

Práctica 2. Raíces de Funciones.

Todos los scripts y funciones se copiarán en un Word de resultados, atendiendo a la Sección y número del ejercicio en el que nos encontremos.

Para aquellos scripts o funciones que sean llamados desde consola de comandos se copiará, además, la llamada realizada, así como el resultado generado, tanto numérico como figuras.

INTRODUCCIÓN

MATLAB proporciona la función $x=fzero(nombre_funcion, x0, options)$ para obtener la raíz de una función. Si la función está definida en un fichero '.m' (en cuyo caso debe tener la estructura $y=nombre_funcion(x)$) el primer argumento de **fzero** es una cadena con el nombre de la función (por ejemplo, el nombre de la función entre comillas simples). Alternativamente, la función puede ser definida usando el comando **@**. Los otros argumentos de entrada a **fzero** son la aproximación inicial $x0$, y un último argumento que permite definir diferentes opciones (utiliza **help** si quieres conocerlas en profundidad), en la sección 2 del archivo "IntroducciónPráctica2.pdf" encontrarás la descripción de las más interesantes. El último argumento se puede omitir.

La función **fzero** emplea el método de Brent, que combina la interpolación cuadrática inversa con la bisección.

SECCIÓN 1. Cálculo de raíces con Matlab.

1. Programa una función en el fichero *mifuncion.m* que evalúe la siguiente expresión matemática $y=e^{\sin(x)} - 2*\cos(x)$. Dibuja dicha función en el intervalo $[0,10]$. ¿Cuántas raíces tiene esta función? Calcular todas las raíces de la función en el intervalo $[0, 10]$ llamando a la función **fzero** cuantas veces sean necesarias, usando distintos valores de $x0$ cada vez. Detalla todas las instrucciones y las raíces que has obtenido.

2. Escribe una función que calcule una raíz de una función cualquiera usando el método de bisección. Para ello escribe una función, $[x,it]=bisecc(funcion,a,b,tol,maxiter)$, que admita como parámetros de entrada cualquier función tipo *fun.m* y el intervalo (a,b) donde se ha de buscar la raíz, un valor máximo para el error cometido *tol* y un número máximo de iteraciones a realizar *maxiter*. La función devolverá la raíz x y el número de iteraciones *it* del método de bisección. Si se supera el número máximo de iteraciones permitidas el programa deberá devolver para la variable x la última solución encontrada. En este caso, además, deberá mostrar un mensaje por pantalla indicando que el algoritmo no ha convergido y el valor del error del error para la última solución encontrada.

Usa tu función **bisecc** para calcular las raíces de la función $f(x)=x-\sin(x)-1$ y comprueba el resultado obtenido comparando tu raíz con la obtenida usando **fzero**.

3. Programa una función $[x,it] = interpol(g,tol,x0,x1, maxiter)$ que obtenga numéricamente una raíz de la función matemática $g(x)$ usando el método iterativo de interpolación lineal con una tolerancia *tol*, siendo $x0$ y $x1$ los valores iniciales (próximos a la raíz exacta x) con los que comienza el método y *maxiter* el número máximo de iteraciones. La ejecución de la función debe incluir la aparición en pantalla de los pasos o iteraciones sucesivas, mostrando los valores de: nº de iteración, solución parcial aproximada x y el valor de $fun(x)$. En este caso, se definirá la función de entrada g usando el comando **@** en vez de mediante un fichero '.m'. Usa tu programa para calcular las raíces de la función $f(x)$ del ejercicio anterior.

4. El cálculo de la raíz cuadrada de 3 se puede determinar calculando la raíz positiva de la ecuación $x^2=3$; si la reescribimos tenemos la expresión $x=(3+x)/(1+x)$. Programa una función que determine la raíz cuadrada de 3 por el **método del punto fijo** con la expresión que se indica, considerando como valor inicial 1. ¿Cuántas iteraciones se han realizado para llegar a tener trece decimales exactos de precisión?
5. Programa una función `[x, it]=newton(fun, der, x0, tol, maxiter)` para calcular la raíz de una función cualquiera usando el método de Newton. En este caso, la función **newton** recibirá como argumentos de entrada tanto la función cuya raíz se desea calcular (*fun*) como su función derivada (*der*), que calcularás analíticamente previamente a la ejecución, *x0* la aproximación inicial y *tol* la tolerancia. Las funciones *fun* y *der* serán definidas en ficheros *.m*. Aplícalo a la función $f(x)=x-\sin(x)-1$. Escribe los resultados en un fichero de texto llamado `newton.txt`. El valor inicial para la raíz será igual a 0.1 y una tolerancia de 10^{-10} .