Metody Numeryczne - Projekt 1

Mikołaj Mróz, Paweł Swiderski

14 grudnia 2022

Projekt nr 1: Metoda Simspona obliczania przybliżonej wartości całki $\int_a^b w_n(x)\,dx$, gdzie w_n jest wielomianem reprezentowanym w bazie wielomianów Czebyszewa II-ego rodzaju:

$$w_n(x) = \sum_{k=0}^{n} a_k U_k(x)$$

Projekt nr 2: Metoda Newtona ("trzech ósmych") obliczania przybliżonej wartości całki $\int_a^b w_n(x)\,dx$, gdzie w_n jest wielomianem reprezentowanym w bazie wielomianów Czebyszewa II-ego rodzaju:

$$w_n(x) = \sum_{k=0}^{n} a_k U_k(x)$$

Spis treści

1 Opis metod

1.1 Metoda Simspona

Pierwszą metodą z której będziemy korzystać w tym projekcie będzie złożona metoda Simpsona. W tej metodzie mamy 2 przedziały, oraz 3 węzły: a, b i a+b/2, gdzie a to węzeł x0 a b to węzeł xn. Kwadratura na nich skonstruowana to kwadratura (wzór) Simpsona, zwana również wzorem parabol. Funkcja f jest przybliżana wielomianem stopnia 2. W tym przypadku kwadratura ma postać :

$$S(f) = \frac{b-a}{6}(f(a) + f(b) + 4f(\frac{a+b}{2}))$$

Złożona metoda Simpsona polega na wybraniu n+1 równo odległych od siebie punktów na przedziale całkowania. Następnie na każdym z uzyskanych przedziałaów stosujemy metodę Simpsona oraz sumujemy uzyskone wartości. Wtedy przybliżona wartość szukanej całki oznaczonej wynosi:

$$S(f) = \frac{H}{6}(f(a) + f(b) + 2\sum_{k=1}^{N-1} f(a+kH) + 4\sum_{k=0}^{N-1} f(a+kH+H/2))$$

Błąd złożonej kwadratury Simpsona wynosi:

$$E(f) = \frac{-1}{180 * 2^4} H^4(b-a)f^{(4)}(\mu)$$

dla pewnego $\mu \in (a, b)$

Idea metody Simpsona:
Metoda Simpsona polega na policzeniu całki z welomianu, który jest oparty na równoodległych punktach należących do funkcji, z której chcemy policzyć całkę.

1.2 Metoda Newtona (”trzech ósmych”)

Drugą wykorzystaną metodą będzie metoda Newtona, zwana inaczej metodą ”trzech ósmych”. Metoda ”trzech ósmych” jest bardzo podobna do metody Simpsona z tym że wykorzystuje ona interpolację wielomianem trzeciego stopnia, a nie drugiego. Następnie postępujemy tak samo jak poprzednio, aby wyprowadzić wzór na złożoną wersję tej metody. Przybliżona wartość szukanej całki oznaczonej wynosi:

$$S(f) = \frac{3H}{8}[f(a) + f(b) + 3\sum_{k=1}^{N/3}(f(a+(3k-2)H) + f(a+(3k-1)H)) + 2\sum_{k=1}^{N/3-1} f(a+3kH)]$$

Błąd złożonej kwadratury Newtona wynosi:

$$E(f) = \frac{-1}{80} H^4(b-a)f^{(4)}(\mu)$$

dla pewnego $\mu \in (a, b)$

1.3 Wielomiany Czebyszewa II-ego rodzaju

Wielomianami Czebyszewa drugiego rodzaju nazywamy wielomiany postaci:

$$U_n(x) := \frac{sin((n+1)arccosx)}{sin(arccosx)}$$

$$x \in (-1, 1), n \in N$$

lub ogólnie:

$$U_0(x) = 1, U_1(x) = 2x, U_n(x) = 2xU_{n-1}(x) - U_{n-2}(x)$$

Zatem wielomiany Czebyszewa są zdefiniowane za pomocą trójtermowego równania rekurencyjnego. Z tej włąsności będziemy korzystać pry obliczaniu jego wartości.

1.4 Algorytm Clenshawa

Aby użyć złożonej metody Simpsona musimy najpierw policzyć wartość wielomianu $w_n(x) = \sum_{k=0}^{n} a_k U_k(x)$. Nie możemy oczywiście sprowadzać go do postaci naturalnej, aby obliczyć jego wartość skorzystamy z jego rekurencyjnej własności. W tym celu użyjemy algorytmu Clenshawa, który jest rekurencyjną metodą obliczania liniowej kombinacji wielomianów Czebyszewa. Stosuje się go do dowolnej klasy funkcji definiowalnych za pomocą trójtermowego równania rekurencyjnego. Zatem możemy go użyć też w tym przypadku. Algorytm Clenshawa działa następująco: Algorytm Clenshawa do obliczania skończonej sumy wielomianów Czebyszewa jest dany wzorem:

Algorytm Clenshawa do obliczania wartości wielomianu reprezentowanego w bazie wielomianów Czebyszewa II-ego stopnia
p - wektor współczynników a_n liniowej kombinacji wielomianów Czebyszewa
x - punkt x, w którym chcemy obliczyć wartość wielomianu

function Clenshaw(p,x)

$b_{n+2} = b_{n+1} = 0$
for j = n:-1:1
$b_j = 2xb_{j+1} - b_{j+2} + a_j$
end
return $xb_1 - b_2 + a_0$

2 Opis funkcjonalności metody w Matlabie

2.1 Cel zadania i sposób jego rozwiązania

Celem zadania było zaimplementowanie całkowania metodą Simspona wielomianów reprezentowanych w bazie wielomianów Czebyszewa II-ego stopnia $w_n(x) = \sum_{k=0}^{n} a_k U_k(x)$.

1. Obliczenie wartości wielomianu
Zrobiliśmy to za pomocą algorytmu Clenshawa, który pozwala obliczyć wartość wielomianu reprezentowanego w bazie wielomianów Czebyszewa II-ego stopnia korzystając z rekurencyjnej własności.
2. Zmodyfikowanie metody Simspona (przystosowanie do algorytmu Clenshawa i wektoryzacja względem n)
Zmodyfikowaliśmy algorytm Simspona, przez co jako argumenty podajemy kombinację liniową wielomianów Czebyszewa II-ego stopnia, początek i koniec przedziału, oraz liczbę przedziałów(która może być podana jako wektor).
3. Porównanie dokładnych wartości całki z wielomianu $w_n(x)$ z wartości całek uzyskanych metodą Simpsona.
W tej części uzyskaliśmy różnice między dokładną i przybliżoną wartością całki z wielomianu.
4. Sprawdzenie jak ilosć przedziałów wpływa na przybliżenie różnych wielomianów
Na koniec zwizualizowaliśmy dokładności przybliżeń dla różnej ilości przedziałow dla ciekawych przypadków wielomianów i wyciągneliśmy wnioski, o których napisaliśmy na końcu pracy po dogłębnej analizie. Do sprawdzenia wykorzystaliśmy funkcje CoefficientsExact oraz ExactIntegral aby otrzymać dokładną wartość całki z naszego wielomianu, który jest reprezentowany za pomocą wielomianów Czebyszewa stopnia II-ego.

2.2 Funkcje zawarte w programie

2.2.1 Funkcja Clenshaw

```
function ret = Clenshaw(p,x)
% ret = Clenshaw(p,x)
% Algorytm Clenshawa do obliczania wartosci wielomianu reprezentowanego
% w bazie wielomianow Czebyszewa II-ego stopnia
% p - wektor wspolczynnikow a_k liniowej kombinacji wielomianow Czebyszewa
% x - punkt x, w kt rym chcemy obliczyc wartosc wielomianu
n = length(p) - 1;
b = zeros(n+2,1);
b(n+2) = 0;
b(n+1) = 0;
for j = n:-1:1
    b(j) = 2*x*b(j+1) - b(j+2) + p(j+1);
end
ret = 2*x*b(1) - b(2) + p(1); %wynik otrzymany algorytmem Clenshawa
```

2.2.2 Funkcja Simpson

```
function res = Simpson(p,a,b,n)
% res = simpson(f,a,b,n)
% Calkowanie Zlozona kwadratura Simpsona
% p - wektor wspolczynnikow a_k z liniowej kombinacji wielomianow
% Czebyszewa
% a,b - krance przedzialu calkowania
% n - ilosc podprzedzialow
res = zeros(length(n),2);
for i = 1:length(n)
    n_iter = n(i);
    h = (b-a)/n_iter;
    v1 = zeros(1,n_iter);
    v2 = zeros(1,n_iter);
    for j = 1:1:n_iter
        v1(j) = Clenshaw(p,a+(j-1)*h);
        v2(j) = Clenshaw(p,a+(j-1)*h+h/2);
    end
    res(i,1) = n_iter;
    res(i,2) = h/6*(Clenshaw(p,a) + Clenshaw(p,b) + 2*sum(v1(2:end)) + 4*sum(v2
        (1:end)));%obliczenie calki
end
```

2.2.3 Funkcja Newton

```
function ret = NewtonApprox(p, a, b, n)
    ret = [length(n),2];
    for i = 1:length(n)
        n_iter = n(i);
        h = (b-a)/n_iter;
        m = n_iter/3;
        v1 = zeros(1,m);
        v2 = zeros(1,m);
        for j = 1:1:m
            v1(j) = Clenshaw(p,a+(3*(j)-2)*h) + Clenshaw(p, a+(3*(j)-1)*h);
            v2(j) = Clenshaw(p,a+3*(j)*h);
        end
        ret(i,1) = n_iter;
        ret(i,2) = (3/8)*h*(Clenshaw(p,a) + 3*sum(v1(1:end)) + 2*sum(v2(1:m-1))
            + Clenshaw(p,b));
    end
```

2.2.4 Funkcja CoefficientsExact

```
function res = CoefficientsExact(p)
% res = CoefficientsExact(p)
% Funkcja zwracajaca wspolczynniki wielomianu w postaci naturalnej
% p - wektor wspolczynnikow a_k z liniowej kombinacji wielomianow
% Czebyszewa
syms x
res = 0;
n = length(p);
for i = 1:n
    res = res + p(i)*chebyshevU(i-1,x);
end
res = sym2poly(res);
```

2.2.5 Funkcja IntegralExact

```
function res = IntegralExact(p,a,b)
% res = IntegralExact(p,a,b)
% Funkcja zwracajaca dokladna wartosc calki na przedziale (a,b) dla p
% a, b - granice calkowania
% p - wektor wspolczynnikow a_k z liniowej kombinacji wielomianow
% Czebyszewa
coeff = CoefficientsExact(p);
q = polyint(coeff);
res = diff(polyval(q,[a b]));
```

3 Ciekawe przykłady, wizualizacja i analiza wyników

Oznaczenia:
a_k - współczynniki liniowej kombinacji wielomianów Czybyszewa II-ego rodzaju
$w_n(x)$ - wielomian którego całkę oznaczoną przybliżamy
a, b - granice całki

3.1 1. przykład

$a_0 = 5$ $a_1 = -3$
$w_n(x) = -6x + 5$
$a = -7, b = 9.6$

Dokładna wartość: $\int_{-7}^{9.6} -6x + 5\, dx = -46.48$

N	Metoda Simpsona	Metoda Newtona	Błąd Simposn	Błąd Newton
3	-46.48	-46.48	0	0
6	-46.48	-46.48	0	0
9	-46.48	-46.48	0	0
12	-46.48	-46.48	0	0
15	-46.48	-46.48	0	0

3.2 2. przykład

$a_0 = 2$ $a_1 = 7$ $a_2 = 3$
$w_n(x) = 12x^2 + 14x - 1$
$a = -1.7, b = 2.3$

Dokładna wartość: $\int_{-1.7}^{2.3} 12x^2 + 14x - 1\, dx = 81.12$

N	Metoda Simpsona	Metoda Newtona	Błąd Simpson	Błąd Newton
3	81.12	81.12	0	0
6	81.12	81.12	0	0
9	81.12	81.12	0	0
12	81.12	81.12	0	0
15	81.12	81.12	0	0

3.3 3. przykład

$a_0 = 6$ $a_1 = -10$ $a_2 = 21$ $a_3 = -12$ $a_4 = -4$
$w_n(x) = -64x^4 - 96x^3 + 132x^2 + 28x - 19$
$a = -3, b = 1$

Dokładna wartość: $\int_{-1.7}^{2.3} -64x^4 - 96x^3 + 132x^2 + 28x - 19\, dx = -159.2$

N	Metoda Simpsona	Metoda Newtona	Błąd Simposn	Błąd Newton
3	-165.9424	-401.9259	6.7424	242.7259
9	-159.2832	-162.1966	0.0832	2.9966
30	-159.2007	-159.2243	0.0007	0.0243
90	-159.2000	-159.2003	0	0.0003
150	-159.2000	-159.2000	0	0

3.4 4. przykład

$a_0 = 1$ $a_1 = -1$ $a_2 = 1 a_3 = -1$ $a_4 = 1$ $a_5 = -1 a_6 = 1$ $a_7 = -1$
$w_n(x) = -128x^7 + 64x^6 + 160x^5 - 64x^4 - 56x^3 + 16x^2 + 4x$
$a = 0, b = 1$

Dokładna wartość: $\int_0^1 -128x^7 + 64x^6 + 160x^5 - 64x^4 - 56x^3 + 16x^2 + 4x\,dx = \frac{12}{35} \approx 0.3429$

N	Metoda Simpsona	Metoda Newtona	Błąd Simpson	Błąd Newton
3	0.2990	-0.2469	0.0439	0.5898
9	0.3425	0.3232	0.0004	0.0197
21	0.3428	0.3422	0.0001	0.007
51	0.3429	0.3428	0	0.0001
999	0.3429	0.3429	0	0

3.5 5. przykład

$a_0 = 3$ $a_1 = -4$ $a_2 = 5$ $a_3 = 4$ $a_4 = 0$ $a_5 = -5$ $a_6 = -1$ $a_7 = 1$
$w_n(x) = 128x^7 - 64x^6 - 352x^5 + 80x^4 + 272x^3 - 4x^2 - 62x - 1$
$a = -2.7, b = 2.2$

Dokładna wartość: $\int_{-2.7}^{2.2} 128x^7 - 64x^6 - 352x^5 + 80x^4 + 272x^3 - 4x^2 - 62x - 1\,dx = -31045.7$

N	Metoda Simpsona	Metoda Newtona	Błąd Simpson	Błąd Newton
3	-33244.9	-59979.5	2199.2	28933.8
9	-31075.0	-32033	29.3	987.3
21	-31046.6	-31081.1	0.9	35.4
51	-31045.7	-31046.7	0.0287	1
102	-31045.7	-31045.7	0	0

3.6 6. przykład

$a_0 = 0$ $a_1 = 0$ $a_2 = 0$ $a_3 = 0$ $a_4 = 0$ $a_5 = 0$ $a_6 = 0$ $a_7 = 0$ $a_8 = 0$ $a_9 = 0$ $a_{10} = 0$ $a_{11} = 0$ $a_{12} = 0$ $a_{13} = 0$ $a_{14} = 0$ $a_{15} = 0$ $a_{16} = 7$ $a_{17} = 0$ $a_{18} = 0$ $a_{19} = 0$ $a_{20} = 0$ $a_{21} = 0$ $a_{22} = 0$ $a_{23} = 1 * 10^-18$
$w_n(x) = 0.0084x^{23} - 0.0461x^{21} + 0.1101x^{19} - 0.1494x^{17} + 0.1270x^{15} - 0.0702x^{13} + 0.0253x^{11} - 0.0059x^9 + 0.0008x^7 - 0.0001x^5$
$a = -1, b = 4$

Dokładna wartość: $\int_{-1}^{4} 0.0084x^{23} - 0.0461x^{21} + 0.1101x^{19} - 0.1494x^{17} + 0.1270x^{15} - 0.0702x^{13} + 0.0253x^{11} - 0.0059x^9 + 0.0008x^7 - 0.0001x^5\,dx = 67.0043$

N	Metoda Simpsona	Metoda Newtona	Błąd Simposn	Błąd Newton
3	117.0584	259.5081	50.0541	192.5038
9	69.0051	93.9434	2.0008	26.9391
51	67.0067	67.0865	0.0024	0.0822
240	67.0043	67.0045	0	0.0002
360	67.0043	67.0043	0	0

4 Wnioski i analiza

Obie metody zwracają dokładną wartość całki gdy $w_n(x)$ jest stopnia co najwyżej trzeciego. Gdy wielomian jest wyższego stopnia zaczynają pojawiać się błędy względem dokładnej wartości. Błąd ten zależy od N, wybranej liczby podprzedziałów. Dla niewielkich N, błąd jest duży, a dla coraz większych, staje on się coraz mniejszy, aż osiąga wartość niemal równą zero. Generalnie błąd metody Newtona jest większy niż błąd metody Simpsona, a szczególnie dla $N = 3$, najmniejszego możliwego N dla metody Newtona, gdzie błąd metody Newtona w wielu przypadkach jest bardzo duży. Na podstawie przykładów widzimy, że obie metody dla dostatecznie dużego N są bardzo dokładne. Radzą sobie one bardzo dobrze z przybliżaniem całek, gdy funkcja podcałkowa jest wielomianem.

5 Literatura

-Hao Jiang, Roberto Barrio, Housen Li, Xiangke Liao, Lizhi Cheng, Fang Su, Accurate evaluation of a polynomial in Chebyshev form, Applied Mathematics and Computation, Volume 217, Issue 23

-https://en.wikipedia.org/wiki/Simpson%27s_rule

-https://en.wikipedia.org/wiki/Chebyshev_polynomials

-https://handwiki.org/wiki/Clenshaw_algorithmSpecial_case_for_Chebyshev_series