



CONTROL 1

Problema 1 (2 pts): Programe en Matlab la siguiente serie:

$$s(n) = 1 + \frac{1}{4} + \frac{1}{9} + \frac{1}{16} + \dots + \frac{1}{n^2}, \quad \lim_{n \rightarrow \infty} s(n) = \frac{\pi^2}{6}$$

Usando su programa complete la siguiente tabla:

n	$s(n)$	$error(n)$ [%]
10		
50		
250		
5000		

Cuadro 1: Tabla de resultados

Solución:

```
%  
clc, clear all  
  
% Rutina para sumar los terminos de la serie  
n=5000;  
for i=1:n  
    s(i) = (1/i)^2;  
end  
  
% Suma para n = 10, n = 50, n = 250, n = 5000  
su(1) = sum(s(1:10));  
su(2) = sum(s(1:50));  
su(3) = sum(s(1:250));  
su(4) = sum(s(1:5000));  
  
% Valor teorico  
teo = pi^2 / 6;
```

```

err(1) = abs((teo - su(1))/ teo)*100;
err(2) = abs((teo - su(2))/ teo)*100;
err(3) = abs((teo - su(3))/ teo)*100;
err(4) = abs((teo - su(4))/ teo)*100;

% Salida de Resultados
disp(['+-----+'])
disp(['|_n_          ' 's(n)_          ' 'error(%)_|'])
disp(['+-----+'])
disp(['|_10_         |_' num2str(su(1)) '  |_' num2str(err(1))])
disp(['|_50_         |_' num2str(su(2)) '  |_' num2str(err(2))])
disp(['|_250_        |_' num2str(su(3)) '  |_' num2str(err(3))])
disp(['|_5000_       |_' num2str(su(4)) '  |_' num2str(err(4))])
disp(['+-----+'])
%
%
```

El resultado por pantalla en Matlab es el siguiente

```

+-----+
| n          s(n)          error(%) |
+-----+
| 10         | 1.5498 | 5.7854
| 50         | 1.6251 | 1.2038
| 250        | 1.6409 | 0.24269
| 5000       | 1.6447 | 0.012157
+-----+
```

Problema 2 (2 pts): Utilice el método de la bisección (programado en Matlab usando *while...end*) para encontrar el coeficiente de fricción (μ) según la fórmula de Colebrook:

$$\frac{1}{\sqrt{\mu}} = 1,14 - 2 \log \left(\frac{e}{D} + \frac{9,35}{Re\sqrt{\mu}} \right), \quad (1)$$

usando los siguientes parámetros: $D = 0,1$ m $e = 0,0001$ m y $Re = 5 \times 10^6$. Con sus resultados complete la siguiente tabla:

Expresión para calcular el error	
Tolerancia para la convergencia	
Intervalo de búsqueda	
Número de iteraciones totales	

Cuadro 2: Tabla de resultados

Sugerencia: Haga un gráfico de la ecuación para determinar el intervalo de búsqueda.

Solución:

En general, es aconsejable hacer una gráfica de la función para determinar el intervalo de búsqueda necesario en el método de la bisección:

```

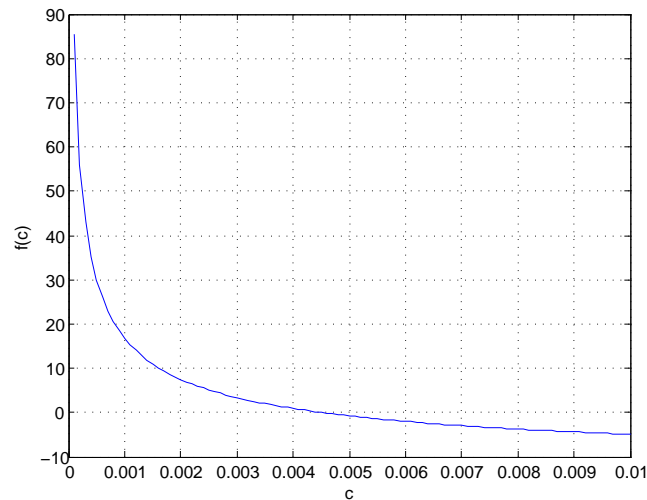
clc, clear all

% Parametros de la ecuacion de Colebrook
D = 0.1;
e = 0.0001;
Re = 5e6;

% Funcion (igualando a cero)
% Definicion en linea usando "@(c)", donde c es la variable
% independiente de la funcion:
f =@(c) 1./sqrt(c) - 1.14 + 2 * log(e/D + 9.35 ./ (Re .* sqrt(c)))

% Se define un vector "c" para realizar un "plot" de la funcion
c = 0:0.0001:0.01;
plot(c, f(c))
xlabel('c')
ylabel('f(c)')
grid on
%
%
```

El resultado es el siguiente:



En la figura podemos apreciar que la función $f(c)$ cruza el eje x entre los valores $x_a = 0,004$ y $x_b = 0,005$. Estos valores proporcionan el intervalo de búsqueda en el método de la bisección.

Usando las definiciones anteriores (parámetros y función), la rutina de Matlab para obtener la raíz de una función usando el método de la bisección está dada por:

```
% Intervalo de busqueda I=[a,b]
a = 0.004;
b = 0.005;

% Comprobacion de cambio de signo
if f(a)*f(b)>0
    % NO hay cambio de signo y se muestra un mensaje
    % No se resuelve el problema
    disp('No hay cambio de signo')
else
    % SI hay cambio de signo
    % SI se resuelve el problema

    % 1) Se calcula el punto medio entre a y b
    p = (a + b)/2;

    % 2) Se calcula el error en ese punto medio del intervalo
    err = abs(f(p));

    % 3) Se inicia el proceso iterativo
    % El contador de itearciones "k" se fija en 1 antes de
    % entrar en ciclo "while"
```

```

k = 1;

% Si el error (variable "err") es mayor que una tolerancia,
% que se fija en la variable "tol", entonces entra
% en el ciclo "while"
tol = 1e-7;

while err > tol

    if f(a)*f(p)<0
        % SI entre a y p hay cambio de signo, el cero esta en
        % el nuevo intervalo que se redefine como [a,b=p]
        b = p;
    else
        % SI entre p y b hay cambio de signo, el cero esta en
        % el nuevo intervalo que se redefine como [a=p,b]
        a = p;
    end

    % Se calcula nuevamente punto medio "p" entre [a,b]
    % (redefinidos segun lo establecido mas arriba)
    p = (a + b)/2;

    % Se calcula el error para este nuevo punto "p"
    err = abs(f(p));

    % Se suma una iteracion al contador "k" antes de salir
    k = k + 1;

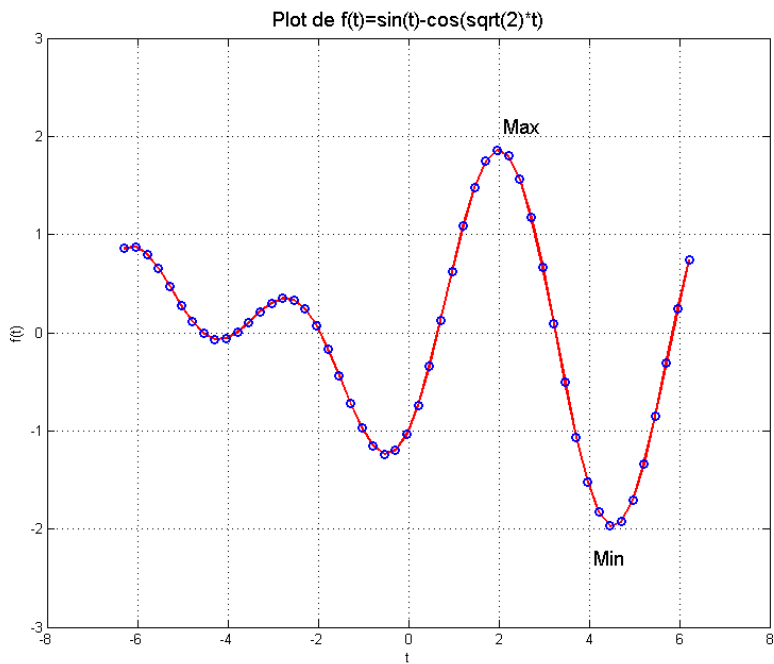
    % Si el error calculado es mayor que la tolerancia el
    % proceso se repite hasta que el error sea menor
end
end

% Aqui se muestran los resultados por pantalla
disp(['*****_RESULTADOS_*****'])
disp(['Raiz_o_cero_para_c=_ ' num2str(p)])
disp(['Numero_de_iteraciones_k=_ ' num2str(k)])
disp(['Evaluacion_de_la_funcion_f(c)= ' num2str(f(p))])
disp(['*****_-----_*****'])
%
%
```

El resultado por pantalla en Matlab es el siguiente

```
***** RESULTADOS *****  
Raiz o cero para c = 0.004504  
Numero de iteraciones k = 22  
Evaluacion de la funcion f(c)= -8.6351e-08  
***** ----- *****
```

Problema 3 (2 pts): Escriba el código Matlab que permite obtener la siguiente figura:



Considere el intervalo $[-2\pi, 2\pi]$, 50 puntos para la gráfica y ajuste los límites de la figura.

% Escriba aquí su código

Solución: La figura se obtiene con las siguientes instrucciones de Matlab

```
%  
%  
clc, clear all, close all  
  
% Vectores "t" y "y(t)" para el plot  
t = -2*pi:4/50*pi:2*pi;  
y = sin(t)-cos(sqrt(2)*t);  
  
% Funcion de Matlab para dibujar  
plot(t,y,'o-r','linewidth',2,...  
      'MarkerEdgeColor','b')  
  
% Configuracion del grafico  
axis tight  
grid on  
  
% Limites del grafico  
xlim([-8 8])  
ylim([-3 3])  
  
% Etiquetas en los ejes  
xlabel('t')  
ylabel('f(t)')  
  
% Titulo del grafico  
title('Plot de  $f(t)=\sin(t)-\cos(\sqrt{2}t)$ ', 'FontSize',14)  
  
% Texto sobre el grafico  
text(2.1,+2.1,'Max','FontSize',14)  
text(4.1,-2.3,'Min','FontSize',14)  
%  
%
```