

Metody numeryczne projekt nr 1

Dominika Gimzicka

nr albumu:

grupa wtorek 12

27.11.2023

1 Treść zadania

Metoda Simposna obliczania przybliżonej wartości całki $\int_a^b f(t)dt$, gdzie

$$f(t) = \sum_{k=1}^n a_k \sin kt$$

dla obliczonej wartości funkcji zastosować metodę Goertzela.

2 Opis metody

2.1 Metoda Goertzela

Przyjmijmy, że dany jest następujący wielomian:

$$\omega(\lambda) = \sum_{n=0}^N a_n \lambda^n, \text{ gdzie } a_n \in \mathbb{C}, n=0,1,\dots,N \text{ oraz punkt } z=x+iy \in \mathbb{C}$$

Za pomocą algorytmu Goertzela obliczymy właśnie $w(z)$. Najpierw wielomian $\omega(\lambda) = \sum_{n=0}^N a_n \lambda^n$ dzielimy przez wielomian $(-z)(\lambda - \bar{z}) = \lambda^2 - \hat{p}\lambda - \hat{q}$. Współczynnikami tego wielomianu są $\hat{p}=2x$ i $\hat{q} = -|z|^2$. Zatem możemy zapisać: $\omega(\lambda) = (-z)(\lambda - \bar{z}) \sum_{n=2}^N b_n \lambda^{n-2} + b_0 + b_1 \lambda$

Widzimy więc, że $\omega(z)=b_0 + b_1 z$, co oznacza, że wartość wielomianu ω w punkcie z jest równa wartości reszty z dzielenia tego wielomianu przez trójmian $\lambda^2 - \hat{p}\lambda - \hat{q}$. Algorytm Goertzela polega na wyznaczeniu wartości współczynników $b_N, b_{N-1}, \dots, b_1, b_0$, a następnie policzeniu wartości tej reszty.

Metoda Goertzela, znana także jako Algorytm Goertzela, bardzo dobrze nadaje się do liczenia wartości funkcji typu $p(t) = \sum_{n=0}^N a_n \cos kt$ oraz $h(t) = \sum_{n=0}^N a_n \sin kt$. Zauważmy, że gdy podstawimy $z=\cos(t)+isin(t)$, to $p(t)=\text{Re}\omega(z)$, a $h(t)=\text{Im}\omega(z)$.

W moim zadaniu będę liczyć $\sum_{k=1}^n a_k \sin kt$, stąd pomnę w dalszej implementacji kodu tej metody współczynnik a_0 .

2.2 Metoda Simpsona

Główną ideą metody Simpsona jest przybliżanie funkcji podcałkowej na danym przedziale parabolą - będziemy obliczać sumy wycinków obszarów pod tą parabolą. Znając wartości y_0, y_1, y_2 funkcji $f(x)$ w 3 punktach $a=x_0, x_1, x_3 = b$ (przy czym $x_2 - x_1 = x_1 - x_0 = h$) przybliżamy funkcję wielomianem Lagrange'a i całkując w przedziale $[a,b]$ otrzymujemy przybliżoną wartość całki:

$$\int_a^b f(x)dx \approx \frac{h}{3}(y_0 + 4y_1 + y_2)$$

Jednak na potrzeby metod numerycznych będziemy korzystać ze złożonej metody Simpsona, którą się wyprowadza ze wzoru powyżej (przeprowadzając wiele przekształceń różnego typu, które nie są istotą tego zadania, więc je pominiemy). Ma ona postać wzoru:

$$\int_a^b f(x)dx \approx \frac{h}{6}[f(a) + f(b) + 2 \sum_{i=1}^{m-1} f(t_i) + 4 \sum_{i=1}^m f(t_{i-1} + \frac{h}{2})]$$

Z tego wzoru będę korzystać dalej obliczając wartość całki w moim algorytmie.

3 Opis programu obliczeniowego

Mój program obliczeniowy składa się z dwóch funkcji (jednej do wyliczania wartości wielomianu metodą Goertzela, a drugiej do przybliżania wartości całki metodą Simpsona).

3.1 metodaGoertzela

Jest to funkcja służąca do obliczania wartości funkcji $f(t) = \sum_{k=1}^n a_k \sin(kt)$ metodą Goertzela. Funkcja metodaGoertzela przyjmuje następujące argumenty:

- współczynniki - lista współczynników wielomianu, gdzie elementami są po kolei a_1, a_2, \dots, a_n
- t - argument funkcji f

Funkcja zwraca wektor *wynik* będący wartością funkcji f w punkcie t. Poniżej zamieszczony jest algorytm napisany przeze mnie w języku MATLAB:

```
1 function [wynik] = metodaGoertzela(wspolczynniki, t)
2     n = length(wspolczynniki);
3     x = cos(t);
4     y = sin(t);
5     z = complex(x, y); % z = cos(t) + isin(t)
6
7 % poniżej rozpatrzone są "szczególne" przypadki współczynników:
8 % 1. lista współczynników jest pusta
9 if n==0
10     error("Lista współczynników wielomianu jest pusta")
11 end
12 % 2. lista współczynników zawiera tylko 1 element
13 if n==1
14     wynik = wspolczynniki(1)*y; % wynik=a1*sin(t)
15 end
16
17 %poniżej implementacja metody, gdy lista współczynników jest dłuższa niż 1
18 p = 2*x;
19 q = -(x^2 + y^2);
20 b = zeros(n+1, 1);
21 b(n) = wspolczynniki(n);
22 for i = 1 : n-1
23     j = n-i;
24     b(j) = wspolczynniki(j) + p*b(j+1) + q*b(j+2);
25 end
26 wynik = y * b(1); % wynik=sin(t)*b1
27
28 end
```

3.2 metodaSimpsona

Jest to funkcja służąca do przybliżania wartości całki $\int_a^b f(t)dt$ metodą Simpsona. Przyjmuje ona następujące argumenty:

- współczynniki - lista współczynników wielomianu, gdzie elementami są po kolei a_1, a_2, \dots, a_n
- a - dolna granica całkowania
- b - górna granica całkowania
- n - liczba przedziałów na jakie dzielimy przedział [a,b]

Funkcja zwraca wektor *wynik* będący przybliżoną wartością całki. Poniżej zamieszczony jest algorytm napisany przeze mnie w języku MATLAB:

```

1 function [wynik] = metodaSimpsona(wspolczynniki, a, b, n)
2 % szczególne przypadki współczynników są "rozwiązywane" w funkcji metodaGoertzela
3
4 % h - odległość między dwoma sąsiednimi punktami podziałowymi
5 h = (b-a)/n;
6
7 % t - wektor punktów podziałowych
8 t = zeros(n, 1);
9 for i =1:n
10     t(i) = a + i*h;
11 end
12
13 % Obliczamy:
14 % sumę wartości funkcji f w punktach podziałowych (s_p)
15 % sumę wartości funkcji f w punktach środkowych każdego przedziału (s_s)
16 s_p = 0;
17 s_s = 0;
18 for iterator = 1:n
19     s_s = s_s + metodaGoertzela(wspolczynniki,t(iterator) - h/2);
20     if iterator<n
21         s_p = s_p + metodaGoertzela(wspolczynniki, t(iterator));
22     end
23 end
24
25 %wyznaczamy wartość całki według podanego wyżej wzoru
26 wynik = (h/6) * (metodaGoertzela(wspolczynniki, a) + metodaGoertzela(wspolczynniki,b) + 2*s_p + 4*s_s);
27 end

```

4 Przykłady obliczeniowe

Poniżej przedstawię najpierw parę ciekawych przykładów, a następnie pokażę zachodzące błędy przy użyciu kliku metryk w formie tabeli.

Zacznijmy od przykładów

(podane dokładne wyniki całek zostały obliczone za pomocą strony <https://www.wolframalpha.com/>)

4.1 pusty wektor współczynników

Na początku pokażę parę przykładów ze "szczególnymi" przypadkami listy współczynników.

```

>> metodaSimpsona([],0,20,10000)
Error using metodaGoertzela
Lista współczynników wielomianu jest pusta

```

Jak widzimy przy podaniu pustego wektora współczynników program zachowuje się prawidłowo i wyrzuca błąd.

4.2 jednoelementowy wektor współczynników

```

>> metodaSimpsona([1],0,20,10000)

ans =

    0.591917938186611

```

W argumencie funkcji podaliśmy wektor współczynników [1] (zatem $a_1=1$), przedział całkowania zaczynający się od 0, a kończący na 20 oraz 10000 punktów podziałowych. Stąd:

$$f(t) = \sum_{k=1}^1 a_k \sin kt = \sin(t)$$

$$\int_0^{20} \sin(t) dt \approx 0.591917938186608$$

Widać tu że błąd przy obliczeniach wielomianów stopnia pierwszego jest niemalże zerowy, a tak niewielka różnica mogła powstać nawet przez specyfikację programu, którego używam do liczenia dokładnych wartości całki.

4.3 dwuelementowy wektor współczynników

```
>> metodaSimpsona([1,2],0,20,10000)

ans =

    2.258855999839019
```

Wartości argumentów funkcji są takie same jak powyżej, za wyjątkiem wektora współczynników, który teraz jest dwuelementowy (więc $a_1 = 1, a_2 = 2$)
Stąd:

$$f(t) = \sum_{k=1}^2 a_k \sin kt = \sin(t) + 2\sin(2t)$$

$$\int_0^{20} (\sin(t) + 2\sin(2t)) dt \approx 2.258855999838869$$

Widać że dla 10000 punktów podziałowych błąd jest niewielki.

```
>> metodaSimpsona([1,2],0,20,1000000)

ans =

    2.258855999838902
```

Zato przy 1000000 widać, że wynik jest lekko dokładniejszy.

```
>> metodaSimpsona([1,2],0,20,100)

ans =

    2.258871217157064
```

Tutaj, przy 100 punktach podziałowych, widać, że błąd jest już bardziej zauważalny.

4.4 trzyelementowy wektor współczynników

```
>> metodaSimpsona([1,2,3],0,20,10000)

ans =

    4.211268980255068
```

Wartości argumentów funkcji są znowu takie same jak powyżej, za wyjątkiem wektora współczynników, który teraz jest trzyelementowy (więc $a_1 = 1, a_2 = 2, a_3 = 3$)
Stąd:

$$f(t) = \sum_{k=1}^3 a_k \sin kt = \sin(t) + 2\sin(2t) + 3\sin(3t)$$
$$\int_0^{20} (\sin(t) + 2\sin(2t) + 3\sin(3t)) dt \approx 4.211268980254026$$

Z tego wynika, że przy wielomianach 3 stopnia i 1000 punktach podziałowych nasz błąd jest nadal pomijalny.

```
>> metodaSimpsona([1,2,3],0,20,1000000)

ans =

    4.211268980253913
```

Przy milionie punktów podziałowych znowu widać, że wynik jest jeszcze dokładniejszy.

```
>> metodaSimpsona([1,2,3],0,20,100)

ans =
```

```
    4.211373006476583
```

Przy stu punktach podziałowych występuje już niepomijalny błąd.

4.5 pięcioelementowy wektor współczynników

W kolejnych przykładach sprawdzimy co się dzieje z wielomianami wyższych stopni, w tym przypadku stopnia 5.

```
>> metodaSimpsona([1,2,3,4,5],0,20,10000)
```

```
ans =
```

```
5.459337351808466
```

Wiemy, że:

$$f(t) = \sum_{k=1}^5 a_k \sin kt = \sin(t) + 2\sin(2t) + 3\sin(3t) + 4\sin(4t) + 5\sin(5t)$$
$$\int_0^{20} (\sin(t) + 2\sin(2t) + 3\sin(3t) + 4\sin(4t) + 5\sin(5t))dt \approx 5.459337351805389$$

Możemy zauważyć, że dla takiego wielomianu błąd widocznie się zwiększył. Sprawdźmy jak to wygląda dla mniejszej liczby punktów podziałowych.

```
>> metodaSimpsona([1,2,3,4,5],0,20,100)
```

```
ans =
```

```
5.459651626248299
```

Widzimy, że wynik się już różni na czwartym miejscu po przecinku, więc przy wielomianach takiego stopnia należy wybrać odpowiednio dużą liczbę punktów podziałowych.

Zobaczmy jak ten błąd zmieni się przy jeszcze większym stopniu wielomianu.

4.6 dziesięcioelementowy wektor współczynników

```
>> metodaSimpsona([1,2,3,4,5,6,7,8,9,10],0,20,10000)
```

```
ans =
```

```
10.929871662281332
```

Zatem:

$$f(t) = \sum_{k=1}^{10} a_k \sin kt = \sin(t) + 2\sin(2t) + 3\sin(3t) + \dots + 10\sin(10t)$$
$$\int_0^{20} (\sin(t) + 2\sin(2t) + 3\sin(3t) + \dots + 10\sin(10t))dt \approx 10.92987166212918$$

Można zauważyć, że błąd przy takiej samej liczbie punktów podziałowych, ale większym stopniu wielomianu staje się bardziej zauważalny.

Sprawdźmy ponownie co się stanie, gdy zmniejszymy liczbę punktów podziałowych do stu.

```
>> metodaSimpsona([1,2,3,4,5,6,7,8,9,10],0,20,100)

ans =

    10.946552587471031
```

Widać, że wynik dokładny oraz przybliżony metodą Simpsona nie są już do siebie podobne, a różnica w wyniku widoczna jest od drugiego miejsca po przecinku. Zatem potwierdza to wniosek z poprzedniego podpunktu, że dla wielomianów większych stopni należy wybrać w metodzie Simpsona większą ilość punktów podziałowych.

4.7 Porównanie błędów

Poniżej za pomocą tabeli porównane są błędy, z paru wybranych przykładów podanych powyżej, za pomocą różnych metryk. Uwzględnij tam:

- błąd bezwzględny (czyli $E_{abs} = |Wynik_{dokladny} - Wynik_{przybliony}|$)
- błąd względny (czyli $E_{rel} = \left| \frac{Wynik_{dokladny} - Wynik_{przybliony}}{Wynik_{dokladny}} \right|$)

Rozpatrzmy dwie wartości liczby punktów podziałowych:

całka	wartość przybliżona	wartość dokładna	E_{abs}	E_{rel}
$\int_0^{20} (\sin(t) + 2\sin(2t))dt$	2.258855999839019	2.258855999838869	$1.5 \cdot 10^{-13}$	$\approx 6.64 \cdot 10^{-14}$
$\int_0^{20} (\sin(t) + 2\sin(2t) + 3\sin(3t))dt$	4.211268980255068	4.211268980254026	$1.042 \cdot 10^{-12}$	$\approx 2.47 \cdot 10^{-13}$
$\int_0^{20} (\sin(t) + \dots + 5\sin(5t))dt$	5.459337351808466	5.459337351805389	$3.077 \cdot 10^{-12}$	$\approx 5.64 \cdot 10^{-13}$
$\int_0^{20} (\sin(t) + \dots + 10\sin(10t))dt$	10.929871662281332	10.92987166212918	$1.52152 \cdot 10^{-10}$	$\approx 1.39 \cdot 10^{-11}$

Tabela 1: liczba punktów podziałowych = 10000

całka	wartość przybliżona	wartość dokładna	E_{abs}	E_{rel}
$\int_0^{20} (\sin(t) + 2\sin(2t))dt$	2.258871217157064	2.258855999838869	0.000015	$\approx 6.74 \cdot 10^{-6}$
$\int_0^{20} (\sin(t) + 2\sin(2t) + 3\sin(3t))dt$	4.211373006476583	4.211268980254026	0.000104	$\approx 2.47 \cdot 10^{-5}$
$\int_0^{20} (\sin(t) + \dots + 5\sin(5t))dt$	5.459651626248299	5.459337351805389	0.00031	$\approx 5.8 \cdot 10^{-5}$
$\int_0^{20} (\sin(t) + \dots + 10\sin(10t))dt$	10.946552587471031	10.92987166212918	0.016680925341851	≈ 0.018

Tabela 2: liczba punktów podziałowych = 100

Na podstawie tabel widzimy, że błędy bezwzględne są zazwyczaj bardzo małe, co wskazuje na to, że różnice między wartościami przybliżonymi a dokładnymi są niewielkie. Błędy względne są również na niskim poziomie, co oznacza, że metoda numeryczna jest skuteczna w osiąganiu precyzyjnych wyników, nawet dla bardziej złożonych funkcji.

Jednak należy zwrócić uwagę na to, że w miarę zmniejszania liczby punktów podziałowych (w tym przypadku do 100), wartości przybliżone coraz bardziej oddalają się od wartości dokładnych. To potwierdza, że większa liczba punktów podziałowych zwiększa dokładność numerycznej aproksymacji całki.

5 Wnioski

Metoda Simpsona (z funkcją wyliczania wartości wielomianu algorytmem Goertzela) okazała się skuteczna do liczenia całki $\int_a^b f(t)dt$ (gdzie $f(t) = \sum_{k=1}^n a_k \sin kt$) jeśli tylko użyjemy wystarczająco dużej liczby punktów podziałowych. Jak można było zauważyć - przy większej liczbie punktów podziałowych błędy są generalnie mniejsze, co świadczy o tym, że zwiększanie liczby punktów podziałowych prowadzi do uzyskania dokładniejszych wyników.

Należy jednak zaznaczyć, że zastosowanie tej metody w MATLABIE pozwoliło na szybkie (nawet dla wielomianów wysokiego stopnia oraz dużej ilości punktów podziałowych) przybliżenie wartości danej całki.

6 Źródła

- Notatki do Metod Numerycznych autorstwa dr inż. Iwony Wróbel oraz dr inż. Magdaleny Jasionowskiej-Skop
- <http://fluid.itcmp.pwr.wroc.pl/~znmp/dydaktyka/metnum/simpson.pdf>
- https://pl.wikipedia.org/wiki/Metoda_Simpsona
- https://eduinformatyka.waw.pl/inf/alg/004_int/0004.php