Métodos Numéricos Avanzados

PRÁCTICAS

Tema 1. RESOLUCIÓN NUMÉRICA DE SISTEMAS NO LINEALES

Cuarto Curso de Grado en Física
(Primer cuatrimestre)

PROF.: JOSÉ ANTONIO SÁNCHEZ PELEGRÍN

Sistemas no lineales (II)

Curso 2023 – 2024

Universidad de Córdoba
Dpto. Informática y Análisis Numérico

Objetivo y ejecución

- En esta práctica se programarán el método de Broyden y el método de continuación para la resolución de sistemas de ecuaciones no lineales.
- Cada uno de los ficheros creados durante la práctica se subirá a la página Moodle de la asignatura al enlace habilitado para ello.

Práctica 2.1. Escribe una función de MATLAB, llamada `SNLBroyden`, que resuelva un sistema no lineal algebraico por el método de Broyden. El encabezamiento será:

```
1
2  function [x,iter,norma,salida]=...
3            SNLbroyden(fun,jacobiana,x0,itermax,tol,detalle)
4  %-------------------------------------------------------------
5  % Metodo de Broyden para sistemas de ecuaciones
6  %-------------------------------------------------------------
7  % Argumentos de entrada
8  %-------------------------------------------------------------
9  % fun       = función vectorial que define el sistema (f(x)=0)
10 % jacobiana = jacobiana de la función f
11 % x0        = vector inicial
12 % itermax   = numero máximo de iteraciones admitidas
13 % tol       = tolerancia de parada
14 % detalle    = 0 No se muestran las iteraciones
15 %             = 1 Se muestran las iteraciones
16 %--------------------
17 % Se han de tener las funciones:
18 % fun.m funcion y=fun(x) donde se define el sistema
19 % jacobiana.m funcion J=jacobiana(x) donde se define la matriz
20 % jacobiana del sistema, o una aproximación suya
21 %
22 %-------------------------------------------------------------
23 % Argumentos de salida:
24 %-------------------------------------------------------------
25 % x      = solución aproximada
26 % iter   = iteraciones realizadas hasta alcanzar la tolerancia
27 % norma  = norma de la diferencia de las dos últimas iteraciones
28 % salida = 1 si se alcanza la tolerancia
29 %        = 0 si no se alcanza la tolerancia en las iteraciones dadas.
30 %-------------------------------------------------------------
31 %
```

Práctica 2.2. Escribe un programa principal que use la función `SNLBroyden` para aproximar la solución del sistema no lineal:

$$\begin{cases} x^3 - 10x + y - z + 3 = 0, \\ y^3 + 10y - 2x - 2z - 5 = 0, \\ x + y - 10z + 2\operatorname{sen} z + 5 = 0. \end{cases}$$

Toma como punto inicial $\mathbf{x}^0 = (1,1,1)$ y como tolerancia $\varepsilon = 10^{-2}$. Resuélvelo después de forma simbólica, en el mismo programa principal, comprobando el resultado obtenido.

Práctica 2.3. Escribe una función de MATLAB, llamada `continuacion`, que resuelva un sistema no lineal algebraico por el método de continuación. El encabezamiento será:

```
function [x,normres]=continuacion(fun,jacobiana,x0,M)

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% Metodo de continuación para sistemas de ecuaciones
%-------------------------------------------------------------
% Argumentos de entrada
%-------------------------------------------------------------
% fun         = función que define el sistema (f(x)=0)
% jacobiana   = matriz jacobiana de fun
% x0          = vector inicial
% M           = Número de pasos a usar en Runge-Kutta
%-------------------------------------------------------------
% Argumentos de salida:
%-------------------------------------------------------------
% x       = solución aproximada
% normres= norma infinito del residuo
%-------------------------------------------------------------
%
```

Práctica 2.4. Escribe un programa principal que use la función `continuacion` para aproximar, de nuevo, la solución del sistema no lineal del ejercicio anterior.

(a) Parte del mismo punto inicial $\mathbf{x}^0 = (1,1,1)$ y primero resuélvelo tomando 4 pasos en el método de Runge-Kutta y luego 40 pasos. Muestra las salidas con 10 cifras decimales.

(b) Calcula el residuo en cada caso y muéstralos también con 10 cifras decimales.