Table of Contents

P7 : Cortes y García

```
clear all
format long
```

Ejercicio 1 - Ejemplo

```
A = toeplitz([1 3 5 7],[1 2 3 4])


A =

     1     2     3     4
     3     1     2     3
     5     3     1     2
     7     5     3     1
```

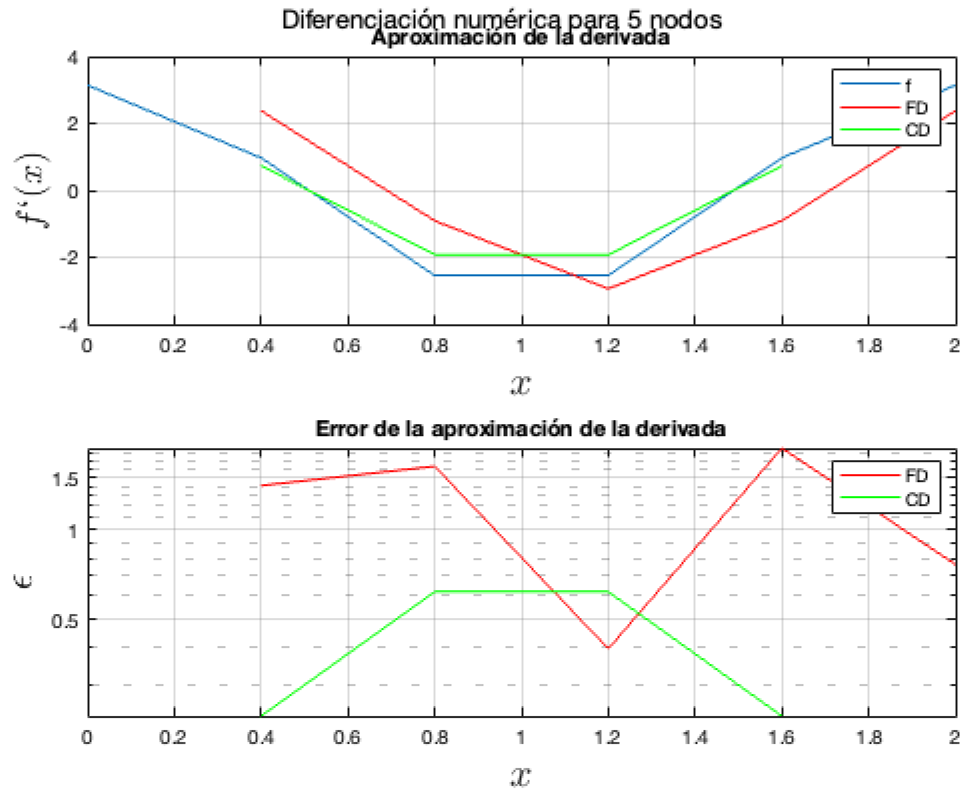
Ejercicio 2,3

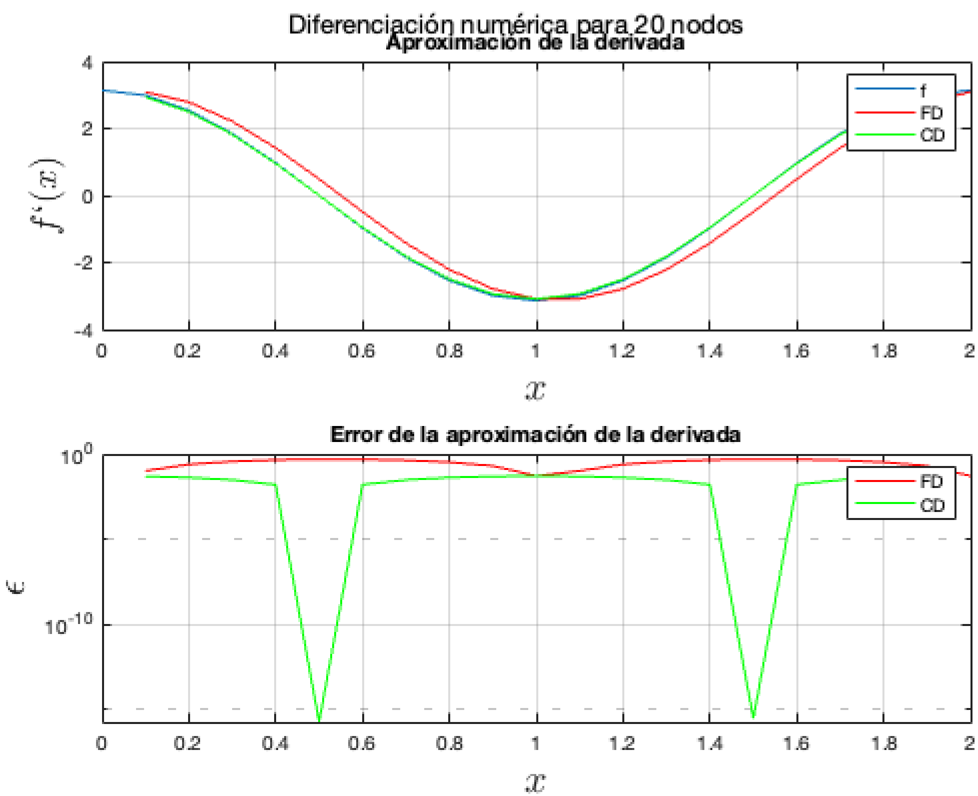
```
f = @(x) sin(pi.*x);
df = @(x) pi.*cos(pi.*x);

nn = [5 20 40 80];

fig = 1; % Para numerar las figuras

for n = nn
    h = 2/n;

    % Matriz FD
    % Queremos que la matriz resultante sea de dimensiones n x (n+1),
    % puesto que tenemos n+1 nodos en los que evaluarla. Por lo tanto,
    % necesitamos que el vector v que vamos a emplear para generar
    % la matriz sea de tamaño n+1, y el vector u sea de tamaño n.
    v_FD = zeros(1, n+1);
    v_FD(1) = -1;
    v_FD(2) = 1;
    u_FD = zeros(1, n);
    u_FD(1) = -1;
    FD = (1 / h) * toeplitz(u_FD, v_FD);
```

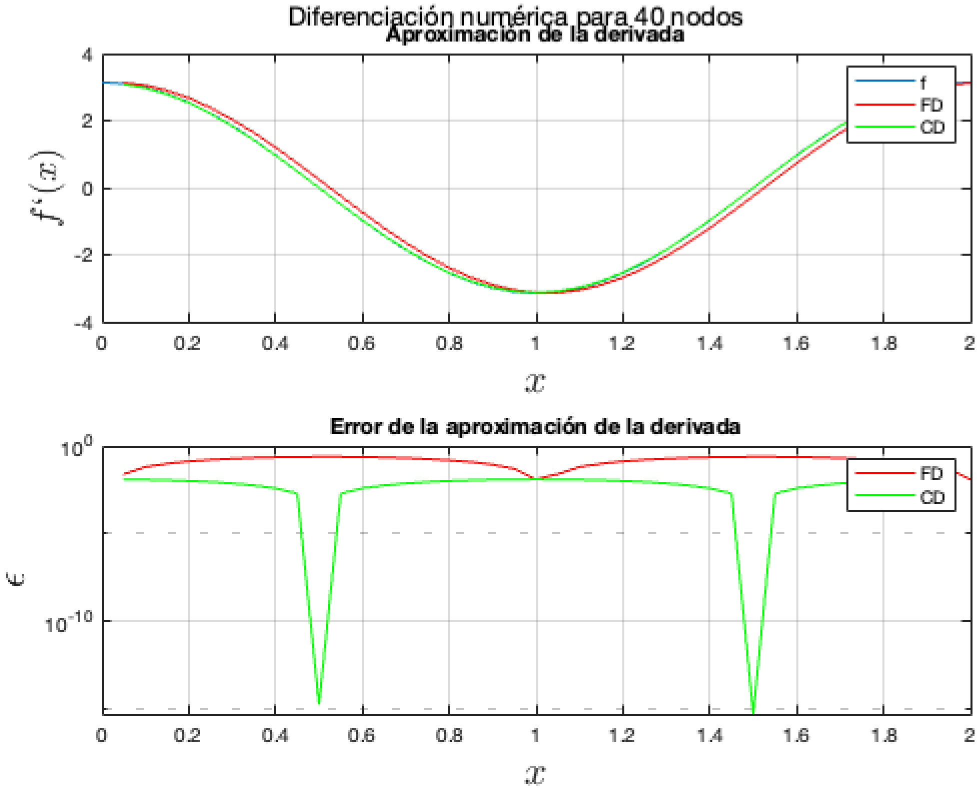
```
    % Matriz CD
    % De manera similar, puesto que queremos que resulte una matriz de
    % dimensiones (n-1) x (n+1), necesitaremos que v tenga dimensión n
+1 y
    % u, n-1.
    v_CD = zeros(1, n+1);
    v_CD(1) = -0.5;
    v_CD(3) = 0.5;
    u_CD = zeros(1, n-1);
    u_CD(1) = -0.5;
    CD = (1/ h) * toeplitz(u_CD, v_CD);

    if n == 5 % Solo mostraremos las matrices cuando n es 5
        disp(FD)
        disp(CD)
    end

    % Generamos los nodos equiespaciados
    jj = [0:n]';
    xj = (2 / n) * jj;

    F = f(xj); % evaluación de la función
    dF = df(xj); % evaluación de la derivada de la función

    dF_FD = FD*F; % Aproximación con el método FD
    dF_CD = CD*F; % Aproximación con el método CD

    figure(fig) % Representación para n
    subplot(2, 1, 1)
    plot(xj, dF, 'linewidth', 1)
    hold on
    plot(xj(2:end), dF_FD, 'linewidth', 1, 'color', 'r')
    plot(xj(2:end-1), dF_CD, 'linewidth', 1, 'color', 'g')
    hold off
    grid on
    title('Aproximación de la derivada', 'Interpreter', 'Latex')
    legend('f', 'FD', 'CD')
    xlabel('$x$', 'Interpreter', 'Latex', 'FontSize', 20)
    ylabel('$f`(x)$', 'Interpreter', 'Latex', 'FontSize', 18)

    % Calculamos los errores absolutos respectivos
    e_FD = abs(dF(2:end) - dF_FD);
    e_CD = abs(dF(2:end-1) - dF_CD);

    subplot(2, 1, 2)
    semilogy(xj(2:end), e_FD, 'linewidth', 1, 'color', 'r')
    hold on
    semilogy(xj(2:end-1), e_CD, 'linewidth', 1, 'color', 'g')
    hold off
    grid on
    title('Error de la aproximación de la derivada')
    legend('FD', 'CD')
    xlabel('$x$', 'Interpreter', 'Latex', 'FontSize', 20)
```

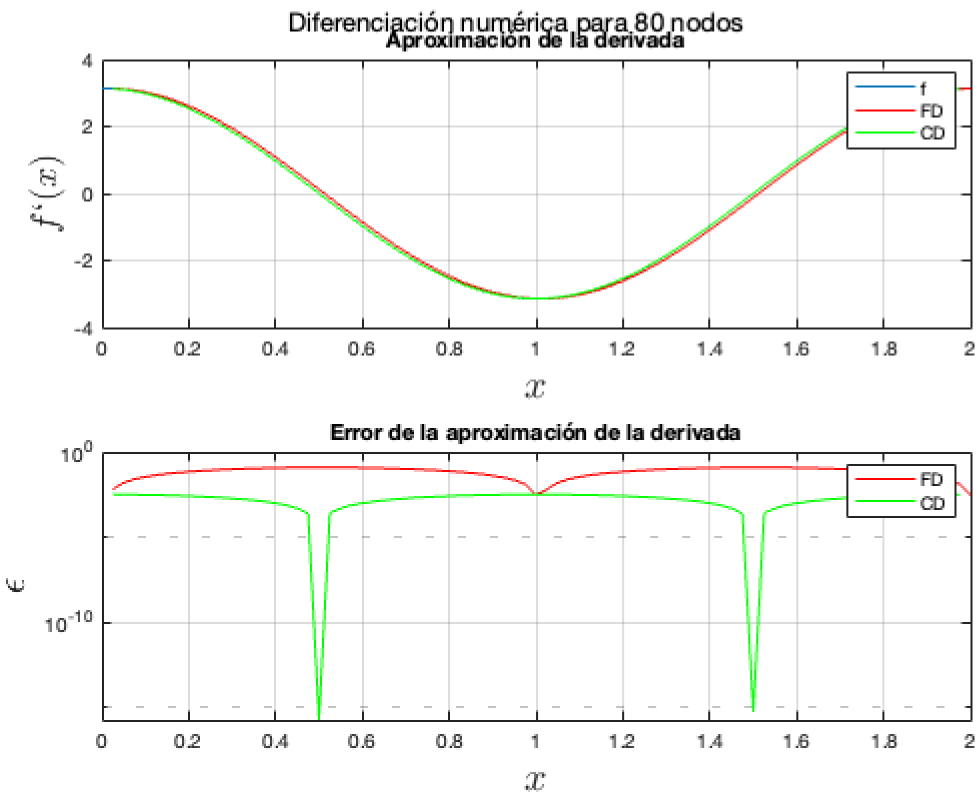
```
    ylabel('$\epsilon$', 'Interpreter', 'Latex', 'FontSize', 20)

    t = ['Diferenciación numérica para ',num2str(n), ' nodos'];
    sgtitle(t)
    xlim([xj(1), xj(end)])

    fig = fig + 1;
end

% Podemos ver que los puntos en los que los errores se
% aproximan al 0 según el método, son aquellos en los que la
 aproximación
% de la derivada intersecciona con la propia derivada en la gráfica de
% representación.

% El error siempre es más pequeño cuando usamos el método FD, que
 tiene
% mucho sentido si tenemos en cuenta que el orden teórico de
 convergencia
% de la aproximación al aplicar el residuo de Cauchy es de orden
 cuadrático
% (tendirá a 0 más rápido que en el caso del método FD).
```

```
  Columns 1 through 3

  -2.500000000000000   2.500000000000000                   0
                   0  -2.500000000000000   2.500000000000000
                   0                   0  -2.500000000000000
                   0                   0                   0
                   0                   0                   0

  Columns 4 through 6

                   0                   0                   0
                   0                   0                   0
   2.500000000000000                   0                   0
  -2.500000000000000   2.500000000000000                   0
                   0  -2.500000000000000   2.500000000000000

  Columns 1 through 3

  -1.250000000000000                   0   1.250000000000000
                   0  -1.250000000000000                   0
                   0                   0  -1.250000000000000
                   0                   0                   0

  Columns 4 through 6

                   0                   0                   0
   1.250000000000000                   0                   0
                   0   1.250000000000000                   0
  -1.250000000000000                   0   1.250000000000000
```


Diferenciación numérica para 5 nodos
Aproximación de la derivada
f
FD
CD
f'(x)
x
Error de la aproximación de la derivada
FD
CD
ϵ
x


Diferenciación numérica para 20 nodos
Aproximación de la derivada
f
FD
CD
f'(x)
x
Error de la aproximación de la derivada
FD
CD
ϵ
x


Diferenciación numérica para 40 nodos
Aproximación de la derivada
f'(x)
x
f
FD
CD
Error de la aproximación de la derivada
ϵ
x
FD
CD

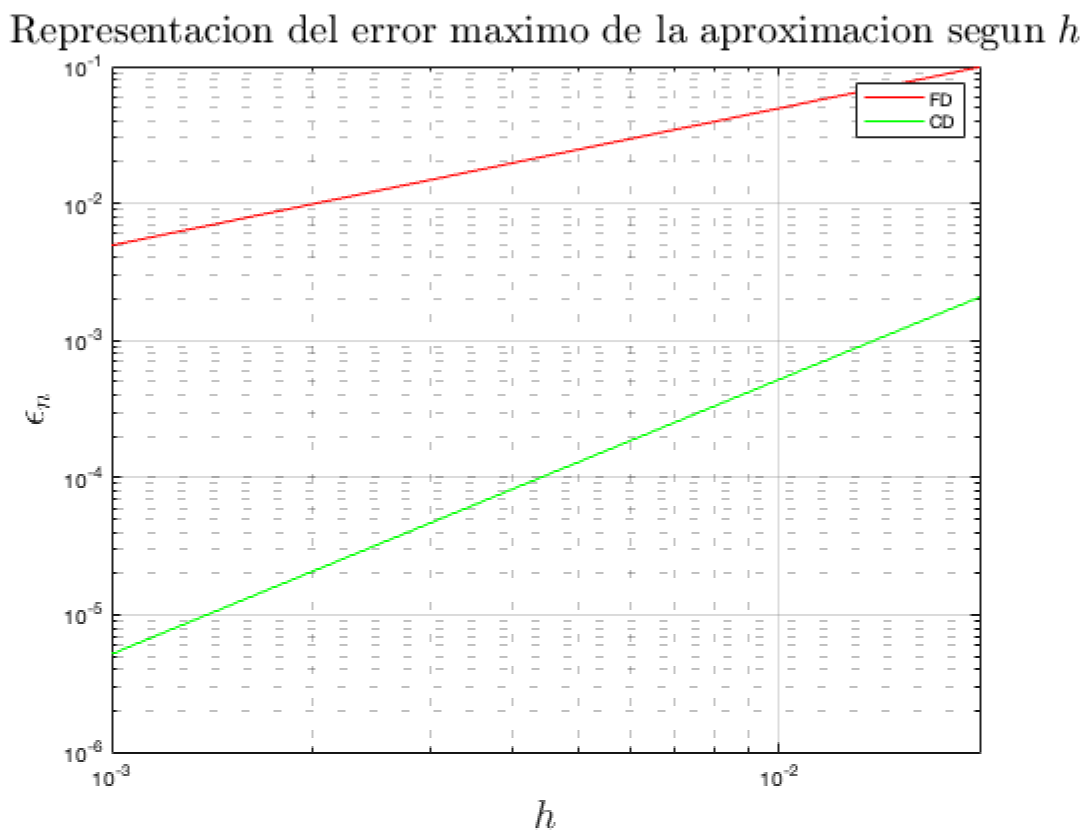

Diferenciación numérica para 80 nodos
Aproximación de la derivada
f'(x)
x
f
FD
CD
Error de la aproximación de la derivada
ϵ
x
FD
CD

Ejercicio 4

```
NN = [100:100:2000];

e_max_FD = [];
e_max_CD = [];
hh = [];

for N = NN
    h = 2 / N;

    % Matriz FD
    v_FD = zeros(1, N+1);
    v_FD(1) = -1;
    v_FD(2) = 1;
    u_FD = zeros(1, N);
    u_FD(1) = -1;
    FD = (1 / h) * toeplitz(u_FD, v_FD);

    % Matriz CD
    v_CD = zeros(1, N+1);
    v_CD(1) = -0.5;
    v_CD(3) = 0.5;
    u_CD = zeros(1, N-1);
    u_CD(1) = -0.5;
    CD = (1/ h) * toeplitz(u_CD, v_CD);

    % Generamos los nodos equiespaciados
    jj = [0:N]';
    Xj = (2 / N) * jj;

    F = f(Xj); % evaluación de la función
    dF = df(Xj); % evaluación de la derivada de la función

    dF_FD = FD*F; % Aproximación con el método FD
    dF_CD = CD*F; % Aproximación con el método CD

    % Calculamos los errores absolutos respectivos
    e_FD = abs(dF(2:end) - dF_FD);
    e_CD = abs(dF(2:end-1) - dF_CD);

    % Actualizamos vectores para la representación gráfica
    hh = [hh h];
    e_max_FD = [e_max_FD max(e_FD)];
    e_max_CD = [e_max_CD max(e_CD)];
end

figure(5)
loglog(hh, e_max_FD, 'LineWidth', 1, 'Color', 'r');
hold on
loglog(hh, e_max_CD, 'LineWidth', 1, 'Color', 'g');
hold off
grid on
```

```
legend('FD', 'CD')
title('Representacion del error maximo de la aproximacion segun $h
$', 'Interpreter', 'Latex', 'FontSize', 20)
xlabel('$h$', 'Interpreter', 'Latex', 'FontSize', 20)
ylabel('$\epsilon_n$', 'Interpreter', 'Latex', 'FontSize', 20)

r_FD = polyfit(log10(hh), log10(e_max_FD), 1);
r_CD = polyfit(log10(hh), log10(e_max_CD), 1);

m_FD = r_FD(1) % Aprox 1
m_CD = r_CD(1) % Aprox 2

% Vemos que la pendiente de la gráfica del error respecto a h es 1
 para el
% método FD y 2 para el método CD. Esto tiene sentido, puesto que el
 método
% FD es de orden lineal mientras que el CD es de orden cuadrático,
 como
% vimos en la clase de teoría. Intuitivamente, tiene sentido que
 aumentar
% el número de nodos que usamos para hacer la aproximación de la
 derivada
% suponga una mejora en la precisión de la misma.
```

```
m_FD =

   0.999930861818534


m_CD =

   1.999958515641246
```



Published with MATLAB® R2020b