**ALGORÍTMICA NUMÉRICA: Examen Computacional 3 15 Diciembre 2023**

***DURACIÓN: 80 Minutos***

Apellidos: JIMENEZ GONZALEZ

Nombre: MARCOS

**Instrucciones: ­­­­**

**Durante el examen no se permitirán accesos que no sean a Moodle.**

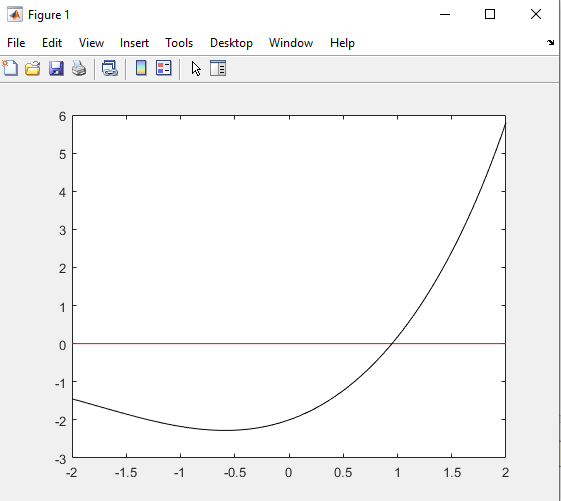
**Trabajar (Word y Matlab) en vuestra unidad USB o en el disco D:\\ para evitar pérdidas de información.**

En la Hoja de Respuestas hay que **incluir el código, resultados, gráficas y respuestas pedidas.** No se darán por válidos los resultados que no se deriven de la secuencia de sentencias incluidas en la solución de cada ejercicio. No se darán tampoco por válidos los códigos que no incluyan gráficas o los resultados pedidos.­­

La entrega de este ejercicio (doc, docx o pdf) se llevará a cabo a través del curso Moodle. No se aceptan entregas fuera de plazo o por correo electrónico.

**ES RESPONSABILIDAD DEL ALUMNO COMPROBAR QUE L­­­­­­­­­­­­­­­­­­----A ENTREGA SE HA HECHO CORRECTAMENTE ANTES DE ABANDONAR EL AULA.**

1. Código. Gráfica y respuestas.­­­­­­­­
2. xx = -2:0.01:2;
3. yy = exp(xx) - cos(xx)-2;
4. plot(xx,xx\*0,'r',xx,yy,'k');



Observamos que tiene una sola raíz

Un intervalo de longitud 1 donde existe la raíz puede ser [0.5:1.5]

f0=exp(0.5) - cos(0.5)-2;

f1=exp(1.5) - cos(1.5)-2;

f0\*f1

ans =

-2.9627

Debido a que el resultado es negativo, queda demostrado la existencia de por lo menos una raíz.

1. Código. Bisección y llamada a la función. Últimas 5 iteraciones con el formato solicitado.
2. s=biseccion(@fun,0.5,1.5,40)
3. %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
4. %%%%%%%%%%%%%%% FUNCIONES %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
5. function f = fun(x)
6. exp(x) - cos(x)-2;
7. end
8. function s=biseccion(fun,a,b,N) % Método de la bisección.
9. % Argumentos Entrada: fun puntero a la función; [a,b] intervalo donde está la solución
10. % N: número máximo de iteraciones
11. % Argumento Salida: s es la aproximación de la raiz
12. fa=fun(a);fb=fun(b); % Evalúo la función f en a y en b.
13. if fa\*fb > 0
14. fprintf('Método no aplicable en intervalo [a,b]\n')
15. end
16. for i=1:N
17. c=(a+b)/2
18. fc=fun(c); %evalúo la función en punto medio intervalo
19. if fa\*fc < 0
20. b=c; %fb=fc; (no es necesario)
21. else
22. a=c;
23. fa=fc;
24. end
25. end
26. s=a % pueder ser s=b o s=(a+b)/2,
27. fprintf('La raiz aproximada es %12.8f\n',s)
28. end
29. Código. Últimas 5 iteraciones con el formato solicitado. Gráfica y respuestas.

3)

clear,clc

x=zeros(1,25);

x0=0.5;

for k=1:25

x(k)=log(cos(x0)+2);

fprintf("Iteración: %2d, Valor calculado = %.12f\n",k,x(k))

x0=x(k)

end

s1 = x(end);

error = abs((s1-x));

semilogy(0:24,error,'bo:');

Iteración: 20, Valor calculado = 0.948814755545

Iteración: 21, Valor calculado = 0.948814755587

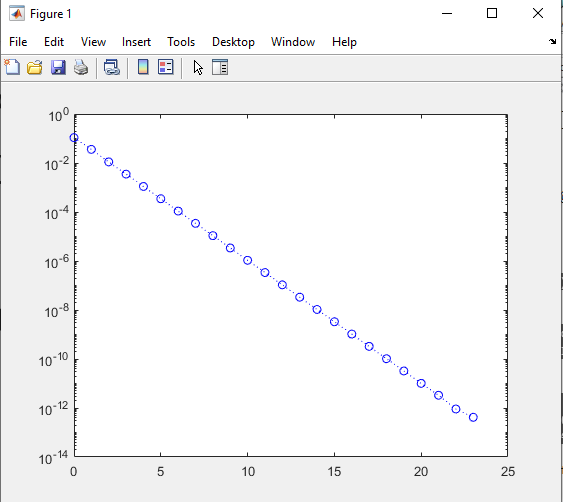
Iteración: 22, Valor calculado = 0.948814755574

Iteración: 23, Valor calculado = 0.948814755578

Iteración: 24, Valor calculado = 0.948814755577

Iteración: 25, Valor calculado = 0.948814755577

­­­



Tanto en la grafica como calculando con cif\_correct=floor(-log10(abs(s1-x)));

, se puede observar que se alcanzas los 8 decimales correctos en la iteracion 16

k = abs(error(2)) / abs(error(1));

0.332154330233484

Al comparar este método con el de la bisección observamos que este es mas lento, debido a que el de la bisección tiene una mayor velocidad de convergencia

1. Código. Iteraciones con el formato solicitado y respuestas.
2. error=100.0; % Control.
3. iteracion=1; % Contador iteraciones, iniciar
4. x0=0.5; % Valor inicial
5. while (error>1e-12) % Criterio parada
6. x1=x0-(exp(x0)-cos(x0)-2)./(exp(x0)+sin(x0)); % Método iterativo
7. error=abs(x1-x0); %Estimación error
8. ncif = floor(-log10(error));
9. fprintf('Iteracion %d Valor calculado %.12f Error %e Cifras %d\n', iteracion,x1,error,ncif) %Imprimir soluciones aprox,...
10. iteracion=iteracion+1;
11. x0=x1;
12. end

Iteracion 1 Valor calculado 1.077432574584 Error 5.774326e-01 Cifras 0

Iteracion 2 Valor calculado 0.956020014220 Error 1.214126e-01 Cifras 0

Iteracion 3 Valor calculado 0.948838857460 Error 7.181157e-03 Cifras 2

Iteracion 4 Valor calculado 0.948814755848 Error 2.410161e-05 Cifras 4

Iteracion 5 Valor calculado 0.948814755577 Error 2.707651e-10 Cifras 9

Iteracion 6 Valor calculado 0.948814755577 Error 0.000000e+00 Cifras Inf

La primera convergencia es cuadrática ya que entre cada par de

% iteraciones se duplica el número de cifras significativas de precisión

% correctas; , además se cumple e(n+1) ~= K e(n)^2