

Tema 10

Traistaru Alexandru Mihai

Grupa 324AA

```
tema10.m x +
1 % Definim funcția f(x)
2 f = @(x) x.^2 + 2.*x - x./3;
3
4 % Limitele de integrare
5 a = 2; % Limita inferioară
6 b = 3; % Limita superioară
7
8 % Numărul de subintervale (cu cât e mai mare, cu atât rezultatul e mai precis)
9 n = 1000;
10
11 % Lungimea fiecărui subinterval
12 dx = (b - a) / n;
13
14 % Vectorul de puncte din stânga fiecărui subinterval
15 x = linspace(a, b-dx, n);
16
17 % Aplicăm metoda dreptunghiurilor
18 integral = sum(f(x) * dx);
19
20 % Afișăm rezultatul
21 fprintf('Valoarea aproximativă a integralei definite este: %.6f\n', integral);
22
```

Command Window

```
>> tema10
Valoarea aproximativă a integralei definite este: 10.496667
fx >>
```



rezolva integrala de la a la b din $f(x) dx$ cu ajutorul metodei dreptunghiurilor in matlab. $f(x) = x^2 + 2x - x/3$, $a=2$, $b=3$. metoda dreptunghiurilor trebuie implementata explicit nu apelata ca functie de biblioteca



Iată codul MATLAB care implementează metoda dreptunghiurilor explicit pentru funcția $f(x) = x^2 + 2x - \frac{x}{3}$, cu limitele $a = 2$ și $b = 3$:

matlab

Copy code

```
% Definim funcția f(x)
f = @(x) x.^2 + 2.*x - x./3;

% Limitele de integrare
a = 2; % Limita inferioară
b = 3; % Limita superioară

% Numărul de subintervale (cu cât e mai mare, cu atât rezultatul e mai precis)
n = 1000;

% Lungimea fiecărui subinterval
dx = (b - a) / n;

% Vectorul de puncte din stânga fiecărui subinterval
x = linspace(a, b-dx, n);

% Aplicăm metoda dreptunghiurilor
integral = sum(f(x) * dx);

% Afișăm rezultatul
fprintf('Valoarea aproximativă a integralei definite este: %.6f\n', integral);
```

Explicații:

1. Funcția $f(x)$: Codul definește funcția $f(x) = x^2 + 2x - \frac{x}{3}$ folosind funcțiile anonime MATLAB.
2. Intervalul $[a, b]$: Limitele sunt setate la $a = 2$ și $b = 3$.

Explicații:

1. Funcția $f(x)$: Codul definește funcția $f(x) = x^2 + 2x - \frac{x}{3}$ folosind funcțiile anonime MATLAB.
2. Intervalul $[a, b]$: Limitele sunt setate la $a = 2$ și $b = 3$.
3. Metoda dreptunghiurilor:
 - Intervalul este împărțit în n subintervale de lungime egală (dx).
 - Punctele de evaluare sunt cele din stânga fiecărui subinterval.
 - Suma ariilor dreptunghiurilor este calculată prin înmulțirea valorii funcției în punctele selectate cu lungimea subintervalului.
4. Rezultatul: Se calculează și afișează valoarea aproximativă a integralei definite.

Cum să rulați codul:

1. Copiați codul într-un fișier MATLAB (de exemplu, `dreptunghiuri.m`).
2. Rulați scriptul în MATLAB.
3. Codul va afișa rezultatul aproximativ al integralei pe ecran.

Dacă doriți să modificăm metoda (de exemplu, să folosim punctul de mijloc), vă pot oferi și acea variantă!



ai vreo sugestie de imbunatatire?



Da, metoda dreptunghiurilor poate fi îmbunătățită pentru a obține rezultate mai precise și mai eficiente. Iată câteva sugestii:

1. Utilizarea punctului de mijloc (Metoda dreptunghiurilor îmbunătățită):

În loc să evaluăm funcția în extremitatea stângă sau dreaptă a fiecărui subinterval, evaluarea în punctul de mijloc al subintervalului produce o aproximație mai precisă. Formula devine:

$$\text{Integrală} \approx \sum_{i=1}^n f\left(x_i + \frac{\Delta x}{2}\right) \cdot \Delta x$$

Implementare în MATLAB:

```
matab Copy code

% Definim funcția f(x)
f = @(x) x.^2 + 2.*x - x./3;

% Limitele de integrare
a = 2; % Limita inferioară
b = 3; % Limita superioară

% Numărul de subintervale
n = 1000;

% Lungimea fiecărui subinterval
dx = (b - a) / n;

% Vectorul de puncte de mijloc ale fiecărui subinterval
x_mid = linspace(a + dx/2, b - dx/2, n);

% Aplicăm metoda dreptunghiurilor îmbunătățită
integral = sum(f(x_mid) * dx);

% Afișăm rezultatul
fprintf('Valoarea aproximativă cu punctul de mijloc: %.6f\n', integral);
```

Această metodă produce rezultate mai apropiate de valoarea exactă a integralei, fără a crește numărul de subintervale.

2. Creșterea dinamică a numărului de subintervale (n):

Introduceți o condiție pentru a opri calculul atunci când eroarea este sub o valoare dată. De exemplu:

Implementare:

matalab

 Copy code

```
% Definim funcția f(x)
f = @(x) x.^2 + 2.*x - x./3;

% Limitele de integrare
a = 2;
b = 3;

% Inițializăm numărul de subintervale și toleranța
n = 10;
tolerance = 1e-6;
error = Inf;

% Calculăm valoarea exactă analitică pentru comparație (opțional)
exact_value = 11.5;

while error > tolerance
    % Creștem numărul de subintervale
    n = n * 2;

    % Lungimea fiecărui subinterval
    dx = (b - a) / n;

    % Vectorul de puncte din stânga fiecărui subinterval
    x = linspace(a, b-dx, n);

    % Aproximăm integrala
    approx = sum(f(x) * dx);

    % Calculăm eroarea relativă
    error = abs(exact_value - approx);

end

% Afișăm rezultatul
fprintf('Valoarea aproximativă: %.6f\n', approx);
fprintf('Numărul de subintervale utilizate: %d\n', n);
```

Această abordare optimizează numărul de subintervale necesare pentru a obține un rezultat precis.

3. Comparația cu alte metode numerice (trapez sau Simpson):

Deși cerința specifică este metoda dreptunghiurilor, ar fi interesant să comparați acest rezultat cu metode mai avansate, cum ar fi metoda trapezului sau metoda Simpson, care sunt mai precise pentru aceeași valoare a n .

4. Vizualizarea grafică a metodei:

Adăugați o reprezentare grafică a funcției și a dreptunghiurilor pentru a înțelege mai bine metoda și precizia ei.

Exemplu:

matlab

 Copy code

```
% Graficul funcției și al dreptunghiurilor
x_vals = linspace(a, b, 1000);
y_vals = f(x_vals);

figure;
hold on;
plot(x_vals, y_vals, 'b-', 'LineWidth', 1.5); % Graficul funcției
for i = 1:n
    rectangle('Position', [x(i), 0, dx, f(x(i))], 'EdgeColor', 'r');
end
hold off;
title('Metoda dreptunghiurilor');
xlabel('x');
ylabel('f(x)');
```

5. Vectorizare completă:

În MATLAB, vectorizarea calculelor îmbunătățește semnificativ performanța. Codul deja este în mare parte vectorizat, dar asigurați-vă că toate operațiile se aplică pe vectori, nu în bucle.

Aceste îmbunătățiri fie cresc precizia metodei, fie optimizează timpul de execuție, fie adaugă claritate vizuală. Dacă doriți să implementăm una dintre aceste sugestii sau să discutăm mai detaliat, sunt aici!

    