Tema lab04

Table of Contents

Interpolarea Spline Liniara

```
function [y] = SplineL(X, Y, x)
  n = length(X)-1;
  for j=1:n
      a(j) = Y(j);
      b(j) = (Y(j+1) - Y(j)) / (X(j+1)-X(j));
  end
  for i=1:length(x)
    for j=1:n
      if x(i) >= X(j) && x(i) <= X(j+1)
          y(i) = a(j) + b(j)*(x(i)-X(j));
      end
    end
  end
end
```

Metoda Spline Patratica

```
function [y,z] = SplineP(X, Y, fpa, x)
  n = length(X)-1;
  for j=1:n
      a(j) = Y(j);
  end
  b(1) = fpa;
  h = X(2)-X(1);
  for j=1:n-1
      b(j+1) = (2*(Y(j+1) - Y(j)) / h) - b(j);
  end

  for j=1:n
```

```
        c(j) = (Y(j+1) - Y(j) - h*b(j))/(h*h);
    end

    for i=1:length(x)
      for j=1:n
        if x(i) >= X(j) && x(i) <= X(j+1)
            y(i) = a(j) + b(j)*(x(i)-X(j)) + c(j)*(x(i)-X(j))^2;
            z(i) = b(j) + 2*c(j)*(x(i)-X(j));
        end
      end
    end
end
```

Interpolarea Spline Cubica

```
%Ij = [xj, xj+1) j=1,n-1
%in = [xn, xn+1]
%Sj(x) =
function [y,z,t] = SplineCubic(X,Y,x,fpa,fpb)
  n = length(X)-1;
  for j=1:n
      a(j) = Y(j);
  end
  B(1,1) = 1;
  B(n+1,n+1) = 1;
  for j=2:n
      B(j,j) = 4;
      B(j,j-1) = 1;
      B(j,j+1) = 1;
  end
  w(1) = fpa;
  w(n+1) = fpb;
  h = X(2)-X(1);
  for j=2:n
      w(j) = 3*(Y(j+1)-Y(j-1))/h;
  end
  w = w';
  b = B\w;
  for j=1:n
      d(j) = -2*(Y(j+1)-Y(j))/(h*h*h) + (b(j+1)+b(j))/(h*h);
      c(j) = 3*(Y(j+1)-Y(j))/(h*h) - (b(j+1)+2*b(j))/h;
  end
  for i=1:length(x)
      for j=1:n
          if x(i)<=X(j+1) && x(i)>=X(j)
              S = a(j) + b(j)*(x(i)-X(j)) + c(j)*(x(i)-X(j))*(x(i)-
X(j)) + d(j)*(x(i)-X(j))*(x(i)-X(j))*(x(i)-X(j));
              Sp = b(j) + 2*c(j)*(x(i)-X(j)) + 3*d(j)*(x(i)-
X(j))*(x(i)-X(j));
              Ss = 2*c(j) + 6*d(j)*(x(i)-X(j));
              break;
```

```
            end
        end
        y(i) = S;
        z(i) = Sp;
        t(i) = Ss;
    end
end
```

Derivarea Numerica

```
function [dy] = DerivNum(x,y,metoda)
  m = length(x)-1;
  switch metoda
      case 'diferente finite progresive'
          for i=2:m
              dy(i) = (y(i+1)-y(i)) / (x(i+1)-x(i));
          end
      case 'diferente finite regresive'
          for i=2:m
              dy(i) = (y(i)-y(i-1)) / (x(i)-x(i-1));
          end
      case 'diferente finite centrale'
          for i=2:m
              dy(i) = (y(i+1)-y(i-1)) / (x(i+1)-x(i-1));
          end
  end
end
```

Metoda Richardson

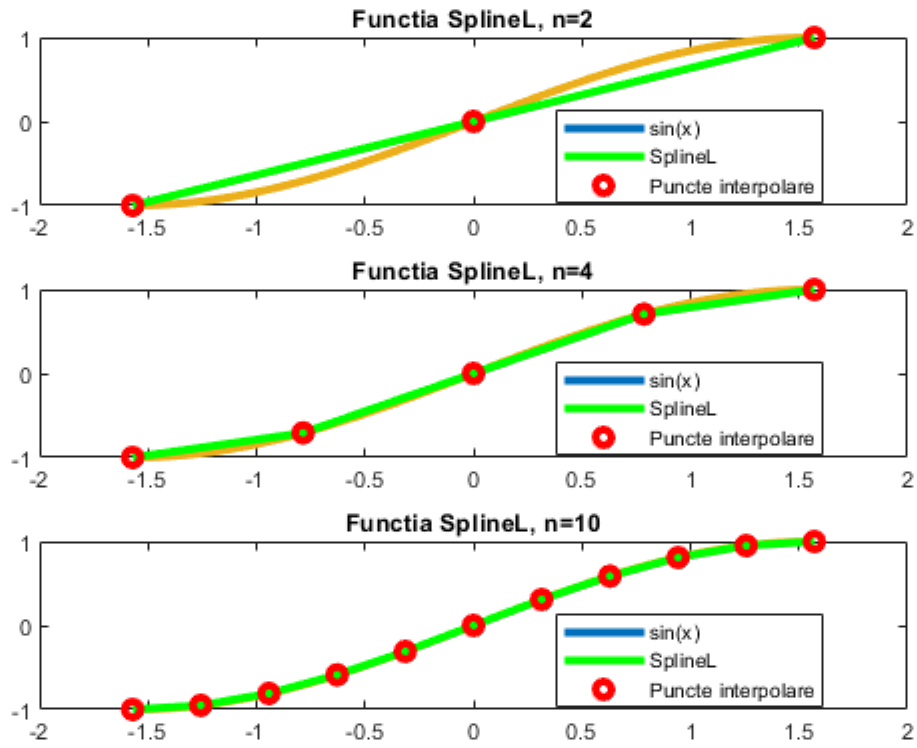
```
function [df] = MetRichardson(f, x, h, n)
  phi = @(x,h)(f(x+h)-f(x))/h;
  for k=1:length(x)
      for i=1:n
          Q(i,1) = phi(x(k),h/(2^(i-1)));
      end
      for i=2:n
          for j=2:i
              Q(i,j) = Q(i,j-1) + (Q(i,j-1)-Q(i-1,j-1))/(2^(j-1)-1);
          end
      end
      df(k) = Q(n,n);
  end
end
```

Metoda Richardson2

```
function [d2f] = MetRichardson2(f, x, h, n)
  phi = @(x,h)(f(x+h)-2*f(x)+f(x-h))/(h*h);
  for k=1:length(x)
      for i=1:n
          Q(i,1) = phi(x(k),h/(2^(i-1)));
      end
      for i=2:n
          for j=2:i
              Q(i,j) = Q(i,j-1) + (Q(i,j-1)-Q(i-1,j-1))/(2^(j-1)-1);
          end
      end
      d2f(k) = Q(n,n);
  end
end
```

Problema 2

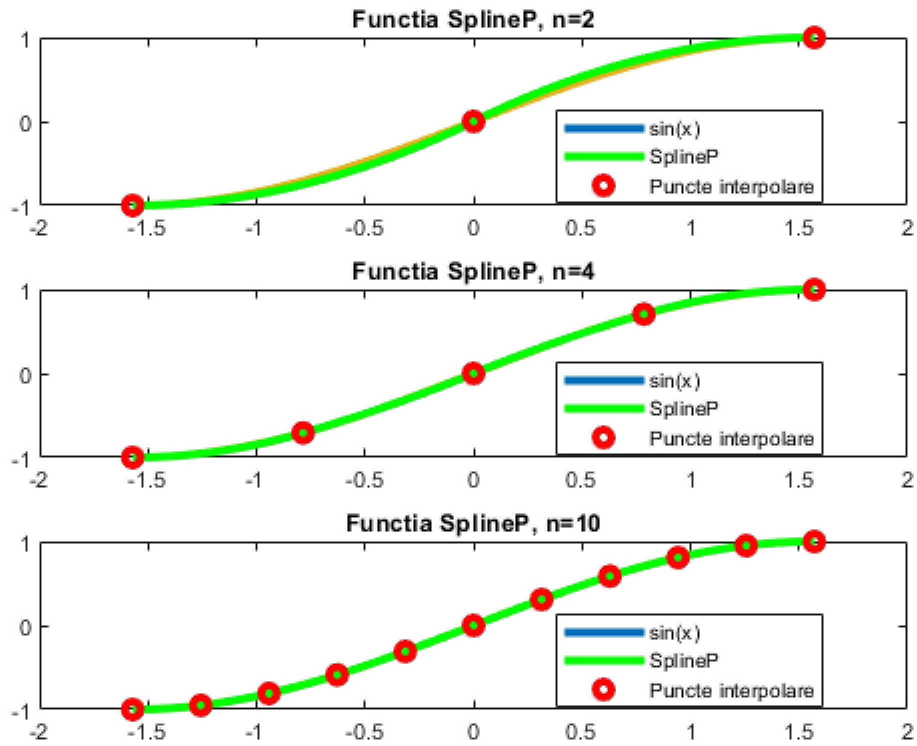
```
f = @(x)sin(x);
xmin = -pi/2;
xmax = pi/2;
x = linspace(xmin,xmax,100);

N = [2, 4, 10];

for idx=1:3
    n = N(idx);
    X = linspace(xmin,xmax,n+1);
    Y = f(X);

    figure(1);
    subplot(3,1,idx);
    plot(x,f(x), 'LineWidth', 3);
    hold on;
    plot(x, SplineL(X, Y, x), 'g', 'LineWidth', 3);
    plot(X, f(X), 'or', 'LineWidth', 3);
    legend('sin(x)','SplineL', 'Puncte
 interpolare', 'Location', 'Best');
    title('Functia SplineL, n='+string(n));
end
```

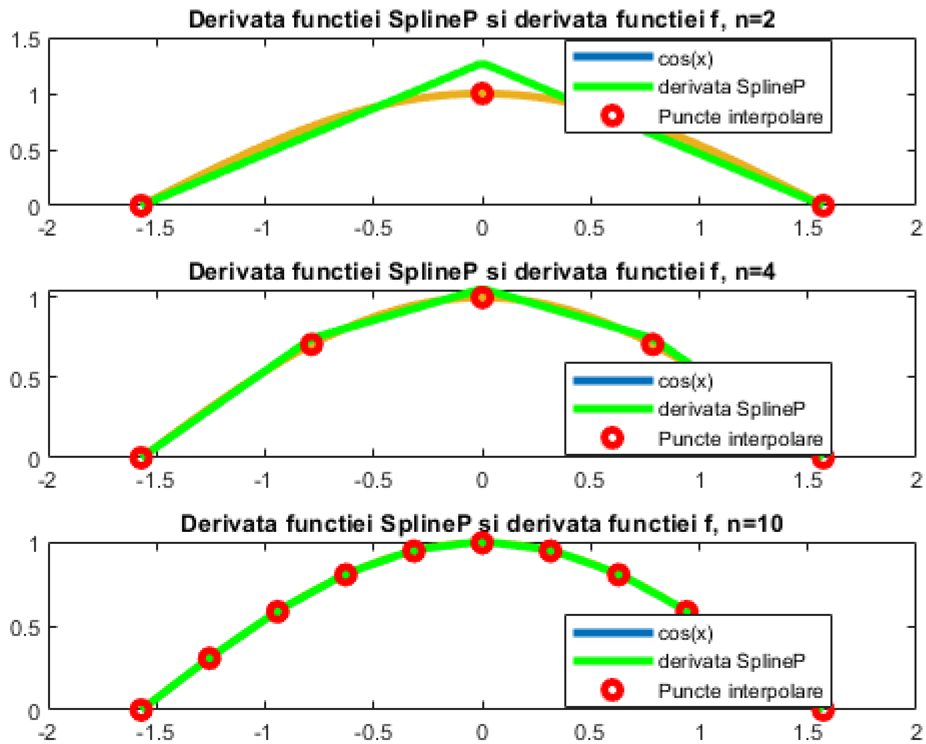


Problema 4

```
f = @(x)sin(x);
fp = @(x)cos(x);
xmin = -pi/2;
xmax = pi/2;
x = linspace(xmin,xmax,100);
fpa = fp(xmin);

N = [2, 4, 10];
for idx=1:3
    n = N(idx);
    X = linspace(xmin,xmax,n+1);
    Y = f(X);

    figure(2);
    subplot(3,1,idx);
    plot(x,f(x), 'LineWidth', 3);
    hold on;

    [yP, zP] = SplineP(X, Y, fpa, x);
    plot(x, yP, 'g', 'LineWidth', 3);
    plot(X, f(X), 'or', 'LineWidth', 3);
    legend('sin(x)','SplineP', 'Puncte
 interpolare', 'Location', 'Best');
```

```
    title('Functia SplineP, n='+string(n));

    figure(3);
    subplot(3,1,idx);
    plot(x, fp(x), 'LineWidth', 3);
    hold on;
    plot(x, zP, 'g', 'LineWidth', 3);
    plot(X, fp(X), 'or', 'LineWidth', 3);
    legend('cos(x)','derivata SplineP','Puncte
 interpolare', 'Location', 'Best');
    title('Derivata functiei SplineP si derivata functiei f,
 n='+string(n));
end
```

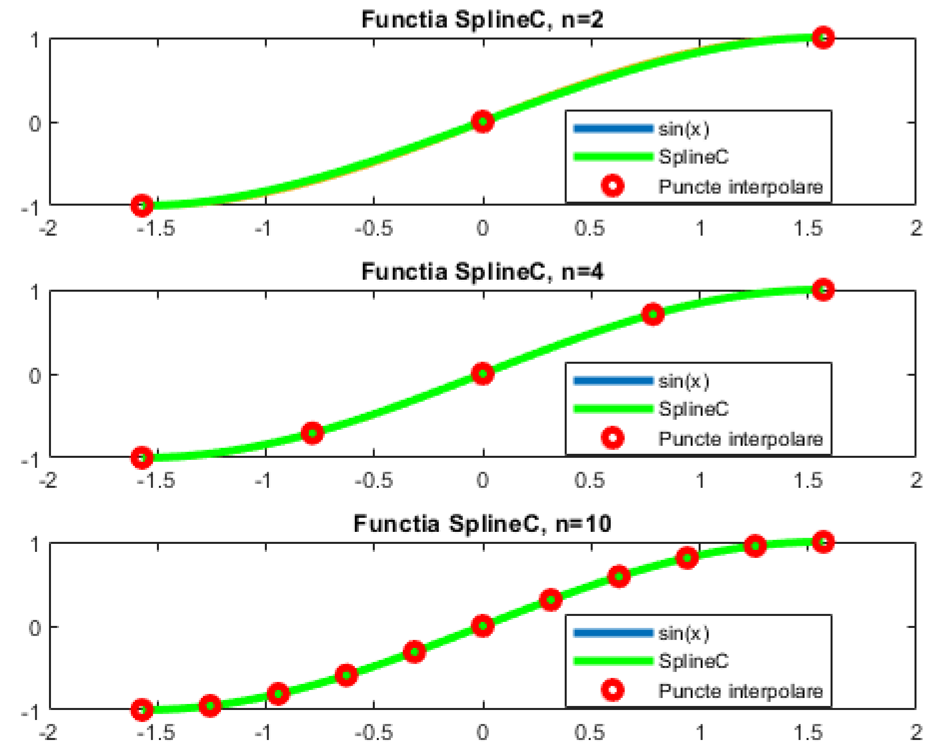


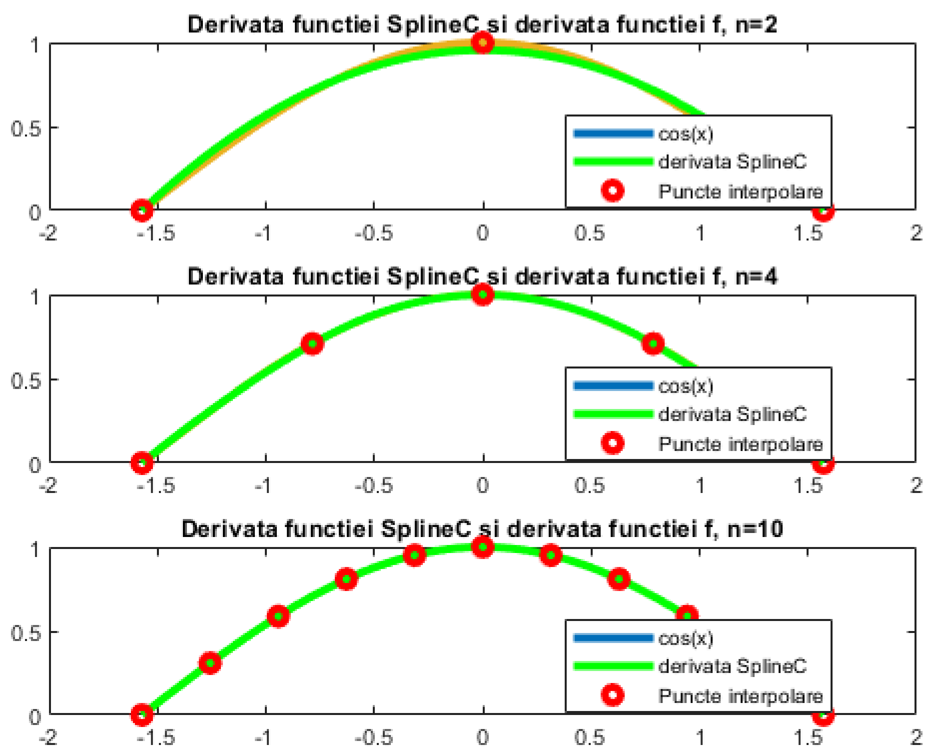


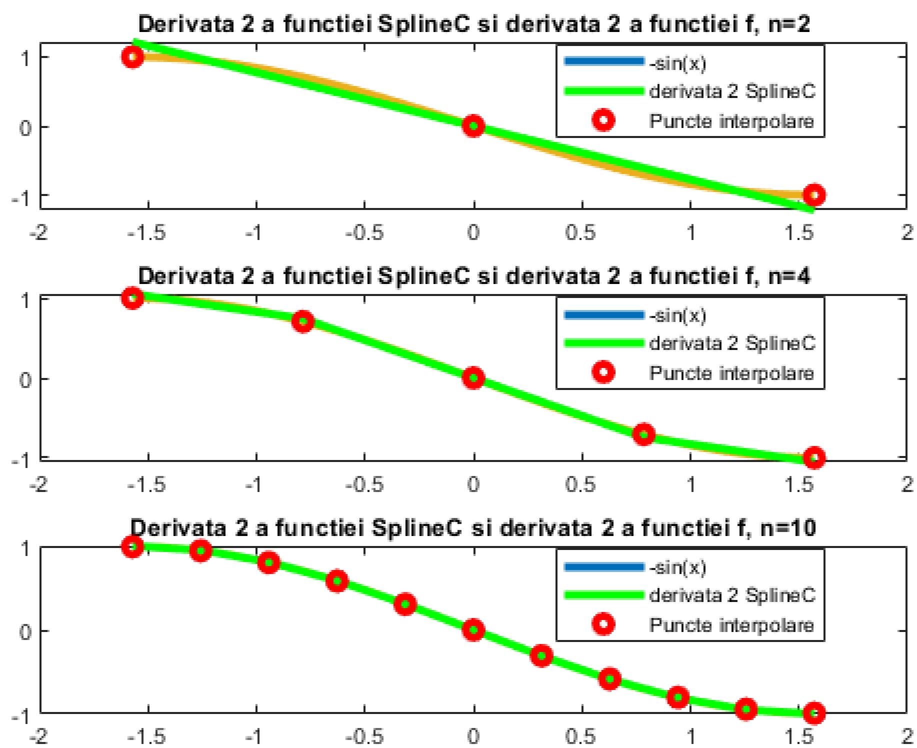
Problema 5

```
f = @(x)sin(x);
fp = @(x)cos(x);
fs = @(x)-sin(x);

xmin = -pi/2;
xmax = pi/2;
x = linspace(xmin,xmax,100);
fpa = fp(xmin);
fpb = fp(xmax);

N = [2, 4, 10];
for idx=1:3
    n = N(idx);
    X = linspace(xmin,xmax,n+1);
    Y = f(X);

    figure(4);
    subplot(3,1,idx);
    plot(x,f(x), 'LineWidth', 3);
    hold on;

    [yC, zC, tC] = SplineCubic(X, Y, x, fpa, fpb);
    plot(x, yC, 'g', 'LineWidth', 3);
```

```
    plot(X, f(X), 'or', 'LineWidth', 3);
    legend('sin(x)','SplineC', 'Puncte
 interpolare', 'Location', 'Best');
    title('Functia SplineC, n='+string(n));

    figure(5);
    subplot(3,1,idx);
    plot(x, fp(x), 'LineWidth', 3);
    hold on;
    plot(x, zC, 'g', 'LineWidth', 3);
    plot(X, fp(X), 'or', 'LineWidth', 3);
    legend('cos(x)','derivata SplineC','Puncte
 interpolare', 'Location', 'Best');
    title('Derivata functiei SplineC si derivata functiei f,
 n='+string(n));

    figure(6);
    subplot(3,1,idx);
    plot(x, fs(x), 'LineWidth', 3);
    hold on;
    plot(x, tC, 'g', 'LineWidth', 3);
    plot(X, fs(X), 'or', 'LineWidth', 3);
    legend('-sin(x)','derivata 2 SplineC','Puncte
 interpolare', 'Location', 'Best');
    title('Derivata 2 a functiei SplineC si derivata 2 a functiei f,
 n='+string(n));
end
```


Functia SplineC, n=2
Functia SplineC, n=4
Functia SplineC, n=10
sin(x)
SplineC
Puncte interpolare

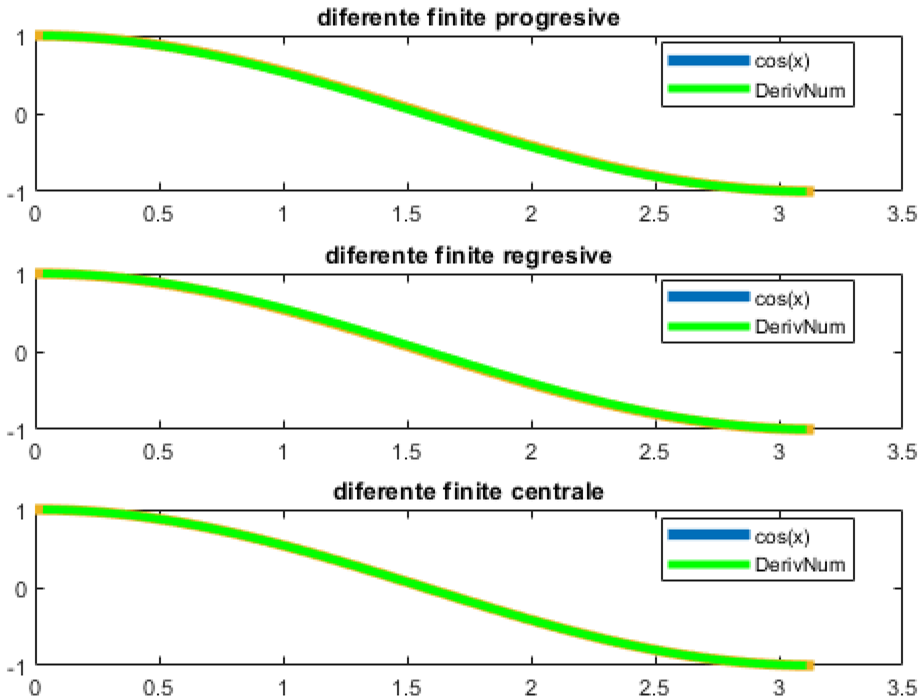

Derivata functiei SplineC si derivata functiei f, n=2
Derivata functiei SplineC si derivata functiei f, n=4
Derivata functiei SplineC si derivata functiei f, n=10
cos(x)
derivata SplineC
Puncte interpolare

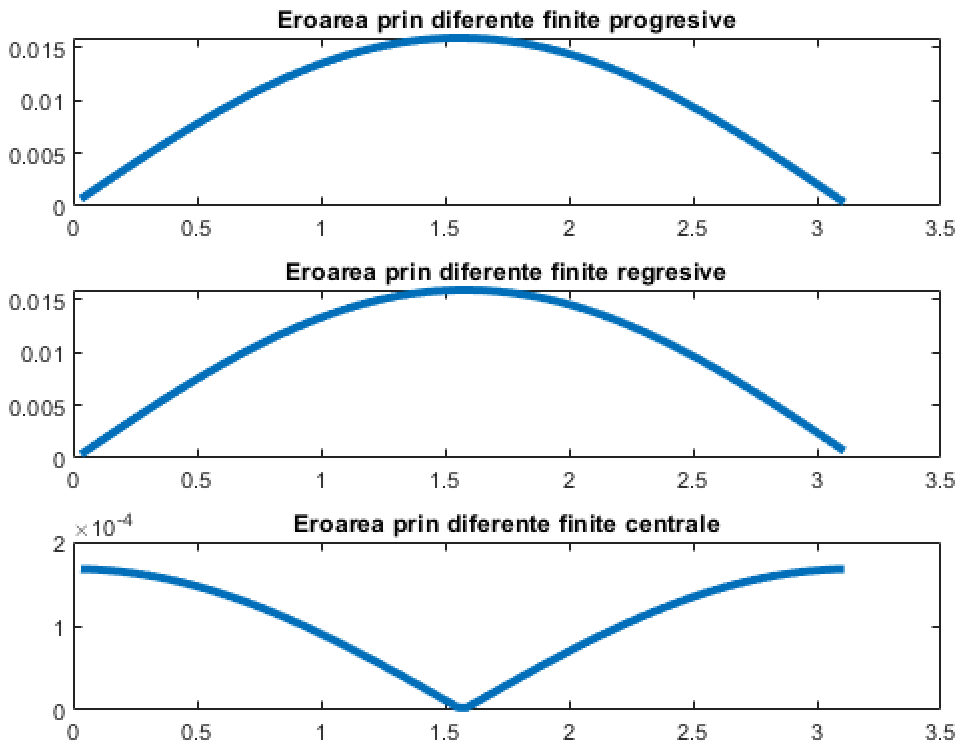


Problema 6

```
f = @(x)sin(x);
fp = @(x)cos(x);
a = 0;
b = pi;
m = 100;

x = linspace(a,b,m);
y = f(x);

metode = ["diferente finite progresive", "diferente finite
 regresive", "diferente finite centrale"];
for idx = 1:3
    metoda = metode(idx);
    dy = DerivNum(x,y,metoda);
    figure(7);
    subplot(3,1,idx);
    plot(x,fp(x), 'LineWidth', 4);
    hold on;
    plot(x(2:length(x)-1),dy(2:length(dy)), 'g', 'LineWidth', 3);
    legend('cos(x)', 'DerivNum', 'Location', 'Best');
    title(metoda);

    figure(8);
```

```
    subplot(3,1,idx);
    plot(x(2:length(x)-1),abs(dy(2:length(dy)) -
 fp(x(2:length(x)-1))), 'LineWidth', 3);
    title("Eroarea prin " + metoda);
end
```

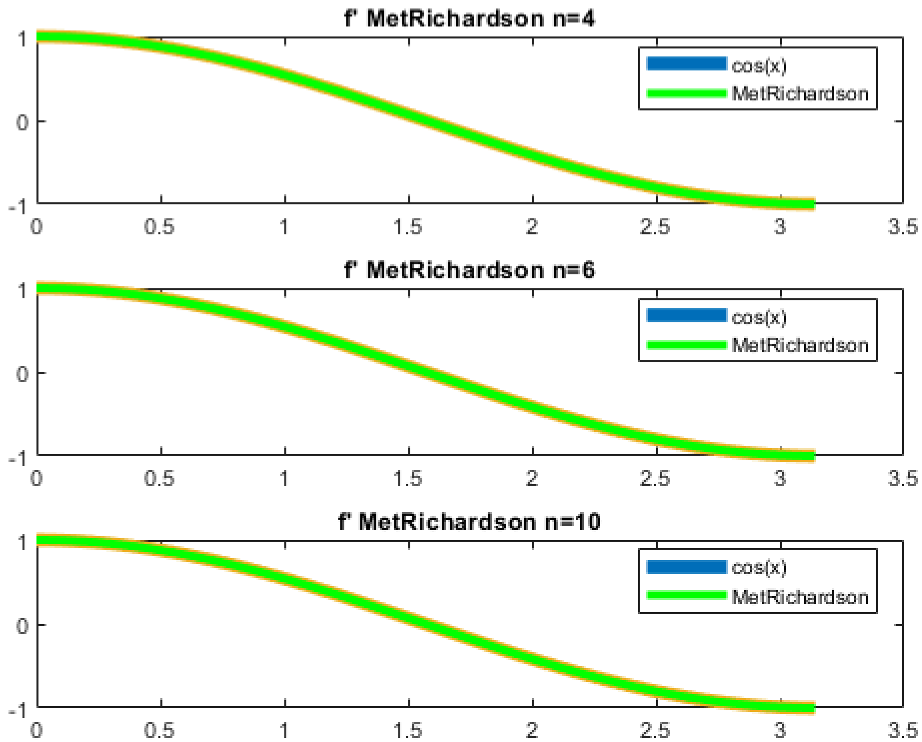


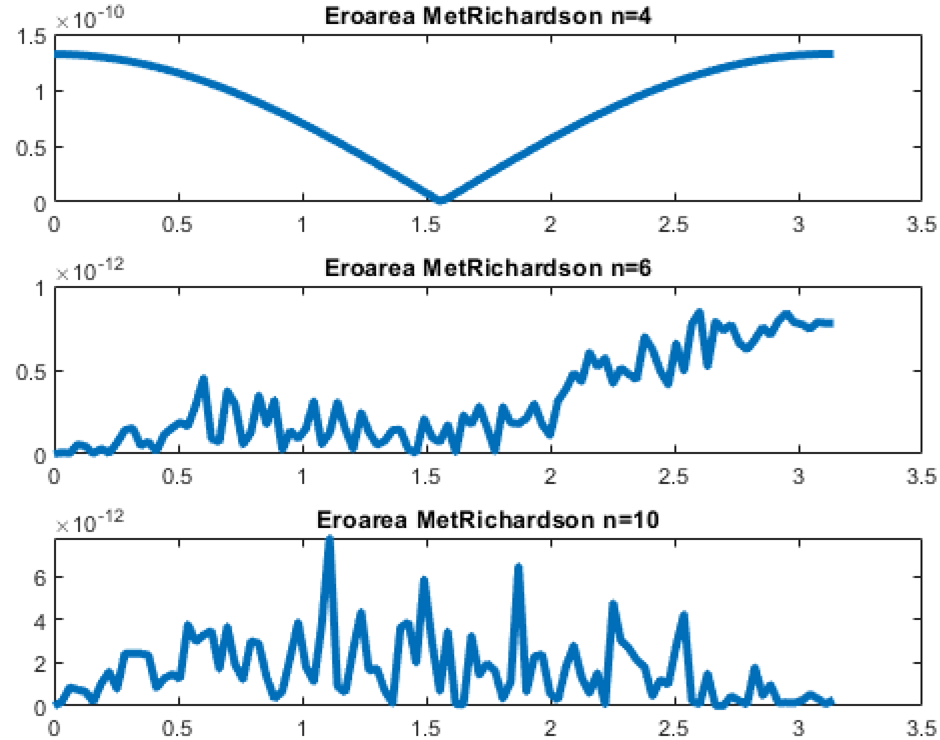


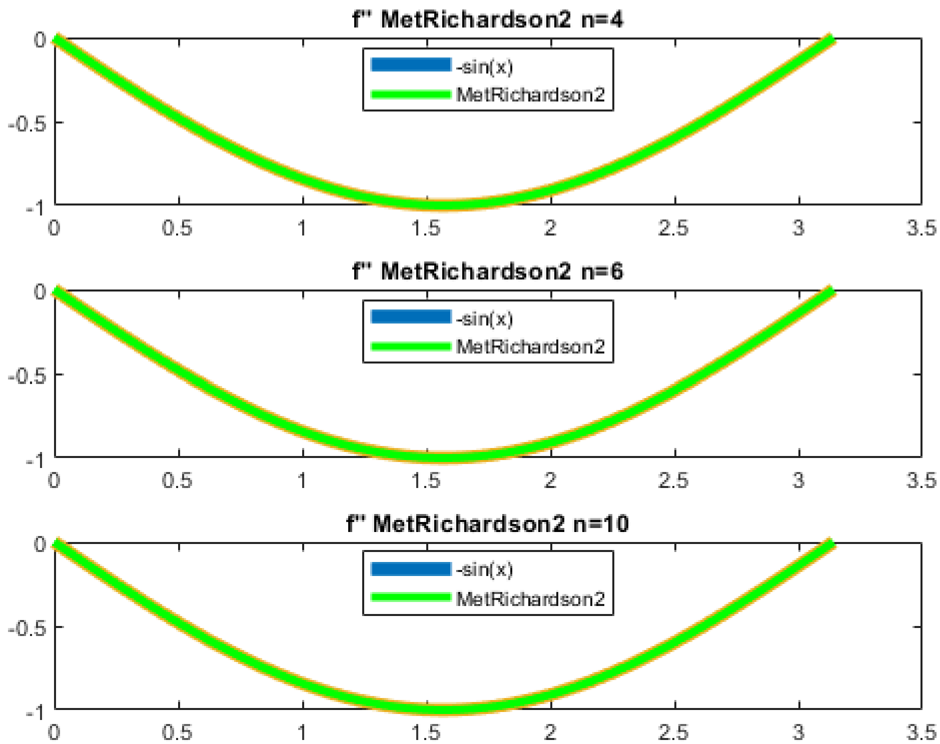
Problema 7

```
f = @(x)sin(x);
fp = @(x)cos(x);
fs = @(x)-sin(x);
a = 0;
b = pi;
N = [4, 6, 10];
x=linspace(a,b,100);
h=x(2)-x(1);

for idx = 1:3
   n = N(idx);

   figure(9);
   subplot(3,1,idx);
   df = MetRichardson(f, x, h, n);
   plot(x,fp(x), 'LineWidth', 5);
   hold on;
   plot(x, df, 'g', 'LineWidth', 3);
   legend('cos(x)', 'MetRichardson', 'Location', 'Best');
   title("f' MetRichardson n="+string(n));

   figure(10);
   subplot(3,1,idx);
```

```
    plot(x, abs(df - fp(x)), 'LineWidth', 3);
    title("Eroarea MetRichardson n="+string(n));

    figure(11);
    subplot(3,1,idx);
    d2f = MetRichardson2(f, x, h, n-1);
    plot(x,fs(x), 'LineWidth', 5);
    hold on;
    plot(x, d2f, 'g', 'LineWidth', 3);
    legend('-sin(x)', 'MetRichardson2', 'Location', 'Best');
    title("f'' MetRichardson2 n="+string(n));
end
```




Eroarea MetRichardson n=4
×10^-10
Eroarea MetRichardson n=6
×10^-12
Eroarea MetRichardson n=10
×10^-12


f'' MetRichardson2 n=4
-sin(x)
MetRichardson2
f'' MetRichardson2 n=6
-sin(x)
MetRichardson2
f'' MetRichardson2 n=10
-sin(x)
MetRichardson2

Published with MATLAB® R2018a