Métodos Numéricos Avanzados
PRÁCTICAS

Tema 2. INTRODUCCIÓN A LA OPTIMIZACIÓN NUMÉRICA

Cuarto Curso de Grado en Física
(Primer cuatrimestre)

PROF.: JOSÉ ANTONIO SÁNCHEZ PELEGRÍN

Problemas de Optimización (I)

Curso 2023 – 2024

Universidad de Córdoba
Dpto. Informática y Análisis Numérico

Práctica 3.1. Escribe una función de MATLAB, llamada `aurea`, que resuelva un problema de optimización en una variable, utilizando el método de la sección áurea. El encabezamiento será:

```

function [optimo,valor,nit,conver]=aurea(fun,a,b,maxiter,toler)
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%% Aproxima el óptimo (mínimo) de una función unimodal en un intervalo
%  usando el método de la sección áurea
%  Datos
%  -----
%  fun    = función para buscarle el mínimo
% [a,b]   = intervalo inicial
% maxiter    = numero máximo de iteraciones admitidas
% toler      = tolerancia de parada
%--------------------
% Se ha de tener la función:
% fun.m función y=fun(x) donde se define la función a optimizar
%---------------------------------------------------------------------
% Argumentos de salida:
%---------------------------------------------------------------------
% optimo = óptimo aproximado
% valor  = valor de la función en el óptimo aproximado
% nit    = iteraciones realizadas hasta alcanzar la tolerancia
% conver = 1 si se alcanza la tolerancia
%        = 0 si no se alcanza la tolerancia en las iteraciones dadas.
%---------------------------------------------------------------------
```

Práctica 3.2. Utiliza el método de la sección áurea para obtener el mínimo y el máximo de la función

$$h(x) = \frac{1}{(x-0.3)^2+0.01} + \frac{1}{(x-0.9)^2+0.04},$$

en el intervalo $[0.5, 1]$.

Comprueba los resultado obtenidos usando la orden `fminbnd` de MATLAB.

Práctica 3.3. Escribe una función de MATLAB, llamada `gradientec`, que resuelva el sistema lineal **Ax** = **b** de matriz simétrica definida positiva, utilizando el método del gradiente conjugado. El encabezamiento y sus primeras líneas serán:

```
    function [sol,iteraciones,resrel,cod]=gradientec(A,b,toler,maxiter,detalle)
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%% Aplica el método de gradiente conjugado para
%            encontrar la solución de Ax=b
%  Datos
%  -----
%      A = matriz del sistema (simétrica, definida positiva)
%      b = segundo miembro del sistema
%      toler = tolerancia en el residuo para terminar de iterar
%      maxiter   = numero máximo de iteraciones admitidas
%      detalle   = 0 No se muestran las iteraciones
%                = 1 Se muestran las iteraciones
%--------------------------------------------------------------------
% Argumentos de salida:
%--------------------------------------------------------------------
% sol         = solución del sistema de ecuaciones con (residuo < toler)
% iteraciones = iteraciones realizadas hasta alcanzar la tolerancia
% resrel      = residuo relativo norm(b-A*sol)/norm(b)
% cod         = 1 si se alcanza la tolerancia
%             = 0 si no se alcanza la tolerancia en las iteraciones dadas.
%--------------------------------------------------------------------
if nargin < 3
    toler=eps; maxiter=1000; detalle=1;
elseif nargin < 4
    maxiter=1000; detalle=1;
elseif nargin < 5
    detalle=1;
end
```

Para comprobar, previamente si se trata de una matriz simétrica y definida positiva se puede utilizar la siguiente función:

```
function def_positiva(A)
% Función que permite comprobar si una matriz A simétrica
% es definida positiva viendo si todos sus autovalores son positivos
%-----------------------------------------------------------------------
% Comprobación de que la matriz A sea simétrica
if A'==A
    disp('La matriz es simétrica')
    % Se calculan los autovectores (v) y los autovalores (lambda) de A
    [v,lambda]=eig(A);
    if  all(diag(lambda)>0)
        disp('La matriz es definida positiva')
    else
        disp('La matriz no es definida positiva')
    end
else
    disp('La matriz  no es simétrica')
end
```

Práctica 3.4. Considera el sistema lineal

$$\begin{pmatrix} 10 & 1 & 2 & 3 & 4 \\ 1 & 9 & -1 & 2 & -3 \\ 2 & -1 & 7 & 3 & -5 \\ 3 & 2 & 3 & 12 & -1 \\ 4 & -3 & -5 & -1 & 15 \end{pmatrix} \begin{pmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \\ x_5 \end{pmatrix} = \begin{pmatrix} 12 \\ -27 \\ 14 \\ -17 \\ 12 \end{pmatrix}.$$

(a) Comprueba que la matriz A es simétrica definida positiva usando la función `def_positiva`.

(b) Calcula la solución del sistema usando la división inversa de MATLAB.

(c) Resuelve ahora el sistema utilizando el método de gradiente conjugado que has implementado en el ejercicio anterior, y compara los resultados. Muestra la solución obtenida, el número de iteraciones que ha necesitado y el residuo relativo.

(d) Utiliza la función `pcg` (*preconditioned conjugate gradient*), implementada en MATLAB. Su sintaxis es:

[x,flag,relres,iter] = `pcg`(A,b,tol,maxiter)

- `flag` = 0 denota que el criterio de parada ha sido alcanzado;
- `relres`= norm(b-A*x)/norm(b);
- `iter` contiene la cantidad de iteraciones efectuadas.

Muestra aquí, también, todos los datos obtenidos.