Metody numeryczne

Nikodem Kocjan

30.04.2024

Aproksymacja Pade funkcji cos(x)

Spis treści

1 Cel ćwiczenia

Celem ćwiczenia jest zastosowania aproksymacji Padé do funkcji trygonometrycznej $\cos(x)$

2 Wstęp teorytyczny

2.1 Aproksymacja Pade

Aproksymacja Padé jest użytecznym narzędziem w analizie numerycznej, pozwalającym na przybliżanie funkcji przez ułamki wymierne. Te przybliżenia często zapewniają lepszą dokładność niż tradycyjne rozwinięcia w szeregi Taylora, szczególnie w przypadku funkcji, które mają osobliwości lub które są badane w szerokim zakresie argumentów.

Dla danej funkcji $f(x)$, aproksymację Padé definiuje się jako ułamek wymierny

$$R_{N,M}(x) = \frac{P_N(x)}{Q_M(x)},$$

gdzie $P_N(x)$ jest wielomianem stopnia N a $Q_M(x)$ jest wielomianem stopnia M z $Q_M(0) = 1$. Wielomiany te są dobierane tak, aby szereg Taylora rozwinięcia $R_{N,M}(x)$ jak najlepiej odpowiadał szeregowi Taylora funkcji $f(x)$ wokół punktu $x = 0$ do jak najwyższego rzędu.

2.2 Zastosowanie aproksymacji Pade w praktyce

W poniższym sprawozdaniu funkcję f(x) przybliżono przy pomocy funkcji wymiernej

$$R_{N,M}(x) = \frac{P_N(x)}{Q_M(x)} = \frac{\sum_{i=0}^{N} a_i x^i}{\sum_{i=0}^{M} b_i x^i} \tag{1}$$

Funkcję rozwinięto w szereg Maclaurina

$$f(x) = \sum_{k=0}^{\infty} c_k x^k \tag{2}$$

Przyrównując pochodne f(x) oraz $R_{N,M}(x)$ dla rzędu $k = 0, 1, \ldots, N+M$ otrzymano układ równań.

$$\left.\frac{d^k R_{N,M}(x)}{dx^k}\right|_{x=0} = \left.\frac{d^k f(x)}{dx^k}\right|_{x=0} \tag{3}$$

$$\sum_{m=1}^{N} b_m \cdot c_{N-m+k} = -c_{N+k}, \quad k = 1, 2, \ldots, N \tag{4}$$

Rozwiązując układ znaleziono współczynniki b słóżące do wyznaczenia współczynników a.

$$a_i = \sum_{j=0}^{i} c_{i-j} \cdot b_j, \quad i = 0, 1, \ldots, N \tag{5}$$

3 Deklaracja zmiennych oraz implementacja bibliotek

Opisane zagadnienie rozwiązano za pomocą programu Visual Studio Code w języku C++. Wykorzystano biblioteki:

- <iostream> - standardowa biblioteka C++
- <cmath> - załączenie kilku funkcji matematycznych
- <vector> - użyty do deklaracji tablic w celu łatwiejszej manipulacji
- <fstream> - zapis do plików

Dodatkowo wykorzystano wiele funkcji oraz metod z biblioteki GSL w celu uproszczenia rozwiązywania układu równań. Zadeklarowane stałe, macierze oraz wektory użyte w programie to:

```
int N = 6; //stopien wielomianu licznika
int M = N; //stopien wielomianu mianownika
vector<double> ctab;  //wektor wspolczynnikow c
gsl_matrix* A = gsl_matrix_alloc(M ,M); //macierz A wykorzystana do
     ukladu rownan
gsl_vector* y = gsl_vector_alloc(M);  //wektor y wykorzystany do
    ukladu rownan
gsl_vector* x = gsl_vector_alloc(M);            //wektor x
vector<double> btab;  //tablica wspolczynnikow b
vector <double> b_odw; //odwrocona tablica b
vector<double> atab; //tablica wspolczynnikow a
double x_begin = -5.0;  //start przedzialu
double x_end = 5.0;   //koniec przedzialu
```

Fragment kodu 1: Deklaracja stalych

4 Opis działania

4.1 Krótki opis programu

Program działa głównie w funkcji main. Utworzono tylko jedną funkcję zewnętrzną.

4.2 Funkcja zewnętrzna

Funkcja zależnie od danego val, oblicza wartość silni. Następnie zwraca ją jako wartość typu integer.

```
int silnia(int val)
{
    int result = (val == 1 || val == 0) ? 1 : silnia(val - 1) * val
    ;
    return result;
}
```

Fragment kodu 2: Obliczanie silni

4.3 Funkcja main

Jako pierwsze z pomocą algorytmu wyznaczono wartości współczynników tablicy C

```
for (int k = 0; k < n + 1; k++) {
  double c;
  if (k % 2 == 0) {
    if (k % 4 == 0) {
      c = 1.0 / silnia(k);
    }
    else {
      c = -1.0 / silnia(k);
    }
  }
  else {
    c = 0;
  }
  ctab.push_back(c);
}
```

Fragment kodu 3: Wyznacznie wartości współczynników c

Na podstawie wartości tablicy C, uzupełniono macierz A, potrzebną do rozwiązania układu równań

```
for (int i = 0; i < M; i++) {
  for (int j = 0; j < M; j++) {
    double value = ctab[N - M + i + j + 1];
    gsl_matrix_set(A, i, j, value);
  }
}
```

Fragment kodu 4: Uzupelnienie macierzy A

Uzupelniono wektor y potrzebny do ukladu rownan

```
for (int i = 0; i < M; i++) {
  double value = -ctab[N + 1 + i];
  gsl_vector_set(y, i, value);
}
```

Fragment kodu 5: Uzupelnienie wektora y

Z pomocą biblioteki GSL rozwiazano uklad rownan. Rezultat zapisano do wektora x

```
gsl_permutation* p = gsl_permutation_alloc(M);
int signum;
gsl_linalg_LU_decomp(A, p, &signum);
gsl_linalg_LU_solve(A, p, y, x);
```

Fragment kodu 6: Rozwiazanie ukladu rownan

Zgodnie z algorytmem, zapisano rezultaty w odwrotnej kolejnosci do tablicy b_odw, oraz na jej podstawie uzupelniono tablice b_tab.

```
for (int i = 0; i < M; i++) {
  b_odw.push_back(gsl_vector_get(x, i));
}
```

```
for (int i = M - 1; i >= 0; i--) {
  btab.push_back(b_odw[i]);
}
```

Fragment kodu 7: Zapisanie rezultatow w odwrotnej kolejnosci

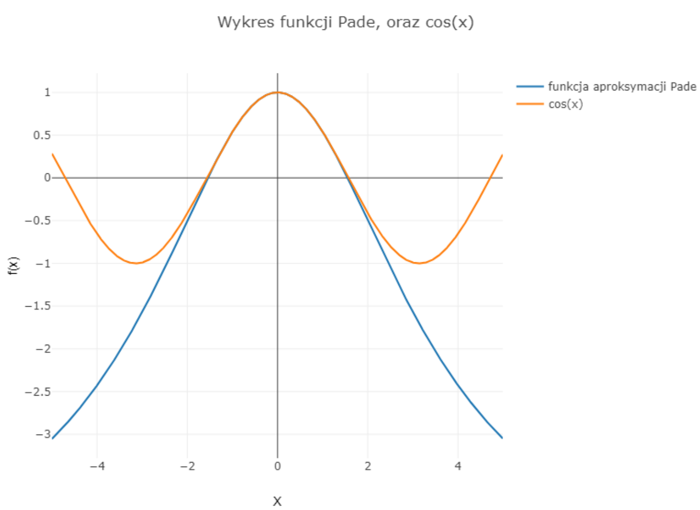
Na końcu wyznaczono wartości współczynników a wielomianu, oraz zapisano je do tablicy.

```
for (int i = 0; i <= N; i++) {
  double a = 0;
  for (int j = 0; j <= i; j++) {
    a += ctab[i - j] * btab[j];
  }
  atab.push_back(a);
}
```

Fragment kodu 8: Uzupelnienie tablicy a

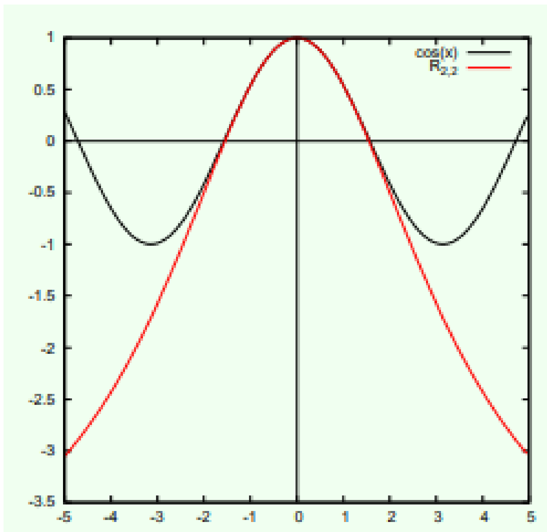
5 Analiza wyników

5.1 Stopień macierzy $N = 2$



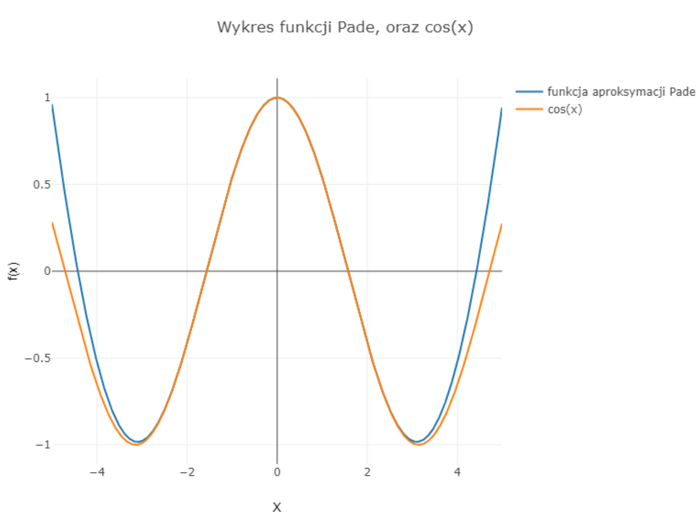
Rysunek 1: Wykres przedstawia nałożone na siebie funkcje cos(x), oraz funkcję aproksymacji Pade dla $N = 2$

Oczekiwane rezultaty:



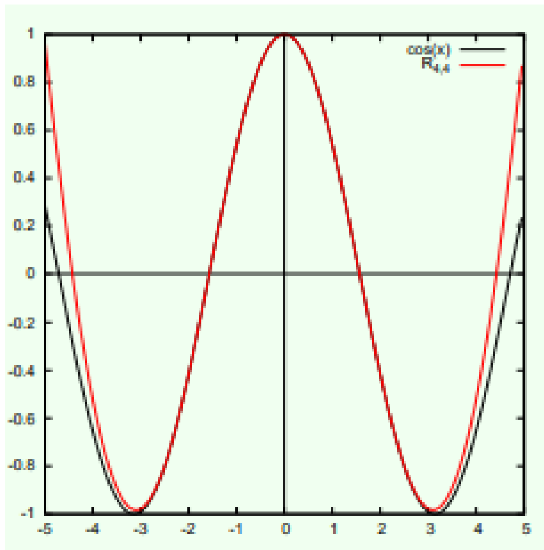
Rysunek 2: Wykres przedstawia oczekiwane rezultaty dla $N = 2$

5.2 Stopień macierzy $N = 4$



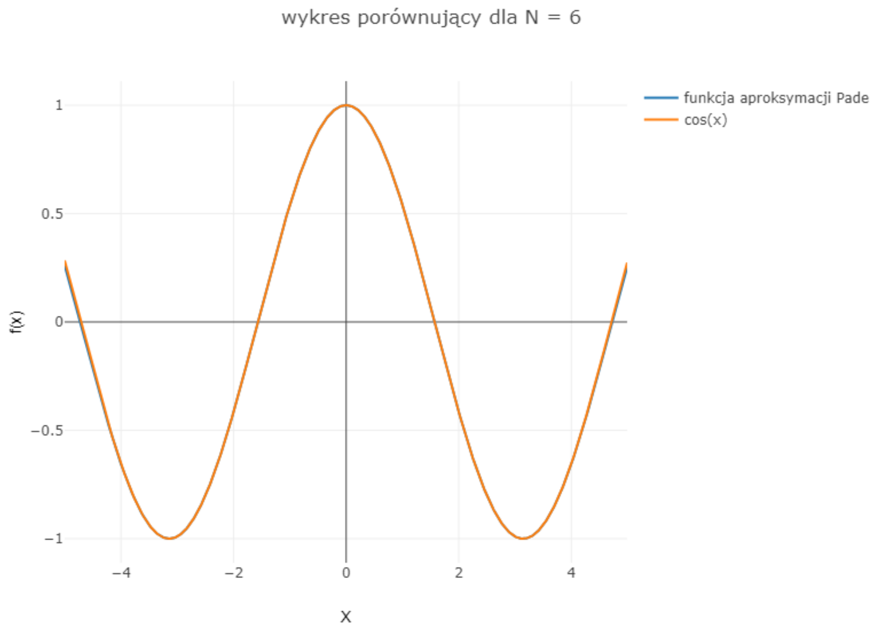
Rysunek 3: Wykres przedstawia nałożone na siebie funkcje cos(x), oraz funkcję aproksymacji Pade dla $N = 4$

Oczekiwane rezultaty:



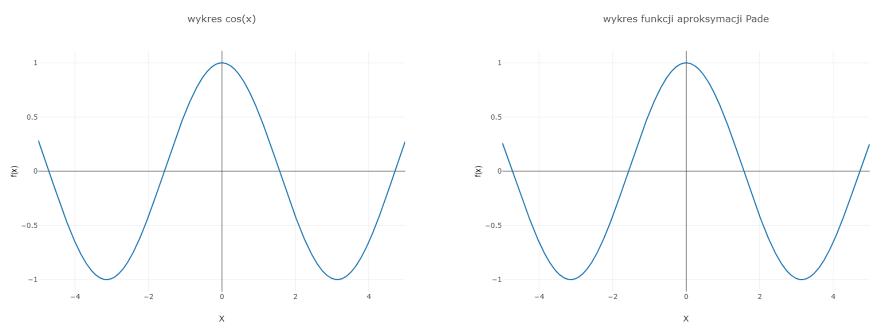
Rysunek 4: Wykres przedstawia oczekiwane rezultaty dla $N = 4$

5.3 Stopień macierzy $N = 6$



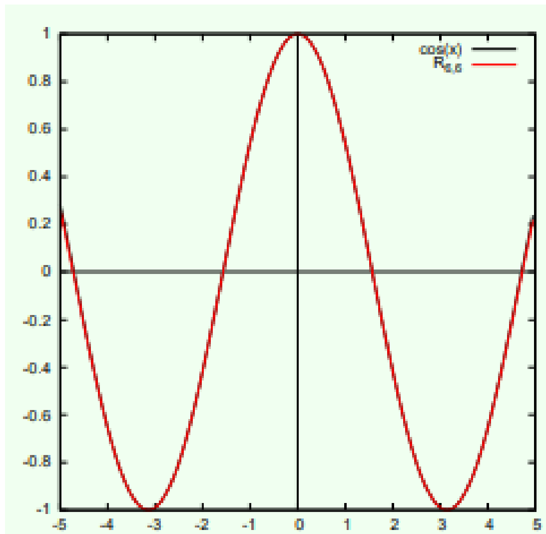
Rysunek 5: Wykres przedstawia funkcję cos(x), oraz funkcję aproksymacji metodą Pade nałożone na jeden wykres

Funkcje niemal na całej długości pokrywają się, dlatego poniżej przedstawiono rezultaty na osobnych wykresach.



Rysunek 6: Wykres z lewej przedstawia funkcję cos(x), a z prawej funkcję aproksymacji metdoą Pade

Oczekiwane rezyltaty:



Rysunek 7: Wykres przedstawia oczekiwane rezultaty dla $N = 6$

6 Wnioski

Aproksymacja Pade jest skutecznym narzędziem w przybliżaniu funkcji, szczególnie w przypadkach, gdy tradycyjne szeregi Taylora są niewystarczające lub mają ograniczone zastosowanie. W szczególności, metoda ta pozwala na uzyskanie dobrej dokładności nawet w szerszym zakresie wartości, co zostało potwierdzone podczas generowania wykresów.

Różne stopnie wielomianów Dla N = 2 i N = 4 wyniki aproksymacji były zadowalające, mimo że dla N = 2 zauważono rozbieżność przy granicach przedziałow. Najlepszą zgodność z funkcją $\cos(x)$ uzyskano dla N = 6. To pokazuje, że zwiększenie stopnia wielomianów w aproksymacji Padé może znacząco poprawić dokładność przybliżenia.

Problematyczność Warto również zwrócić uwagę na to, że choć metoda Padé jest bardzo użyteczna, może wymagać zastosowania zaawansowanych technik numerycznych i obliczeniowych do wyznaczania odpowiednich wielomianów, co może być czasochłonne i zazwyczaj bardzo skomplikowane.