Laborator 8. Integrarea numerica

Consideratii teoretice

Integrarea este un concept fundamental in rezolvarea unui numar mare de probleme in inginerie si stiinta. In multe situatii nu pot fi obtinute solutii analitice, fiind necesara aplicarea metodelor de integrare numerica.

Aceasta lucrare de laborator prezinta cateva functii MatLab de integrare numerica.

In MatLab pot fi calculate integrale definite de forma urmatoare:

$$q = \int_a^b g(x)dx$$

Unele din cele mai folosite functii de integrare MatLab sunt **quad(), quad8(), trapz().**

quad() – calculeaza integrala prin metoda recursiva Simpson;
quad8() – calculeaza integrala prin metoda adaptiv recursiva Newton Cotes de ordinul 8
trapz() – calculeaza integrala prin metoda trapezelor

Daca integrala se aproximeaza prin trapeze, iar intervalul [a,b] este impartit in n sectiuni egale, adica pasul si abscisa curenta sunt:

$h = \frac{b-a}{n} > 0$ unde x_j = a+jh si $1 \leq j \leq n$, putem exprima integrala ca fiind :

$$S_T = \frac{h}{2}\left[y_a + y_b + 2\sum_{j=1}^{n-1} y_j \right] \quad \text{unde } y_j = f(x_j)$$

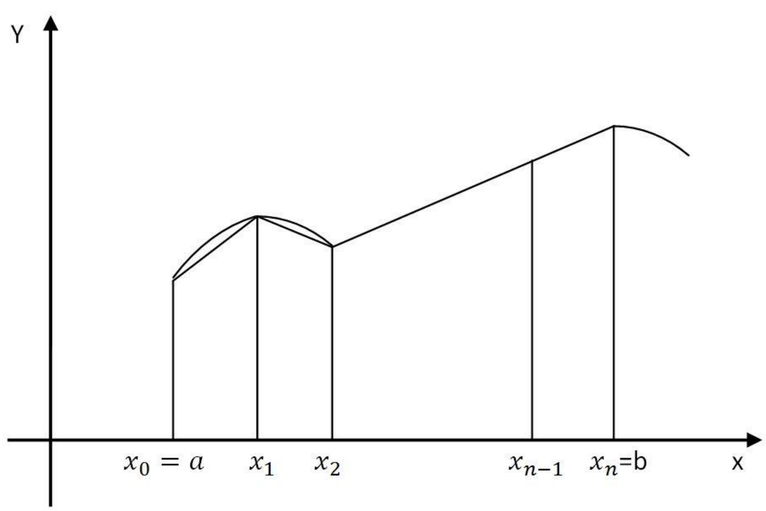


Fig. 1

Daca integrala este apoximata prin aria subsectiunilor unor curbe patratice, iar intervalul[a,b] este divizat in 2n sectiuni egale :

$h = \frac{b-a}{2n} = x_{j+1} - x_j > 0$ unde x_j = a+jh si 1≤ j≤ n integrala poate fi exprimata :

$S_T = \frac{h}{3}[y_a + y_b + 4(y_1 + y_3 + ... + y_{2n-1}) + 2(y_2 + y_4 + ... + y_{2n-2})]$, cunoscuta sub numele de formula generalizata a lui Simpson.

Pentru calculul integralelor definite prin metoda trapezelor, in MatLab se utilizeaza functia **trapz().** Aceasta functie presupune ca functia de integrat *f* este precizata sub forma de valori numerice, $\{y_k = f(x_k)\}_{k=1,...,n}$, in punctele echidistante $\{x_k\}_{k=1,..,n}$(x_1=a , x_n=b) ale intervalului de integrare [a , b].

Sintaxa axestei functii este :

rezultat = trapz(x,y)

unde : x – reprezinta vectorul valorilor $\{x_k\}$

y – reprezinta vectorul valorilor $\{y_k = f(x_k)\}$

rezultat – reprezinta aproximarea cu metoda trapezelor a integralei definite $\int_a^b f(x)dx$.

Exemplu1 :

Fie functia *f* data prin puncte de relatiile $f(x_i) = \frac{\sin(x_i)}{i^2+1} \cdot \cos\left(\frac{i}{x_i}\right)$, $x_i = \pi + i \cdot \frac{\pi}{30}$, i = 1,2,...,150. Sa se calculeze valoarea integralei functiei *f* prin metoda trapezelor.

```
for i=1:150
x(i)=pi+i*pi/30;                        % generam vectorul x
y(i)=sin(x(i))/(i^2+1)*cos(i/x(i)); % generam vectorul y = f(x)
end
rezultat=trapz(x,y) % calculam integrala folosind metoda trapezelor
```

In urma acestor instructiuni in fereastra de comenzi MatLab vom obtine:
rezultat = -0.0025

Exemplu2 :

Fie integrala $q = \int_0^1 e^{-x^2} dx$. Sa se calculeze valoarea integralei.

```
x = linspace(0,1,5);      % definim vectorul x
y = exp(-x5.^2);          % definim functia y = f(x)
q = trapz(x5, y5)         % calculam integrala cu trapz
```

In urma instructiunilor de mai sus in fereastra de comenzi MatLab vom obtine
q = 0.7430.

O alta metoda de a calcula aceasta integrala este folosirea functiei **quad().**

Functia MatLab **quad** realizeaza calcului integralei definite a unei functii prin metoda adaptiv-recursiva Simpson (o varianta mai performanta a metodei Simpson, pasul de parcurgere a intervalului de integrare este calculat implicit de catre functia Matlab). Functia **quad** presupune ca functia de integrat f este cunoscuta sub expresia sa analitica, y = f(x).

Sintaxa acestei functii este :

rezultat = **quad** (nume_fisier,a,b)
rezultat = **quad** (nume_fisier,a,b,precizia)
rezultat = **quad** (nume_fisier,a,b,precizia,control)

unde: nume_fisier – reprezinta un sir de caractere care contine numele fisierului-functie in care a fost scrisa expresia functiei de integrat.

a si b – reprezinta limitele de integrare (capetele intervalului [a,b] pe care se realizeaza integrarea)

precizia – este un argument optional prin care se poate modifica precizia implicita 10^{-6}

control – este un argument obtional care controleaza afisarea pe ecran a valorilor intermediare

rezultat – reprezinta aproximarea integralei definite $\int_a^b f(x)dx$

Similar functiei **quad** este si sintaxa functiei **quad8**.

Pentru exemplul anterior sa calculam integrala cu ajutoul functiei **quad**.

Definim o functie in fisierul q.m.

```
function y = q(x)
y = exp(-x.^2);
```

, iar intr-un fisier de tip script scriem:

```
rez = quad('q',0,1), urmand ca in fereastra de comenzi sa se returneze
```

rez = 0.7468

Exemplu3 :

Sa se calculeze $\int_0^{\pi} \ln(x+1)\cdot\sin(x)dx$

```
function y = z(x)
y=log(x+1).*sin(x);
```

Si apelam dintr-un script pentru a vedea rezultatul

```
rez1= quad('z',0,pi)
```

rez1 = 1.8113

Semnalele care pot fi descompuse cel mai simplu in semnale elementare sunt semnalele periodice. Un semnal s(t) este periodic daca satistafe relatia: s(t) = s(t+T), unde T reprezinta perioada de repetitie a semnalului.

Forma trigonometrica a unui astfel de semnal este:

$$s(t) = \frac{a_0}{2} + \sum_{n=1}^{\infty}\left[a_n \cos nwt + b_n \sin nwt\right],$$

unde coeficientii formei trigonometrice se calculeaza astfel:

$$a_n = \frac{2}{T}\int_{-\frac{T}{2}}^{\frac{T}{2}} s(t)\cos nwtdt \quad ; \quad b_n = \frac{2}{T}\int_{-\frac{T}{2}}^{\frac{T}{2}} s(t)\sin nwtdt \quad ; \; a_0 = \frac{2}{T}\int_{-\frac{T}{2}}^{\frac{T}{2}} s(t)dt$$

Pornind de la aceste formule vom descrie un program MatLab de reprezentare a unor semnale periodice cunoscute:

Problema 1:

Fisierul script v-a contine:

```
clear all;
close all;
clc;

f=50;         % definim frecventa semnalului
w=2.*pi.*f;   % definim pulsatia
T=1./f;       % definim perioada semnalului
A=10;         % definim amplitudinea semnalului
tip=3;        % alegem tipul semnalului

t=0:T./100:T;   % vectorul timp
g=generare_g(t,f,A,tip); % atribuim un semnal variabilei g
plot(t,g,'r');grid on;hold on; %afisam g
xlabel('t [s]');
ylabel('g');
title('tensiunea g');
%-------------------------------------
Na=6;
    a0=(2./T).*quad(@generare_g,-T/2,T/2,[],[],f,A,tip) %calc coef a0

for k=1:Na
    a(k)=(2./T).*quad(@coef_Fa,-T/2,T/2,[],[],f,A,k,tip);%calculam an
    b(k)=(2./T).*quad(@coef_Fb,-T/2,T/2,[],[],f,A,k,tip);%calculam bn
end
disp('        a         b');  %afisam in fereastra de comenzi caractere
disp([a',b']); % afisam in fereastra de comenzi valorile a si b

%=============calculam semnalul din coeficienti======================
f_ap=a0./2+a(1).*cos(w.*t)+a(2).*cos(2.*w.*t)+a(3).*cos(3.*w.*t)+...
           b(1).*sin(w.*t)+b(2).*sin(2.*w.*t)+b(3).*sin(3.*w.*t)
%======================================================================
 plot(t,f_ap,'b');grid on;
```

Dupa cum putem observa, acest fisier script apeleaza 3 fisiere-functie:
generare_q; coef_Fa; coef_Fb;

Fisierul generara_q.m

```
function g=generare_g(t,f,A,tip)
T=1./f;
w=2.*pi.*f;
switch tip
    case 1
        % Generez un triunghi bipolar
      g=2.*A.*asin(sin(w.*t))./pi;
           case 2
        % Generez un triunghi monopolar 2 f
        g1=2.*A.*asin(sin(w.*t))./pi;
        g=abs(g1);
            case 3
      % Generez un triunghi monopolar  f
        g1=2.*A.*asin(sin(w.*t))./pi;
        g2=abs(g1);
        g=(g1+g2)./2;
            case 4
             % Generez o tensiune alternativa sinusoidala
        g=A.*sin(w.*t);
            case 5
             % Generez o tensiune redresata monopolara 2 f
        g1=A.*sin(w.*t);
        g=abs(g1);
            case 6
             % Generez o tensiune redresata monopolara  f
        g1=A.*sin(w.*t);
        g2=abs(g1);
        g=(g1+g2)./2;

end
g;
```

Acest fisier permite selectarea unui semnal dorit si transmite prin parametrul de iesire *g* semnalul ales catre fisierul script.

Fisierul coef_Fa.m

```
function fa=coef_Fa(t,f,A,k,tip)
T=1./f;
w=2.*pi.*f;
g=generare_g(t,f,A,tip);
fa=g.*cos(k.*w.*t);
```

Fisierul coef_Fb.m

```
function fb=coef_Fb(t,f,A,k,tip)
T=1./f;
w=2.*pi.*f;
g=generare_g(t,f,A,tip);
fb=g.*sin(k.*w.*t);
```

Cele doua fisiere **coef** contin functia ce caracterizeaza calculul unui coeficient, urmand apoi a fi integrata in fisierul script.

Rezultatul acestor instructiuni este:

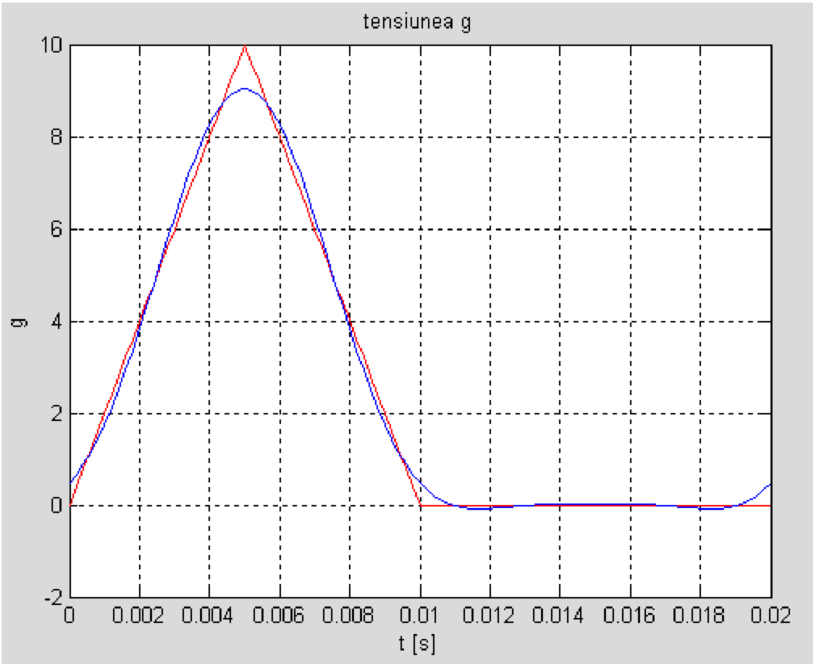


Fig. 2

Cu rosu este reprezentat semnalul initial, iar cu albastru este reprezentat semnalul construit din coeficientii acestuia.

Tema :

Sa se gaseasca o modalitate de a imbunatati aproximarea semnalului cu ajutorul coeficientilor.

Problema 2:

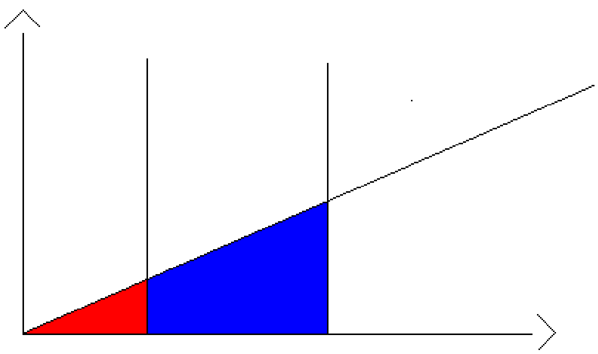


Fig.3

Pornind de la graficul din figura 3, sa se determine aria sectiunii albastre stiind ca acest grafic reprezinta o dreapta ce trece prin origine.

Pentru rezolvarea aceastei probleme vom scrie urmatoarele instructiuni in MatLab:

Fisierul script :

```
clear all;
close all;
clc;
A=1;
t0=0;
tf=0.02;
pas=tf./100;
t=t0:pas:tf;
ta=tf./5;
tb=3.*tf./4;

m=20;   % Dreapta prin origine [ 0 0]
g=m.*t;
It=m.*(tb.^2-ta.^2)./2; % formula teoretica de calcul
t1=[ta,tb,tb,ta];     t2=[0,ta,ta];   %suprafetele
g1=[0,0,m.*tb,m*ta];  g2=[0,0,m.*ta]; %suprafetele

plot(t,g,'r',[ta,ta],[0,A],'b',[tb,tb],[0,A],'m');grid on;hold on;
fill(t1,g1,'b',t2,g2,'r');
xlabel('t [s]');
ylabel('g (t)');
text(ta,0.5,'ta');
text(tb,0.5,'tb');

[Inta]=integrare_g(t,g,ta);             % integrare 0 - ta
[Intb]=integrare_g(t,g,tb);             % integrare 0 - tb
Int=Intb-Inta;                          % integrare ta -tb tb > ta
er=It-Int;   % eroarea absoluta
disp('   Intb,      Inta,       Int,         It,         er');
disp([Intb,Inta,Int,It,er]);
```

Fisierul-functie pentru calculul suprafetelor :

```
function [Int]=integrare_g(t,g,ta)
%Integreaza g(t) intre 0 si ta > 0
k=2;
%---------------------------
Ia=0;
while t(k)<=ta
    Ia=Ia+(t(k)-t(k-1)).*g(k);
k=k+1;
end
Int=Ia;
%---------------------------
```

Aceste instructiuni vor avea ca rezultat :

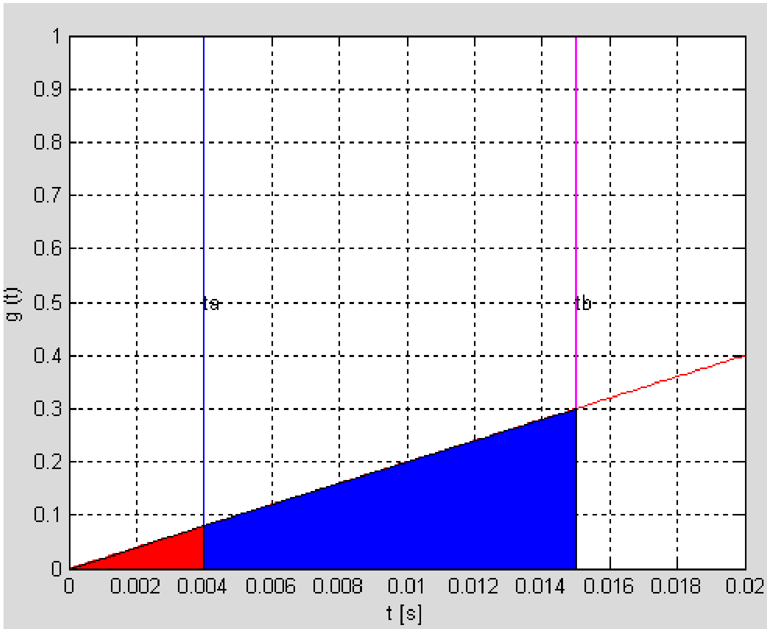


Fig. 4

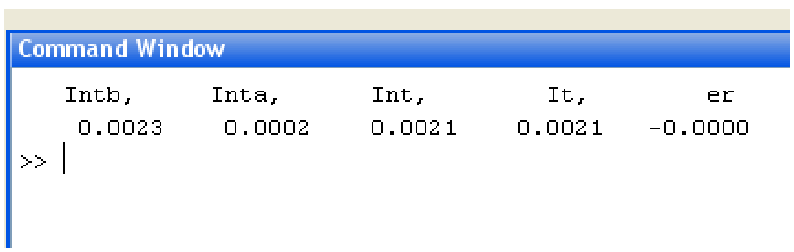


Fig. 5

O alta metoda de apoximare este folosirea sumei Riemann. Problema urmatoare propune calculul suprafetei unei drepte si a unui semnal sinusoidal redresat.

Problema 3:

```
clear all;
close;
clc;
t=0:0.1:3;      % infuenta pasului 0.001
f=4.*t;
t1=2;           % t1<3;
f1=4.*t1;
%-----------------------
fig=figure('Name','Calculul suprafetelor');
%------------------
subplot(221);
plot(t,f,'r');grid on;
xlabel('t [s]');
ylabel('f =4.*t');
%--------------------
```

```
subplot(223);
stem(t,f,'r');grid on;
xlabel('t [s]');
ylabel('f =4.*t');
%-------------Riemann
k=2;
Sf=0;
while t(k)<=t1
      Sf=Sf+(t(k)-t(k-1)).*f(k-1);
k=k+1;
end
Sif=Sf               % suprafata unei drepte
%----------------
St=t1.*f1./2;        % Teoretic
er_a=St-Sif;         % Erore absoluta
disp(['      St       Sif       er_a']);
disp([St,Sif,er_a]); % mesaj in fereastra de comenzi


%----------------------------- Tensiune redresata
f=50;
w=2.*pi.*f;
T=1./f;
pas=T./100;
t=0:pas:T;
A=4;
g1=A.*sin(w.*t);   % semnal sinusoidal
g2=abs(g1);        % semnal monopolar 2f
g=(g1+g2)./2;      % semnal monopolar f
%-------------------------

k=2;
Sg=0;
while t(k)<=T./2                 % t(k)<=t1
      Sg=Sg+(t(k)-t(k-1)).*g(k-1); % f(k-1)
k=k+1;
end
Sig=Sg   % suprafata unui semnal redresat

subplot(222);
plot(t,g,'r');grid on;hold on;
fill(t,g,'y');
xlabel('t [s]');
ylabel('g =(g1+g2)./2');
%--------------------
subplot(224);
stem(t,g,'r');grid on;
xlabel('t [s]');
ylabel('g');
%------------------
%------------------
St_g=2.*A./w;            % Teoretic
er_ag=St_g-Sig;          % Erore absoluta
disp(['     St_g      Sig      er_ag']);
disp([St_g,Sig,er_ag]);
```

Rezultatul este :

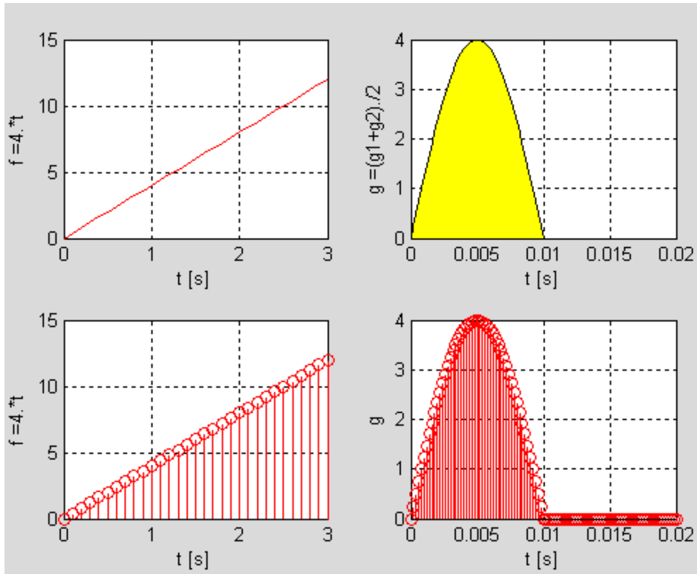
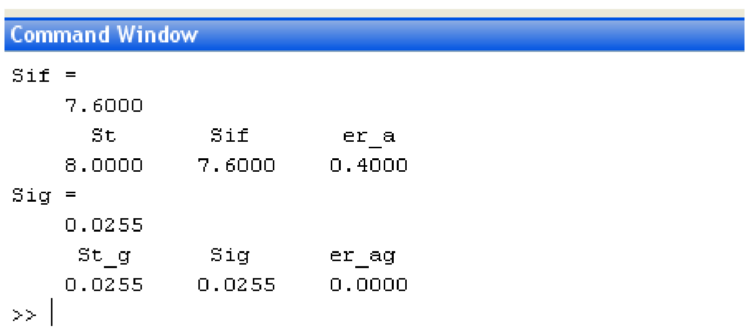


Fig. 6



```
Command Window
Sif =
    7.6000
      St        Sif       er_a
    8.0000    7.6000    0.4000
Sig =
    0.0255
     St_g       Sig       er_ag
    0.0255    0.0255    0.0000
>>
```

Fig. 7

Tema:

Realizati o interfata care sa permita selectarea unui semnal dorit de la problema 1 cu ajutorul unui buton de tip popupmenu.

Pentru un semnal triunghiular monopolar f calculati suprafata acestuia si determinati coeficientii de ordinul 3.

```
% Triunghi monopolar f
 g1=2.*A.*asin(sin(w.*t))./pi;
 g2=abs(g1);
 g=(g1+g2)./2;
```