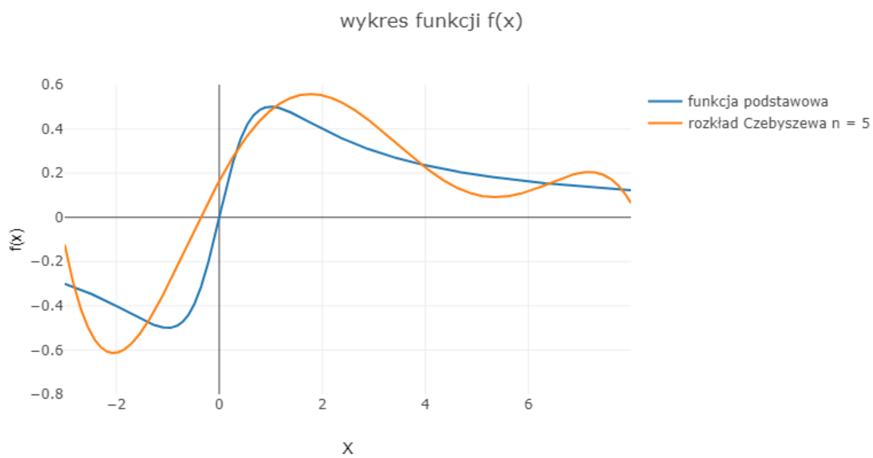
Rysunek 4: Wielomian interpolacji w porównaniu z funkcją podstawową, rozkładu równomiernego oraz ilości węzłów $n = 15$

Warto zwrócić uwagę, że w przedziale od [-3, -2.3] wartości funkcji odbiegają wysoko, a w przedziale [7.4, 8] wartości funkcji odbiegają nisko. W celu lepszej czytelności wykresu, ustawiono zakres osi OY na [-1.2, 0.6]

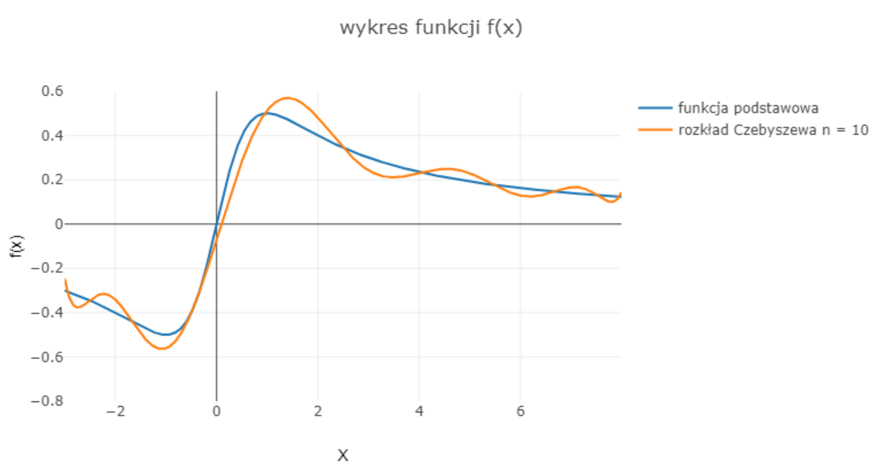


Rysunek 5: Wielomiany interpolacji w porównaniu z funkcją podstawową, rozkładu równomiernego dla ilości węzłów 5, 10, 15

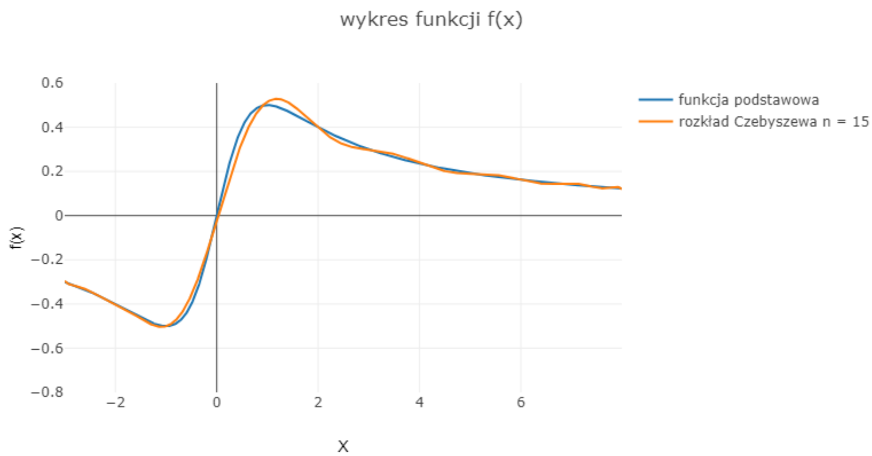
5.2 Rozłożenie węzłów metodą Czebyszewa, dla różnych ilości węzłów



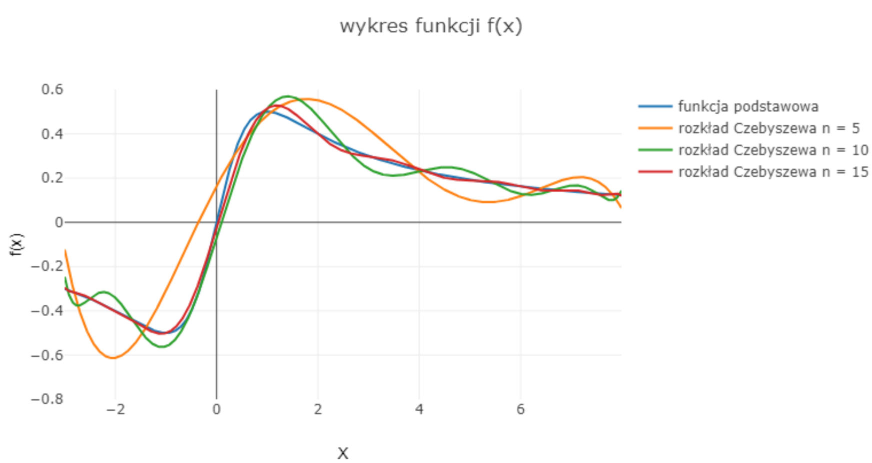
Rysunek 6: Wielomiany interpolacji w porównaniu z funkcją podstawową, rozkładu Czebyszewa dla ilości węzłów $n = 5$



Rysunek 7: Wielomiany interpolacji w porównaniu z funkcją podstawową, rozkładu Czebyszewa dla ilości węzłów $n = 10$

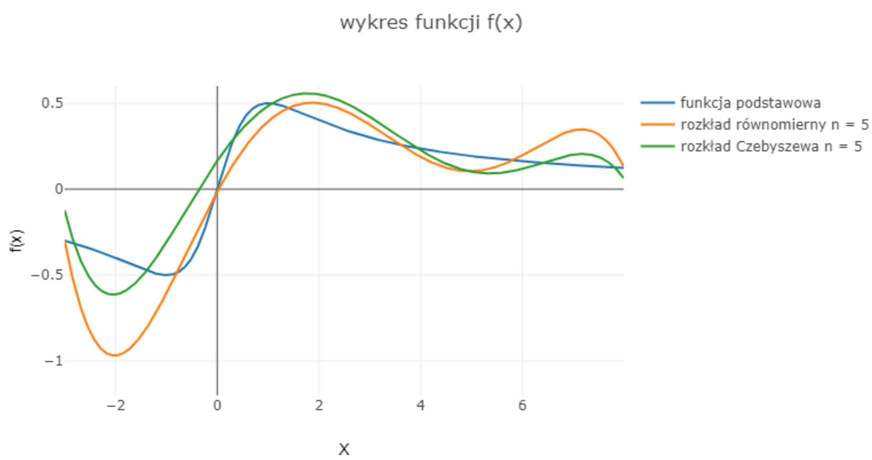


Rysunek 8: Wielomiany interpolacji w porównaniu z funkcją podstawową, rozkładu Czebyszewa dla ilości węzłów $n = 15$

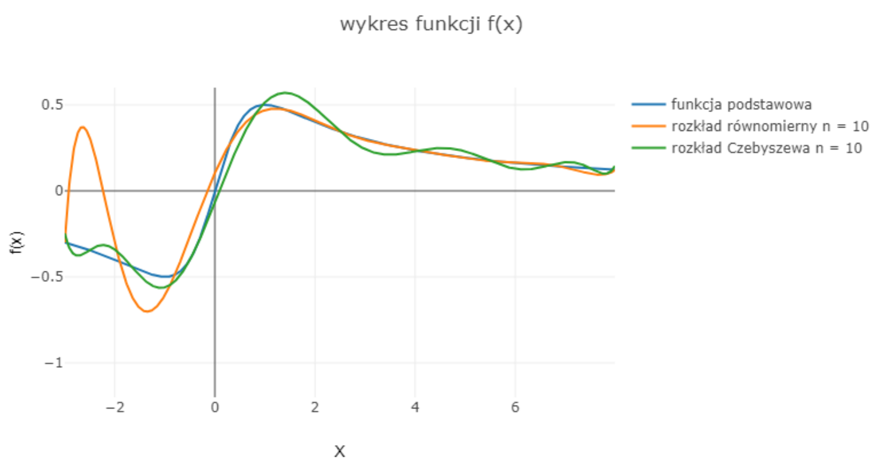


Rysunek 9: Wielomiany interpolacji w porównaniu z funkcją podstawową, rozkładu Czebyszewa dla różnych ilości węzłów nałożone na jeden wykres

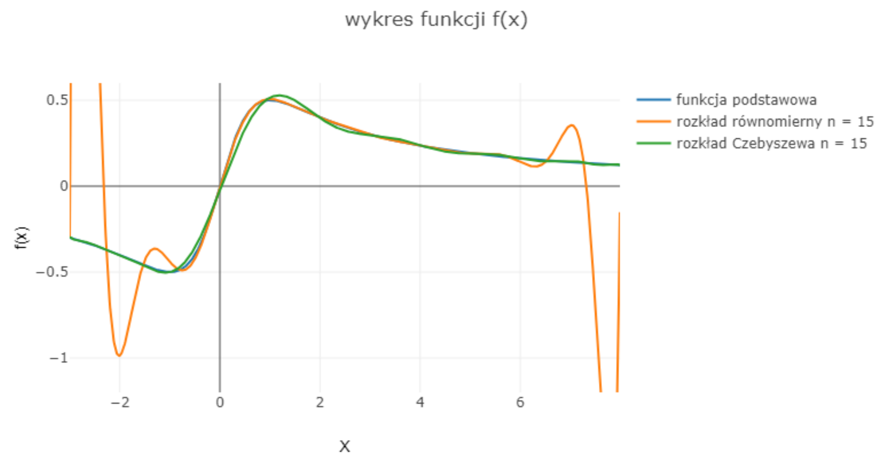
5.3 Porównanie metod dla różnych ilości węzłów



Rysunek 10: Wielomiany interpolacji w porównaniu z funkcją podstawową, rozkładu równomiernego oraz rozkładu Czebyszewa dla $n = 5$



Rysunek 11: Wielomiany interpolacji w porównaniu z funkcją podstawową, rozkładu równomiernego oraz rozkładu Czebyszewa dla $n = 10$



Rysunek 12: Wielomiany interpolacji w porównaniu z funkcją podstawową, rozkładu równomiernego oraz rozkładu Czebyszewa dla $n = 15$

6 Wnioski

Zagęszczanie węzłów metodą rozłożenia równomiernego Analizując wykresy, można zauważyć, że dla większej gęstości rozłożenia węzłów, wielomian interpolacyjny pokrywa funkcję podstawową w dużo większym stopniu. Jednocześnie problemem tej metody okazuje się, że dla ilości węzłów $n = 15$, w kilku przedziałach wykres znaczocą odbiega od funkcji interpolowanej.

Zagęszczanie węzłów metodą rozłożenia Czebyszewa Przypadek rozłożenia węzłów metodą Czebyszewa również zbliża się do funkcji podstawowej wraz z wzrostem gęstości węzłów. Jednocześnie, porównując do równomiernego rozłożenia, nie występują duże odchylenia, co może mieć znaczący wpływ przy analizie wielomianu. Dla $n = 15$ wielomian interpolacji prawie całkowicie pokrywa funkcję podstawową.

Efektywność metody Czebyszewa Metoda Czebyszewa okazała się bardziej efektywna w minimalizacji maksymalnego błędu interpolacji niż tradycyjne, równomierne rozłożenie węzłów. Węzły rozmieszczone według zer wielomianów Czebyszewa skutkowały lepszym przybliżeniem funkcji, co jest istotne przy analizie funkcji z skomplikowanymi zachowaniami.

Znaczenie ilości węzłów Analiza różnych ilości węzłów pokazała, że zwiększanie ich liczby generalnie poprawia dokładność interpolacji dla obu metod, ale szczególnie korzystne efekty daje metoda Czebyszewa, która efektywnie radzi sobie z problemami na brzegach przedziałów.