

Práctica 3 – Obtención de raíces por acotación

Ejercicio 1

Acotar visualmente la raíz de las funciones de trabajo $f_1(x)=x^2-e^x$ y $f_2(x)=2x^3-e^x$ mediante Gnuplot. Para ello se pide:

1. Crear una figura mediante el programa Gnuplot con la representación conjunta de las dos funciones de trabajo tal que:
 - A) Los ejes de la figura estén acotados en los intervalos $x = [-5, 5]$ y $f(x) = [-20, 20]$.
 - B) En la figura aparezca: a) ejes de abscisas y ordenadas, b) título “Funciones trabajo Práctica 3”, c) etiquetas en los ejes de abscisas “x” y ordenadas “f(x)” y d) leyendas “Función Trabajo 1”, “Función Trabajo 2”.
 - C) La figura quede guardada en un archivo “*figura.png*”.
2. Construir un programa Fortran que:
 - A) Escriba en un fichero “*pinta.gnpl*” las órdenes de Gnuplot usadas en el apartado 1.
 - B) Llame a Gnuplot para ejecutar las órdenes almacenadas en el fichero “*pinta.gnpl*”

Ejercicio 2

Calcular la raíz de las funciones de trabajo $f_1(x)=x^2-e^x$ y $f_2(x)=2x^3-e^x$ mediante el método de la bisección. Para ello se pide:

1. Construir un programa (principal) Fortran que:
 - A) Pida por pantalla: a) los valores extremos del intervalo inicial de acotación para la raíz de cada función y verifique que son válidos (si no lo son pedir unos nuevos), b) la precisión (error en x) y error de cero (error en y) deseada de las raíces, y c) el número máximo de iteraciones que se permite al método.
 - B) Use una función vectorial (“*function*”) para que aplique el método de la bisección a $f_1(x)$.
 - C) Use la misma función vectorial para que aplique el método a $f_2(x)$.
 - D) Proporcione por pantalla el valor de las raíces (números reales) y el número de iteraciones usadas por el método (números enteros) para cada función de trabajo.
 - E) Informe si se ha superado el número máximo de iteraciones (el método no converge), dando el resultado para la última iteración realizada. Además, indique que se debe: a) reducir el intervalo inicial de acotación y/o, b) mejorar la precisión y/o, c) reducir el error de cero, y/o, d) aumentar el número máximo de iteraciones.
2. Construir un programa (método) Fortran que codifique mediante una función vectorial (“*function*”) el método de la bisección. Esta función vectorial debe tener como argumento de entrada una función escalar (“*real function*”) que calcule el valor de la función de trabajo en el punto x que se desee.
3. Construir un programa (trabajo) Fortran que codifique mediante funciones escalares (“*real function*”) cada función de trabajo.

Ejercicio 3

Repetir el ejercicio 2, usando el método de la falsa posición.

Ejercicio 4

Comparar conjuntamente los dos métodos de cálculo de raíces realizados en esta práctica (bisección y falsa posición). Se pide hacerlo de dos maneras distintas:

Modo 1:

- A) Construir un programa (principal) Fortran que: a) Pida por pantalla los valores extremos del intervalo inicial de acotación verificando que son válidos, la precisión (error en x) y error de cero (error en y), y el número máximo de iteraciones. b) use dos veces la función vectorial con el método de la bisección, y otras dos veces la función vectorial con el método de la falsa posición, c) proporcione por pantalla las raíces obtenidas por cada método para cada función de trabajo, así como el número de iteraciones empleadas, y d) informe si se ha superado el número máximo de iteraciones.
- B) Construir un programa (método1) Fortran que codifique mediante una función vectorial el método de la bisección (ejercicio 2).
- C) Construir un programa (método2) Fortran que codifique mediante una función vectorial el método de la falsa posición (ejercicio 3).
- D) Construir un programa (trabajo) Fortran que codifique mediante funciones escalares cada función de trabajo.

Nota: La compilación puede hacerse en cualquier orden.

Modo 2:

- A) Construir un programa (principal) Fortran que: a) use a un módulo en el que se encuentran codificadas las dos funciones vectoriales con ambos métodos, b) pida por pantalla los valores extremos del intervalo inicial de acotación verificando que son válidos, la precisión (error en x) y error de cero (error en y), y el número máximo de iteraciones, c) use dos veces la función vectorial con el método de la bisección, y otras dos veces la función vectorial con el método de la falsa posición, d) proporcione por pantalla las raíces obtenidas por cada método para cada función de trabajo, así como el número de iteraciones empleadas, y e) informe si se ha superado el número máximo de iteraciones.
- B) Construir un programa (módulo) Fortran que construya un módulo el cual codifica mediante dos funciones vectoriales (una debajo de otra) el método de la bisección (ejercicio 2) y el método de la falsa posición (ejercicio 3).
- C) Construir un programa (trabajo) Fortran que codifique mediante funciones escalares cada función de trabajo.

Nota: La compilación debe hacerse con el programa módulo en primer lugar, y el resto en cualquier orden:

gfortran modulo.f90 trabajo.f90 principal.f90 -o salida.exe