



AKADEMIA GÓRNICZO-HUTNICZA W KRAKOWIE

WYDZIAŁ FIZYKI I INFORMATYKI STOSOWANEJ

Metody Numeryczne

Laboratorium 09: Aproksymacja Pade funkcji
 $\cos(x)$

Andrzej Świętek

23.04.2024

Contents

1	Wstęp teoretyczny	2
1.1	Ułamki Padégo	2
1.2	Metoda rozwiązania	2
1.3	Zalety i Zastosowania	4
2	Problem	5
3	Implementacja	6
4	Wyniki	8
4.1	$N = 2$ i $M = 2$	9
4.2	$N = 4$ i $M = 4$	9
4.3	$N = 6$ i $M = 6$	9
5	Wizualizacja Wyników	10
6	Analiza Wyników	12
6.1	$N = 2$ i $M = 2$	12
6.2	$N = 4$ i $M = 4$	12
6.3	$N = 6$ i $M = 6$	12
7	Wnioski	12

1. Wstęp teoretyczny

Aproksymacja Padégo jest techniką stosowaną do przybliżania funkcji przez ułamki wymierne. Jest to szczególnie użyteczne w przypadkach, gdy tradycyjne metody interpolacji wielomianowej lub szeregów skończonych nie dają satysfakcjonujących wyników, szczególnie w obszarach, gdzie funkcja może mieć osobliwości, np. bieguny lub zbieżności asymptotyczne.

1.1. Ułamki Padégo

Ułamki Padégo to specjalny rodzaj ułamków, które przybliżają funkcje jako ilorazy dwóch wielomianów. Mają one postać:

$$R_{M,N}(x) = \frac{P_M(x)}{Q_N(x)}$$

gdzie $P_M(x)$ i $Q_N(x)$ to wielomiany stopnia M i N , odpowiednio. Ułamek Padégo $R_{M,N}(x)$ jest rzeczywistą aproksymacją funkcji rzeczywistej w zakresie x , jeśli:

1. $P_M(x)$ i $Q_N(x)$ nie mają wspólnych pierwiastków rzeczywistych w zakresie x .
2. Stopień wielomianu $P_M(x)$ wynosi co najwyżej M .
3. Stopień wielomianu $Q_N(x)$ wynosi co najwyżej N .
4. $M + N$ jest zawsze równy stopniowi szeregu Maclaurina, który chcemy przybliżyć.

1.2. Metoda rozwiązania

Rozwiązanie problemu aproksymacji Padégo sprowadza się do znalezienia współczynników $P_M(x)$ i $Q_N(x)$, które najlepiej przybliżają funkcję w danym zakresie.

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

z $b_0 = 1$

ALGORYTM:

1. **Wybór stopni wielomianów:** Określ stopnie wielomianów N i M dla licznika i mianownika ułamka Padégo.
2. **Obliczenie pochodnych:** Oblicz pochodne funkcji aproksymowanej $f(x)$ w punkcie $x = 0$ do stopnia $N + M$. Pochodne te będą wykorzystane do wyznaczenia współczynników c_k .
3. **Wyliczenie współczynników c_k :** Skaluj obliczone pochodne przez odpowiednie wartości silni i zapisz je jako współczynniki c_k .
4. **Rozwiązanie układu równań:** Skonstruuj układ równań liniowych $Ax = y$, gdzie A to macierz współczynników związanych z mianownikiem, a y to wektor wartości pochodnych. Rozwiąż ten układ równań, na przykład za pomocą metody dekompozycji LU.
5. **Obliczenie współczynników b_i :** Po rozwiązaniu układu równań, przypisz uzyskane wartości do współczynników b_i mianownika.
6. **Obliczenie współczynników a_i :** Korzystając z obliczonych wcześniej współczynników b_i oraz współczynników c_k , wyznacz współczynniki a_i licznika.
7. **Konstrukcja funkcji aproksymującej:** Zbuduj funkcję aproksymującą $R_{N,M}(x)$ używając wyznaczonych współczynników a_i i b_i .
8. **Ocena dokładności:** Zweryfikuj dokładność aproksymacji, porównując wartości funkcji aproksymującej z wartościami funkcji pierwotnej w odpowiednio dobranym przedziale.
9. **Opcjonalnie: Poprawki i optymalizacje:** W razie potrzeby można wprowadzić dodatkowe poprawki lub optymalizacje algorytmu, na przykład przez zwiększenie stopni wielomianów, zmianę punktów aproksymacji itp.
10. **Analiza wyników:** Dokładnie przeanalizuj uzyskane wyniki, w tym stabilność numeryczną, dokładność i zgodność z oczekiwaniami. W przypadku potrzeby, podejmij działania korygujące lub dostosowawcze.

1.3. Zalety i Zastosowania

Aproksymacja Padégo ma kilka zalet w porównaniu do tradycyjnych metod aproksymacji:

1. **Dobra aproksymacja w obszarach singularności:** Aproksymacja Padégo radzi sobie dobrze z funkcjami, które mają osobliwości, takie jak bieguny czy zbieżności asymptotyczne.
2. **Ekonomiczność obliczeniowa:** W przypadku, gdy funkcja jest złożona lub kosztowna w obliczeniach, ułamki Padégo mogą dać dobrą aproksymację z mniejszym nakładem obliczeń niż tradycyjne metody.
3. **Szerokie zastosowanie:** Metoda ta jest stosowana w wielu dziedzinach nauki i inżynierii, w tym w fizyce, matematyce, inżynierii chemicznej itp.

W skrócie, aproksymacja Padégo jest użyteczną techniką przybliżania funkcji, szczególnie w przypadkach, gdy tradycyjne metody zawodzą. Dzięki możliwości aproksymacji funkcji w obszarach singularności i szybkości obliczeń, jest to popularne narzędzie w analizie numerycznej i inżynierii.

2. Problem

Wykonanie aproksymacji Padego funkcji

$$f(x) = \cos(x)$$

kolejno dla $N = M = 2, 4, 6$. W tym celu wykonujemy następujące kroki

1. Ustalamy $N = n + m$, i liczymy pochodne $f^{(k)}(0)$ dla $k = 0, 1, 2, \dots, n$

$$\left. \frac{d^k \cos(x)}{dx^k} \right|_{x=0} = \begin{cases} f^{(2p)}(0) = (-1)^p & \text{dla } p = 0, 1, 2, 3, \dots \\ f^{(2p+1)}(0) = 0 & \text{dla } p = 0, 1, 2, 3, \dots \end{cases}$$

Współczynniki c_k to skalowane pochodne ze wzoru Taylora i należy wyznaczyć je ze wzoru:

$$c_k = \frac{f^{(k)}(0)}{k!}$$

Wartości współczynników c_k zachowujemy w wektorze $\vec{c} = [c_0, c_1, \dots, c_n]$

2. Rozwiązujemy układ danych równań używając biblioteki GSL

$$A \cdot \vec{x} = \vec{y}$$

gdzie:

$$A_{i,j} = c_{N-M+i+j+1}, \quad i, j = 0, 1, \dots, M-1$$

$$y = -c_{N+1+i}, \quad i = 0, 1, \dots, M-1$$

po rozwiązaniu układu równań zachowujemy współczynniki wielomianu $Q_M(x)$

$$b_0 = 1 \quad \text{oraz} \quad b_{M-i} = x_i, \quad i = 0, 1, 2, \dots, M-1$$

Współczynniki zapisujemy w wektorze $\vec{b} = [b_0, b_1, \dots, b_M]$

3. Wyznaczamy współczynniki wielomianu $P_N(x)$ zgodnie z wzorem

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

Zapisujemy je do wektora $\vec{a} = [a_0, a_1, \dots, a_N]$

4. Dla ustalonego n tworzymy wykresy $f(x)$ oraz $R_{N,M}(x)$ na jednym rysunku w zakresie $x \in [-5, 5]$.

3. Implementacja

```
1 void calculate_derivatives(double* derivatives, int n){
2     for (int k = 0; k <= n; ++k) {
3         if(k % 2 == 0) {
4             derivatives[k] = pow(-1, k/2);
5         } else {
6             derivatives[k] = 0;
7         }
8     }
9 }
```

Listing 1: Implementacja funkcji Wyznaczającej kolejne pochodne w punkcie $x_0 = 0$

```
1 constexpr long long factorial(int n) {
2     return n <= 1 ? 1 : n * factorial(n-1);
3 }
```

Listing 2: Implementacja silni

```
1 double R_NM(double x, double *a, double *b, int N, int M){
2     double sum_a = 0.0;
3     double sum_b = 0.0;
4     for (int i = 0; i <= N; ++i) sum_a += a[i] * pow(x, i);
5     for (int i = 0; i <= M; ++i) sum_b += b[i] * pow(x, i);
6     return sum_a / sum_b;
7 }
```

Listing 3: Implementacja funkcji aproksymującej $R_{N,M}(x)$

```
1 void calculate_vector_c(double*c, double* f_k, int n) {
2     cout << "\nc_k:\n";
3     for (int k = 0; k < n+1; ++k) {
4         c[k] = f_k[k] / factorial(k);
5         cout << "c_" << k << " = " << c[k] << endl;
6     }
7 }
```

Listing 4: Wyznaczenie wektora c_i

```
1 void calculate_vector_a(double*a, double* b, double* c, int N) {
2     fill(a, a + N + 1, 0.0);
3     for (int i = 0; i <= N; i++) {
4         double sum = 0.0;
5         for (int j = 0; j <= i; j++)
6             sum += c[i - j] * b[j];
7
8         a[i] = sum;
9     }
10 }
```

Listing 5: Wyznaczenie wektora z współczynnikami a_i

```

1 // Funkcja rozwarzajaca uklad rownan
2 void solve_system(double* c, double* f_k, double* b, int N, int M) {
3     gsl_matrix* A = gsl_matrix_alloc(M, M);
4     gsl_vector* x = gsl_vector_alloc(M);
5     gsl_vector* y = gsl_vector_alloc(M);
6     gsl_permutation* p = gsl_permutation_alloc(M);
7
8     // Inicjalizacja macierzy A i wektora y
9     for (int i = 0; i < M; i++) {
10         gsl_vector_set(y, i, -f_k[N + 1 + i]);
11         for (int j = 0; j < M; j++) {
12             double value = c[N - M + i + j + 1];
13             gsl_matrix_set(A, i, j, value);
14         }
15     }
16
17     print_gsl_matrix(A);
18
19     // Rozwiazanie ukladu rownan
20     int signum;
21     gsl_linalg_LU_decomp(A, p, &signum);
22     gsl_linalg_LU_solve(A, p, y, x);
23
24     for( int i =0; i <= (int)x->size; i++ ) i == (int)x->size? cout << "\n" :
25     cout << gsl_vector_get(x, i) << " ";
26
27     // Przypisanie rozwiazania do wektora b
28     b[0] = 1.0;
29     vector<double> bk, reversed;
30     for (int i = 0; i < M; i++)
31         reversed.push_back(gsl_vector_get(x, i));
32
33     for (int i = M - 1; i >= 0; i--)
34         bk.push_back(reversed[i]);
35
36     for (int i = 0; i < bk.size(); ++i) {
37         b[i+1] = bk[i];
38     }
39
40     // Zwolnienie pamieci
41     gsl_matrix_free(A);
42     gsl_vector_free(x);
43     gsl_vector_free(y);
44     gsl_permutation_free(p);
45 }

```

Listing 6: Wyznaczenie wektora z współczynnikami b_i


```

1 // Stopnie wielomianow Q_M(x) i P_N(x)
2 int M = 4;
3 int N = 4;
4
5 // Stopien pochodnych
6 int n = M+N;
7
8 // Inicjalizacja tablicy na przechowywanie pochodnych
9 double f_k[n+1];
10 double c[n+1];
11
12 calculate_derivatives(f_k, n);
13
14 cout << "Pochodne funkcji cos(x) w punkcie x = 0:\n";
15 for (int k = 0; k <= n; ++k) {
16     cout << "f^(" << k << ")(0) = " << f_k[k] << endl;
17 }
18
19 calculate_vector_c(c, f_k, n);
20
21 // Rozwiazanie ukladu rownan => wektor b
22 solve_system(c, c, b, N, M);
23
24 // vector a
25 double a[n+1] = {0};
26 calculate_vector_a(a, b, c, N);
27
28 ofstream file(
29     "data/approximation_results_" + to_string(N) + "_" + to_string(M) + ".csv"
30 );
31 file << "x y\n";
32 for (double x = -5.0; x <= 5.0; x += 0.1 ){
33     double approximated = R_NM(x, a, b, N, M);
34     file << x << " " << approximated << endl;
35 }
36 file.close();

```

Listing 7: Wywołanie wszystkich funkcji (main)

4. Wyniki

Tabele prezentują wartości poszczególnych elementów wektorów a_i i b_i stanowiących współczynniki dla wielomianów w liczniku i mianowniku głównego ułamka. Dodatkowa kolumna x_i prezentuje rozwiązanie układu liniowego uzyskanego z rozwiązania metodą LU.

4.1. $N = 2$ i $M = 2$

Table 1: Wartości wektorów x oraz współczynniki wielomianów $Q_M(x)$ i $P_N(x)$ dla wartości $N = 2$, $M = 2$.

i	a_i	b_i	x_i
0	1	1	0.0833333
1	0	0	0
2	-0.416667	0.0833333	—

4.2. $N = 4$ i $M = 4$

Table 2: Wartości wektorów x oraz współczynniki wielomianów $Q_M(x)$ i $P_N(x)$ dla wartości $N = 4$, $M = 4$.

i	a_i	b_i	x_i
0	1	1	0.000859788
1	0	0	0
2	-0.456349	0.0436508	0.0436508
3	0	0	0
4	0.0207011	0.000859788	—

4.3. $N = 6$ i $M = 6$

Table 3: Wartości wektorów x oraz współczynniki wielomianów $Q_M(x)$ i $P_N(x)$ dla wartości $N = 6$, $M = 6$.

i	a_i	b_i	x_i
0	1	1	3.23554×10^{-6}
1	0	0	0
2	-0.470596	0.0294042	0.000423729
3	0	0	0
4	0.0273883	0.000423729	0.0294042
5	0	0	0
6	-0.000372342	3.23554×10^{-6}	—

5. Wizualizacja Wyników

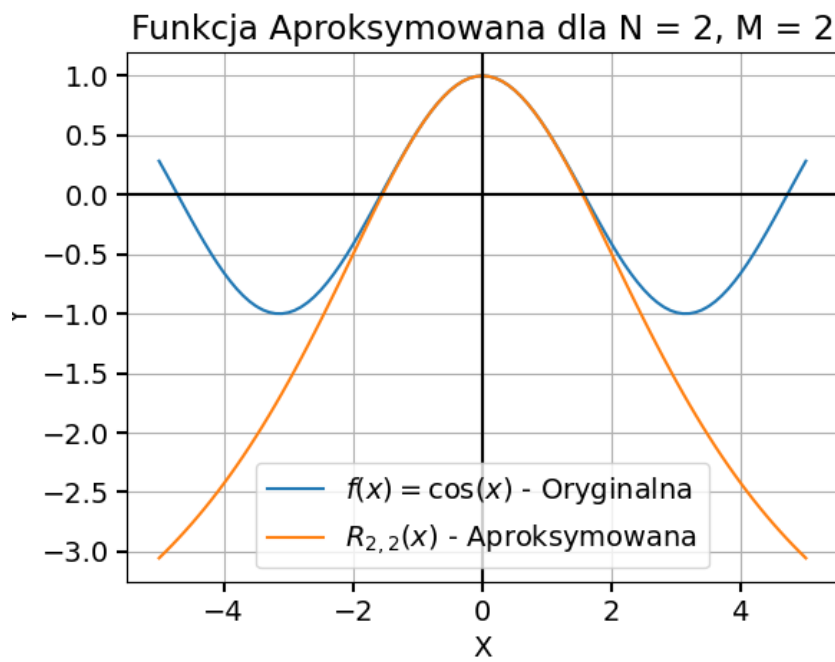


Figure 1: Wartość funkcji aproksymowanej dla $N = 2$ i $M = 2$

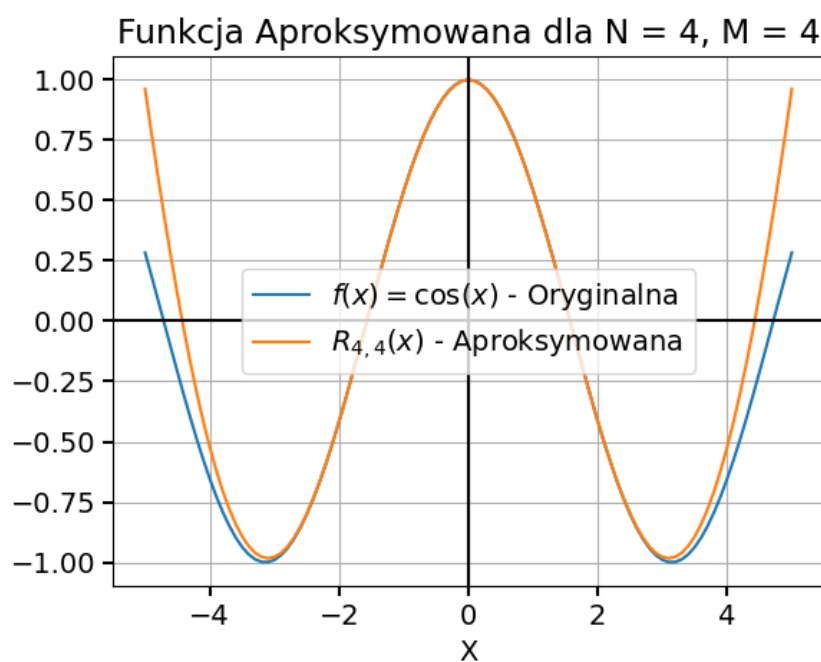


Figure 2: Wartość funkcji aproksymowanej dla $N = 4$ i $M = 4$

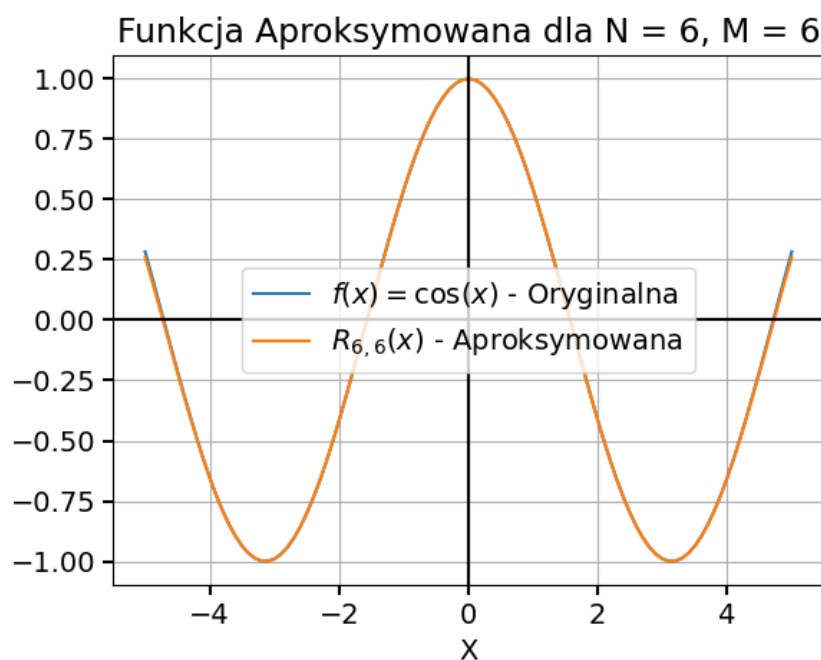


Figure 3: Wartość funkcji aproksymowanej dla $N = 6$ i $M = 6$

6. Analiza Wyników

6.1. $N = 2$ i $M = 2$

Dla wielomianów stopnia drugiego zarówno w liczniku jak i mianowniku funkcja aproksymująca będąca rozwiązaniem daje wyniki dalekie od ideału. W przedziale $[-2, 2]$ przybliżenie jest dość dokładne a wykres prawie całkowicie pokrywa się z wykresem funkcji aproksymowanej ($\cos(X)$). Niestety dla argumentów z poza tego przedziału funkcja nie zachowuje się jak cosinus a jak klasyczny wielomian gdzie maleje do nieskończoności. Funkcja też jest parzysta tak jak cosinus.

6.2. $N = 4$ i $M = 4$

Czwarty stopień wielomianu licznika jak i mianownika pozwala z dużą precyzją przybliżyć przebieg funkcji $\cos(x)$. W większej części przedziału funkcja aproksymowana pokrywa się zupełnie z aproksymującą. Niewielki odchył od poprawnego wyniku jest zauważalny w przedziałach $x \in [-5, -3]$ i $x \in [3, 5]$. Dla lewego skraju przedziału błąd maleje a w prawej na skutek aproksymacji funkcja aproksymowana rośnie nieznacznie szybciej niż oryginalna. Wynik ten może już być dostatecznie dobry do zastosowań praktycznych i wykorzystania wyniku w dalszych pracach jeśli czynnik precyzji na krańcach przedziału nie jest tak istotny. Podsumowując funkcja aproksymowana jest podobna zarówno pod względem przyjmowanych wartości, monotoniczności i całościowego przebiegu do oryginału.

6.3. $N = 6$ i $M = 6$

Dla wielomianów stopnia szóstego w liczniku i mianowniku w zadanym przedziale funkcja aproksymowana jest identyczna jak oryginał. Wykresy pokrywają się całkowicie. Tak wynik (w tym przedziale) można uznać za bardzo dokładny i pozwalający go wykorzystywać w dalszych obliczeniach bez obaw o narastający błąd.

7. Wnioski

Czynniki wpływające na dokładność wyników:

- Stopień wielomianów N i M : Wyższe stopnie wielomianów mogą prowadzić do dokładniejszych wyników, ale również zwiększają złożoność obliczeniową. Odpowiedni dobór stopni N i M jest kluczowy.
- Przedział aproksymacji: Istotne jest, aby przedział, na którym przeprowadzamy aproksymację, był odpowiednio dobrany. Poza tym przedziałem funkcja aproksymująca może znacznie odbiegać od funkcji pierwotnej.
- Osobliwości funkcji: Metoda Padégo jest szczególnie skuteczna w obszarach, gdzie funkcja posiada osobliwości, takie jak bieguny lub zbieżności asymptotyczne. W tych obszarach tradycyjne metody mogą zawodzić.
- Stabilność numeryczna: Zapewnienie stabilności numerycznej podczas obliczeń jest kluczowe, aby uniknąć błędów zaokrągleń i numerycznych.

Potencjalne błędy i uwarunkowanie:

- Błędy obliczeniowe: Ze względu na ograniczenia precyzji obliczeń numerycznych, mogą występować błędy zaokrągleń, błędy numeryczne podczas rozwiązywania układów równań, czy błędy wynikające z niedokładności obliczeń.
- Uwarunkowanie układu równań: Metoda aproksymacji Padégo polega na rozwiązaniu układu równań liniowych. Jeśli macierz tego układu jest źle uwarunkowana, może to prowadzić do niestabilności numerycznej i błędów wynikających z niedokładności obliczeń.
- Singularności funkcji: W przypadku funkcji posiadających osobliwości, takich jak bieguny, konieczne jest ostrożne manipulowanie tymi punktami, aby uniknąć błędów numerycznych.
- Zbieżność szeregów Padégo: Nie zawsze można zagwarantować zbieżność szeregów Padégo do funkcji pierwotnej. W niektórych przypadkach może być konieczne zastosowanie różnych technik i poprawek.

Dokładność i efektywność metody aproksymacji Padégo zależy więc od wielu czynników, w tym od właściwego dobrania stopni wielomianów, odpowiedniego uwarunkowania układu równań, stabilności numerycznej oraz precyzji obliczeń.