Sprawozdanie
Metody Numeryczne 2
Temat 2

Przemysław Woźniakowski

2018-11-02

1 Treść zadania

Interpolacja funkcjami bikwadratowymi na obszarze $D = D_1 - D_2$, gdzie $D_1 = [-2, 2] \times [-2, 2]$, $D_2 = [-1, 1] \times [-1, 1]$, podzielonymi na $12n^2$ kwadratów przystających. Tablicowanie funkcji, przybliżenia i błędu w środkach kwadratów powstałych po podzieleniu kwadratów o boku $\frac{1}{n}$ na pół w pionie i poziomie. Obliczenie błędu maksymalnego w tych punktach.

2 Opis metody

Metoda polega na interpolacji funkcji, przy pomocy wielomianu dwóch zmiennych z przestrzeni:

$$P_4 :\overset{\text{def.}}{=} \{p : p(x,y) = \sum_{0 \leqslant \mu+\kappa \leqslant 4} a_{\mu\kappa} x^\mu y^\kappa, \quad \mu, \kappa \leqslant 2\}$$

Jego postać to:

$$p(x,y) = a_0 + a_1 x + a_2 y + a_3 xy + a_4 x^2 + a_5 y^2 + a_6 x^2 y + a_7 xy^2 + a_8 x^2 y^2$$

współczynniki $a_0, a_1, ..., a_8$ wyznaczane są poprzez rozwiązanie układu równań z 9 niewiadomymi, stworzonego na podstawie 9 węzłów interpolacji, którymi są wierzchołki, środki boków i środek kwadratu. Układowi odpowiada macierz $((x_1, y_1), (x_2, y_2), ..., (x_9, y_9)$ - kolejne punkty interpolacji, f(x,y) - funkcja interpolowana):

$$\left[\begin{array}{ccccccccc|c}
a_0 & a_1x_1 & a_2y_1 & a_3x_1y_1 & a_4x_1^2 & a_5y_1^2 & a_6x_1^2y_1 & a_7x_1y_1^2 & a_8x_1^2y_1^2 & f(x_1,y_2) \\
a_0 & a_1x_2 & a_2y_2 & a_3x_2y_2 & a_4x_2^2 & a_5y_2^2 & a_6x_2^2y_2 & a_7x_2y_2^2 & a_8x_2^2y_2^2 & f(x_2,y_2) \\
. & . & . & . & . & . & . & . & . & . \\
. & . & . & . & . & . & . & . & . & . \\
. & . & . & . & . & . & . & . & . & . \\
a_0 & a_1x_9 & a_2y_9 & a_3x_9y_9 & a_4x_9^2 & a_5y_9^2 & a_6x_9^2y_1 & a_7x_9y_9^2 & a_8x_9^2y_9^2 & f(x_9,y_9)
\end{array}\right]$$

3 Warunki, założenia

Funkcja interpolująca działa dobrze dla funkcji interpolowanych, które mają na obszarze D ograniczoną pochodną. Funkcję takie jak np. tangens(x+y) na niektórych odcinkach rosną bardzo szybko, co połączone z faktem, że punkty interpolacji są rozłożone równomiernie powoduje powstawanie bardzo dużego błędu maksymalnego.

4 Implementacja metody

Metoda została zaimplementowana w funkcji interpolbikwad, która posługuje się funkcją pomocniczą ibikwad, wywołując ją dla każdego z $12n^2$ kwadratów. Funkcja interpolbikwad przyjmuje jako parametr interpolowaną funkcję. Zwraca zaś macierz $48n^2 \times 5$ w której wierszach znajdują się informacje na temat badanych punktów w formacie: współrzędna x punktu, współrzędna y, wartość funkcji, przybliżenie, błąd (wartość bezwzględna z różnicy wartości funkcji i przybliżenia). Zwraca także wektor, w którym zawarte są informacje na temat punktu z największym błędem.

Implementacja funkcji interpolbikwad:

```
function [A,maxerr] =interpolbikwad(fun,n)
A=ones(n^2*4,5);
count=1;
for i=0:(n-1)
    for j=0:(n-1)
                xwekt=[-2+(1/n)*i,-1+(1/n)*i, (1/n)*i,
                   1+(1/n)*i,1+(1/n)*i,1+(1/n)*i,
                   1+(1/n)*i,(1/n)*i,-1+(1/n)*i,
                  -2+(1/n)*i,-2+(1/n)*i,-2+(1/n)*i];
                ywekt=[1+(1/n)*j,1+(1/n)*j,1+(1/n)*j,
                   1+(1/n)*j,(1/n)*j,-1+(1/n)*j,
                  -2+(1/n)*j,-2+(1/n)*j,-2+(1/n)*j,
                  -2+(1/n)*j,-1+(1/n)*j,(1/n)*j];
       for k=1:12
        A(((count-1)*4 +1):count*4,:)=ibikwad(fun,
           xwekt(k),ywekt(k), 1/n);
        count=count+1;
       end
    end
end
[unused,ind] = max(A(:,5));
maxerr=A(ind,:);
end
```

Funkcja pomocnicza ibikwad:

Paramatetry:
fun - funkcja interpolowana
x_0 - współrzędna x lewego dolnego wierzchołka kwadratu
y_0 - współrzędna y lewego dolnego wierzchołka kwadratu
l - długość boku kwadratu

Funkcja zwraca:
wynik - macierz 4×5, w której znajdują się informacje na temat 4 punktów znajdujących się w środkach 4 mniejszych kwadratów.
Implementacja funkcji ibikwad:

```
function [wynik] = ibikwad(fun,x0,y0,l)
A=ones(9,9);
b=ones(9,1);
x=[x0 x0 x0+l x0+l x0 x0+l/2 x0+l x0+l/2 x0+l/2];
 %wspolrzedne x 9 wezlow interpolacji
y=[y0 y0+l y0+l y0 y0+l/2 y0+l y0+l/2 y0 y0+l/2];
%wspolrzedne y 9 wezlow
for i=1:9
        A(i,:)=[1 x(i) y(i) x(i)*y(i)   x(i)^2 y(i)^2
           x(i)^2*y(i) y(i)^2*x(i) x(i)^2*y(i)^2];
        b(i)=fun(x(i),y(i));
end
px=[x0+l/4;x0+l/4; x0+l*3/4; x0+3*l/4]; %wspolrzedne x
    punktow, ktorych wartosc przyblizamy
py=[y0+l/4;y0+l*3/4; y0+l*3/4; y0+l/4]; %wspolrzedne y
    punktow, ktorych wartosc przyblizamy
a=A\b; %obliczanie wspolczynnikow funkcji
   bikwadratowej
wynik(:,1)=px;
wynik(:,2)=py;
wynik(:,4) =a(1) + a(2).*px + a(3).*py + a(4).*px.*py
   + a(5).*px.^2 + a(6)*py.^2
+ a(7).*px.^2.*py + a(8)*py.^2.*px+a(9).*px.^2.*py.^2;
wynik(:,3) = fun(px,py);
wynik(:,5) = abs(wynik(:,4)-wynik(:,3));
end
```

5 Przykłady i wnioski

Pierwszą funkcją, która została przebadana jest $f(x,y) = cos(x+y) + xy^2$):

Maksymalny błąd dla n=5: 0.00012486
Maksymalny błąd dla n=10: $1.5623 * 10^{-5}$
Maksymalny błąd dla n=20: $1.953 * 10^{-6}$
Maksymalny błąd dla n=40: $2.4414 * 10^{-7}$

Warto zauważyć, że:
Przy wzroście n z 5 do 10 błąd maleje: 7.9923 razy.
Przy wzroście n z 10 do 20 błąd maleje: 7.9993 razy.

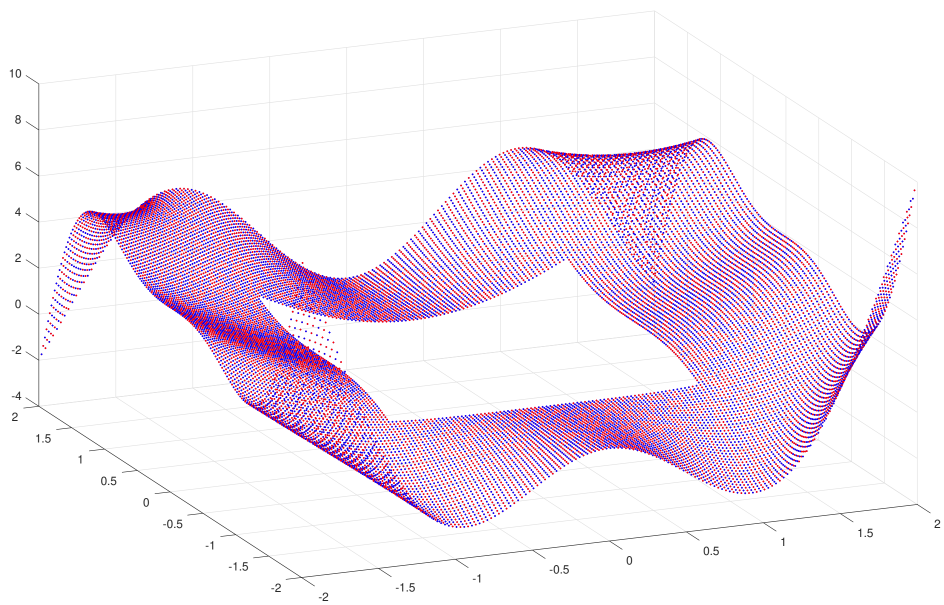
Przy wzroście n z 20 do 40 błąd maleje: 7.9997 razy.
Oznacza, że przy dwu-krotnym wzroście n błąd maleje około 2^3 razy.

Dla funkcji $f(x, y) = sin(xy)xy^2 + x^2$:

Maksymalny błąd dla n=5: 0.0087967
Maksymalny błąd dla n=10: 0.0012505
Maksymalny błąd dla n=20: 0.00016493
Maksymalny błąd dla n=40: $2.1121 * 10^{-5}$
Maksymalny błąd dla n=80: $2.6706 * 10^{-6}$

Przy wzroście n z 5 do 10 błąd maleje: 7.0345 razy.
Przy wzroście n z 10 do 20 błąd maleje: 7.5822 razy.
Przy wzroście n z 20 do 40 błąd maleje: 7.8086 razy.
Przy wzroście n z 40 do 80 błąd maleje: 7.9088 razy.

Jak widać dla tej funkcji iloraz maksymalnego błędu dla danego n i maksymalnego błedu dla 2n również zbiega do 2^3 wraz ze wzrostem n.
Dla przykładu, tak wygląda wykres z zaznaczonym wartościami w węzłach interpolacji i w punktach przybliżanych.



Wykres dla funkcji $f(x, y) = sin(xy)xy^2 + x^2$ i n=20
czerwone punkty to wartości funkcji w węzłach, niebieskie wartości przybliżone

Dla funkcji bikwadratowych i małych n-ów wynik jest oczywisty:
$f(x,y) = 3 + x + y + xy^2 + 2x^2y^2 + 3y^2$:

Maksymalny błąd dla n=1: 0
Maksymalny błąd dla n=3: 0
Maksymalny błąd dla n=5: $7.1054 * 10^{-15}$
Maksymalny błąd dla n=40: $2.13161 * 10^{-14}$

Dość nieoczekiwanie jednak, błąd dla małych n-ów jest zerowy, a dla większych nie. Wynika to z faktu, że dla większych n-ów, współrzędne punktów to liczby z wieloma cyframi znaczącymi po przecinku, przez co przy wyznaczaniu współczynników a_i pojawiają się błędy, które skutkują błędami wyniku. Są to jednak odchylenia rzędu 10^{-14} więc nie stanowią problemu.

Sprawdźmy co zajdzie dla funkcji, która na obszarze D nie ma ograniczonej pochodnej: $f(x,y) = tg(x+y)$:

Maksymalny błąd dla n=5: 20.0244
Maksymalny błąd dla n=10: 32.8557
Maksymalny błąd dla n=20: 117.9269
Maksymalny błąd dla n=40: 146.7918
Maksymalny błąd dla n=80: 291.3566

Jak widać, dla funkcji nie spełniających założeń błędy faktycznie są znaczne. Funkcje, w których x i y występują w potęgach wyższych niż 2 (nie bikwadratowe) są przybliżane dosyć dobrze (zwłaszcza dla większych n-ów): $f(x,y) = x^5y^3 + y^4$:

Maksymalny błąd dla n=5: 0.11518
Maksymalny błąd dla n=10: 0.015596
Maksymalny błąd dla n=20: 0.0020281
Maksymalny błąd dla n=40: 0.00025855
Maksymalny błąd dla n=80: $3.2638 * 10^{-5}$

Na koniec krótki przykład tablicowania $f(x,y) = cos(x+y)+xy^2$ dla $n = 40$:

x	y	f(x,y)	przybliżenie	błąd
-1.006250	-1.006250	-1.446347678981863	-1.446347899197492	0.000000220215629
-1.006250	-1.018750	-1.483084999982018	-1.483084999479925	0.000000000502093
-1.018750	-0.018750	0.508016529320688	0.508016738973071	0.000000209652382
-1.018750	-0.006250	0.519059091911495	0.519059091317449	0.000000000594046
-1.006250	-0.006250	0.529702675263588	0.529702467591359	0.000000207672228
-1.006250	-0.018750	0.518745127067745	0.518745126473699	0.000000000594046
-1.018750	0.981250	0.018391928096724	0.018391936106068	0.000000008009344
-1.018750	0.993750	-0.006367903646172	-0.006367904790194	0.000000001144022
-1.006250	0.993750	0.006210694376622	0.006210690180684	0.000000004195939
-1.006250	0.981250	0.030818131510078	0.030818130366056	0.000000001144022

5.1 Podsumowanie

Podsumowując, funkcja dla funkcji spełniających założenia działa zadowalająco. Dla funkcji bikwadratowych działa dobrze dla małych n-ów, natomiast przy większych pojawiają się drobne błędy wynikające z rozwiązywania układu równań. Funkcja, faktycznie nie działa dla funkcji o nieograniczonej pochodnej na obszarze D.