Sprawozdanie
Metody Numeryczne 2
Temat 4

Przemysław Woźniakowski

2018-12-12

1 Treść zadania

Aproksymacja średniokwadratowa ciągła w przestrzeni $L_p^2(-1,1)$ dla p(x)=1, w bazie wielomianów Legendre'a. Całkowanie 2-punktową złożoną kwadraturą Gaussa-Legendre'a. Tablicowanie funkcji, przybliżenia i błędu w m punktach przedziału [-1,1] oraz obliczenie błędu średniokwadratowego w tych punktach.

2 Opis metody

Aproksymacja średniokwadratowa polega na jak najdokładniejszej (takiej aby błąd średniokwadratowy był jak najmniejszy) reprezentacji funkcji jako kombinacji liniowej w określonej bazie (w tym przypadku w bazie wielomianów Legendre'a).
Wielomiany Legendre'a to wielomiany ortagonalne, a więc macierz Grama jest macierzą diagonalną. Wielomiany $g_1, g_2 ... g_n$ to kolejne wielomiany Lagandre'a tworzące bazę. $g_{i+1}(x) = \frac{2i+1}{i+1} x g_n(x) - \frac{i}{i+1} g_{i-1}(x), \quad g_0(x) = 1, g_1(x) = x$

$$\begin{Vmatrix} < g_1, g_1 > & 0 & 0 & ... & 0 & 0 \\ 0 & < g_2, g_2 > & 0 & ... & 0 & 0 \\ 0 & 0 & < g_3, g_3 > & ... & 0 & 0 \\ . & . & . & . & . & . \\ . & . & . & . & . & . \\ . & . & . & . & . & . \\ 0 & 0 & 0 & ... & < g_{n-1}, g_{n-1} > & 0 \\ 0 & 0 & 0 & ... & 0 & < g_n, g_n > \end{Vmatrix} \times \begin{Vmatrix} a_1 \\ a_2 \\ a_3 \\ . \\ . \\ . \\ a_{n-1} \\ a_n \end{Vmatrix} = \begin{Vmatrix} < f_1, g_1 > \\ < f_2, g_2 > \\ < f_3, g_3 > \\ . \\ . \\ . \\ < f_{n-1}, g_{n-1} > \\ < f_n, g_n > \end{Vmatrix}$$

$f^* = a_1 g_1 + a_2 g_2 + a_3 g_3 + ... + a_{n_1} g_{n-1} + a_n g_n$

gdzie $a_i = \frac{<f_i, g_i>}{<g_i, g_i>}$.
Iloczyn wektorowy $< f_i, g_i > = \int p(x) f_i(x) g_i(x) dx$ obliczany jest przy pomocy złożonej dwupunktowej kwadratury Gaussa-Lagandre'a.
Błąd średniokwadratowy: $\sqrt{\frac{1}{m} \sum_{i=1}^{m} (f_i(x) - f_i^*(x))^2}$.

3 Warunki, założenia

Funkcja, ponieważ wykorzystuje złożoną 2-punktowa kwadraturę Gaussa-Lagandre'a, nie działa dla funckji, które nie są Lipschitzowskie na przedziale [-1,1].

4 Implementacja metody

Metoda została zaimplementowana w funkcji quadaproxgl(f,n,m), która wykorzystuje 2 funkcje pomocnicze: legpoly0(n) i calculateintegralfun(f,N). Funkcja znajduje aproksymację funkcji w bazie składającej się z n+1 kolejnych wielomianów Lagandre'a. Iloczyny sklarane obliczane są przy pomocy złożonych 2-punktowych kwadratur Gaussa-Lagandre'a.
Funkcja przyjmuje:
f - funkcję aproksymowaną
n - parametr określający wymiar bazy (wymiar wynosi n+1, od 1 do 10)
m - ilość przedziałów na których liczona jest kwadratura i w których liczony jest błąd średniokwadratowy
Funkcja zwraca:
f_m - aproksymację funkcji
wektor - tablicę wartości funkcji, przybliżenia i błędu.
err - błąd średniokwadratowy.

```
function [wektor,err,fm] = quadaproxgl(f,n,m)
f=sym(f);
syms  x a1 a2 a3 a4 a5 a6 a7 a8 a9 a10;
P=x;
for i=1:n+1
    P(i)=legpoly0(i-1); %wyznaczanie kolejnych
       wielomianow Legendre'a
end
b=[a1;a2;a3;a4;a5;a6;a7;a8;a9;a10];
Ln=0;
for i=1:n+1
    Ln = Ln + b(i) * P(i);
end
c=sym('c');
for i=1:n+1
    c(i)=f*P(i);
end
g=sym('g');
for i=1:n+1
    g(i)=P(i)*P(i);
end
ffun=matlabFunction(f);
fskalar(1)=calculateintegralfun(ffun,m);
for i=2:n+1
    ffun=matlabFunction(c(i));
    fskalar(i)=calculateintegralfun(ffun,m); %liczenie
        iloczynow skalarnych <f,g>
end
```

```
ffun=@(x)1+0.*x;
gskalar(1)=calculateintegralfun(ffun,m);
for i=2:n+1
    ffun=matlabFunction(g(i));
    gskalar(i)=calculateintegralfun(ffun,m); %liczenie
        iloczynow skalarnych <g,g>
end
for i=1:n+1
    a(i) = fskalar(i)/gskalar(i);
end
switch n %podstawianie wspolczynnikow i generowanie f*
    case 0
    fun = subs(Ln,{a1},{a(1)});
    case 1
    fun = subs(Ln,{a1,a2},{a(1),a(2)});
        (...)
    case 9
    fun = subs(Ln,{a1,a2,a3,a4,a5,a6,a7,a8,a9,a10},{a
       (1),a(2),a(3),a(4),a(5),a(6),a(7),a(8),a(9),a
       (10)});
end
f1=matlabFunction(f);
fm=matlabFunction(fun);
X= linspace(-1,1); %rysowanie wykresu
F1 = f1(X);
F2 = fm(X);
figure
plot(X,F2,'r*',X,F1,'b');
title(['Aproksymacja sredniokwadratowa ciagla dla n=',
    num2str(n),' i m=', num2str(m)])
xlabel('-1 <= x <= 1')
ylabel('wartosc')
legend({'f*(x)','f(x)'},'Location','southeast')
wektor=ones(m,3);
for i=1:m
    wektor(i,1) = f1(-1 + 2/m *i);
    wektor(i,2)= fm(-1 + 2/m *i);
    wektor(i,3)=abs(wektor(i,1)-wektor(i,2));
end
err  = sqrt(sum(wektor(:,3).*wektor(:,3)) /m);
end
```

Funkcja legpoly0(n) zwraca n-ty wielomian Lagandre'a.

```
function pval = legpoly0 ( n )
syms x;
if ( n < 0 )
  pval = 0;
elseif ( n == 0 )
  pval = 1;
elseif ( n == 1 )
  pval = x;
else
  p1 = 1;
  p2 = x;
  for i = 2 : n
    p0 = p1;
    p1 = p2;
    p2 = ( ( 2*i-1) * x * p1 - ( i - 1 ) * p0 ) / i;
  end
  pval = p2;
end
```

Iloczyny skalarne obliczane są za pomocą funkcji calculateintegralfun(f,N). Przyjmuje ona:
f - funkcję całkowaną
N - ilość przedziałów, na których liczona jest całka.

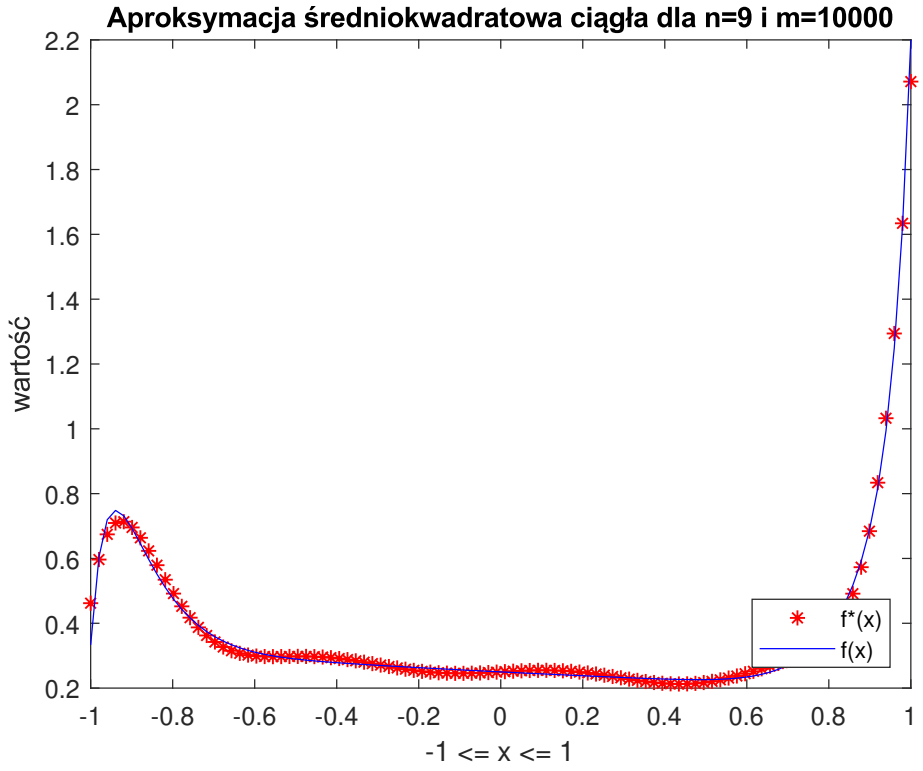
```
    function [intval] = calculateintegralfun(f,N)
H= 2/N;
intval =0;
for i=0:N-1
  intval =intval + f(-1+i*H + H*(1/2)*(1- 1/sqrt(3)))
                 + f(-1+i*H + H*(1/2)*(1+ 1/sqrt(3)));
end
 intval = intval * H/2;
end
```

W funkcji wykorzystałem, zmienne symboliczne, gdyż umożliwiają one wygodne zwracanie funkcji f^*, zamiast samego jej tablicowania. Ponadto maksymalny wymiar bazy wynosi 10, ale jak pokaże w przykładach, na tak małym obszarze nie stanowi to problemu.

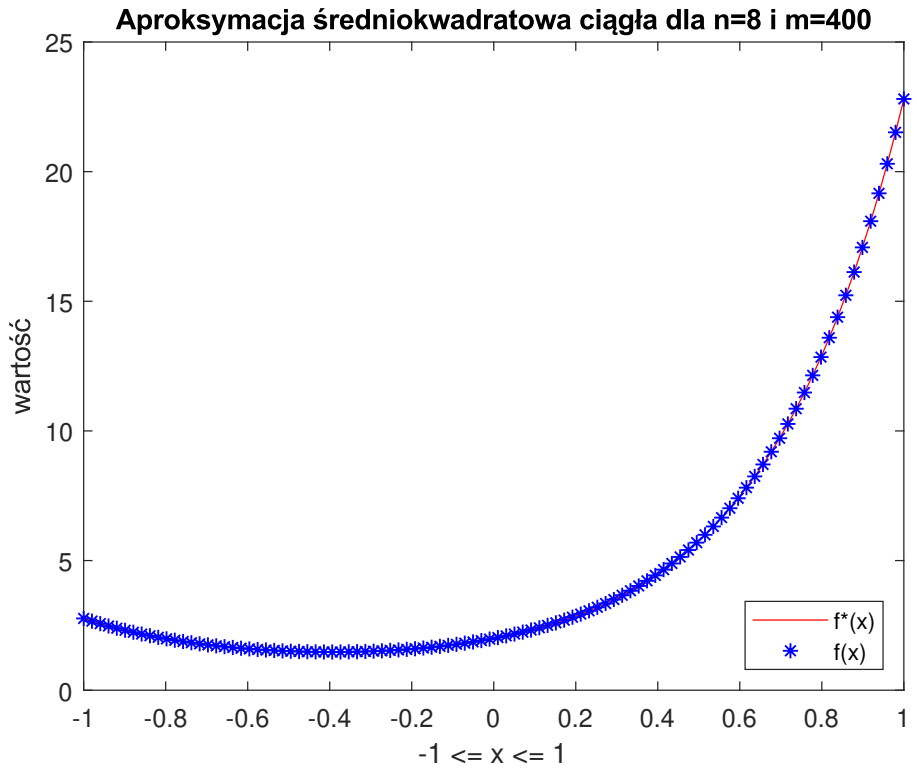
5 Przykłady i wnioski

1. Funkcja $f_1(x) = x^4 + e^x + sin(x) + 1 + 100x$**.**
Dla n=6, i m=50, err max =0.000586 błąd średniokwadratowy =0.00015703
Dla n=6, i m=100, err max =3.7577e-05 błąd średniokwadratowy =9.533e-06
Dla n=8, i m=1000, err max =8.9821e-08 błąd średniokwadratowy =1.4795e-08

2. Funkcja $f_2(x) = x^8 + x^{27} + 1/(x+4)$.
Dla n=6, i m=100, err max =0.48266 błąd średniokwadratowy =0.069021
Dla n=8, i m=200, err max =0.26443 błąd średniokwadratowy =0.036657
Dla n=9, i m=10000, err max =0.12864 błąd średniokwadratowy =0.017346



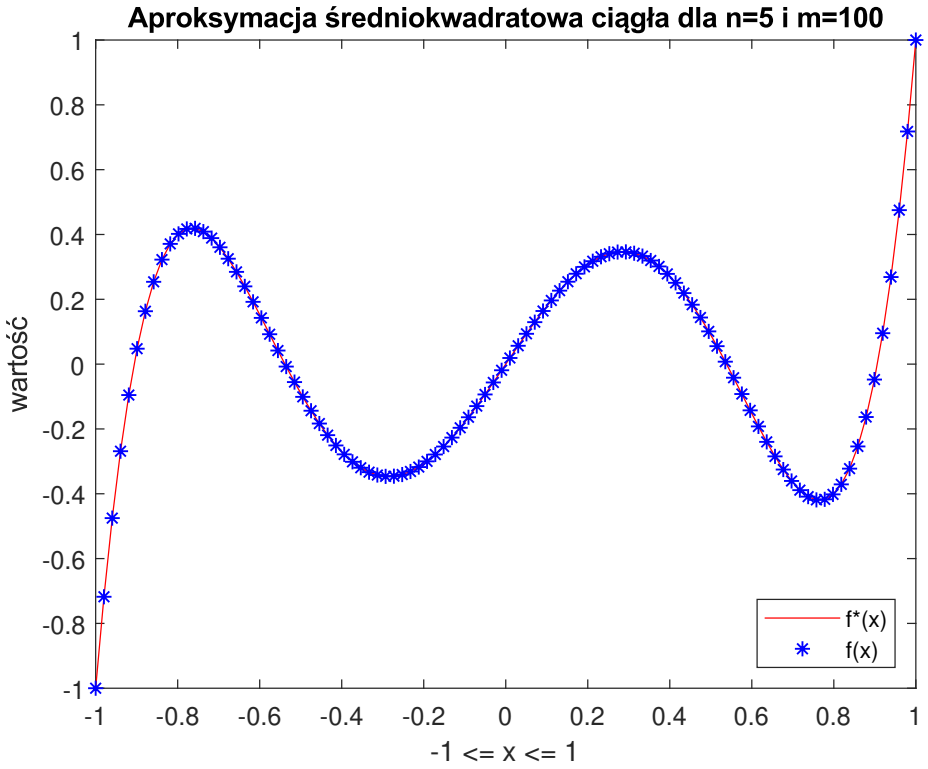
3. Funkcja $f_3(x) = e^{3x} + e^{x^2}$.
Dla n=6, i m=100, err max =0.027392 błąd średniokwadratowy =0.0058848
Dla n=7, i m=200, err max =0.0063546 błąd średniokwadratowy =0.001352
Dla n=8, i m=400, err max =0.00091299 błąd średniokwadratowy =0.00016938



4. Funkcja $f_4(x) = tg(x)$
Dla n=5, i m=100, err max =0.0066435 błąd średniokwadratowy =0.0015083
Dla n=5, i m=200, err max =0.0066427 błąd średniokwadratowy =0.0014858
Dla n=8, i m=200, err max =0.00096664 błąd średniokwadratowy =0.00019372

5. Funkcja $\frac{63x^5}{8} - \frac{(35x^3}{4} + \frac{15x}{8}$

Dla n=5, i m=100, err max =8.5908e-07 błąd średniokwadratowy =2.9805e-07
Dla n=5, i m=200, err max =5.371e-08 błąd średniokwadratowy =1.8613e-08
Dla n=5, i m=400000, err max =4.885e-15 błąd średniokwadratowy =2.1527e-15



Przykład tablicowania dla $f_3(x)$, n=7,m=200:

wartość f	wartość f^*	błąd
2.7160	2.7127	0.0033
2.6656	2.6636	0.0020
2.6168	2.6159	0.0009
2.5694	2.5694	0.0000
2.5236	2.5243	0.0007
2.4792	2.4804	0.0012
1.6768	1.6765	0.0004
1.6941	1.6936	0.0005
1.7122	1.7116	0.0006
1.7311	1.7304	0.0007
1.7509	1.7501	0.0008
1.7715	1.7706	0.0009
1.7930	1.7920	0.0010
1.8155	1.8144	0.0011
17.1276	17.1302	0.0025
17.6219	17.6244	0.0025
18.1311	18.1335	0.0024
18.6558	18.6579	0.0021
19.1964	19.1981	0.0017
19.7535	19.7546	0.0010
20.3276	20.3278	0.0002
20.9191	20.9181	0.0010

5.1 Wnioski i zakończenie

Metoda działa dla tych funkcji, dla których działa kwadratura Gaussa-Legendre'a.

Błąd średniokwadratowy jest mniejszy gdy dla tego samego n weźmiemy większy m (kwadratura jest dokładniejsza, bo liczymy ją na większej ilości przedziałów).

Dla tego samego m błąd jest mniejszy dla większego n (dodatkowe wielomiany pozwalają dokładniej przedstawić funkcję).