

## Práctica 6: Resolución aproximada dun sistema non linear polo método iterativo de Newton discretizado

1. Nesta práctica debes programar, en **MATLAB**, a resolución do sistema non linear:

$$F(x) = \theta,$$

para  $F : \mathbb{R}^n \longrightarrow \mathbb{R}^n$ , mediante o **método de Newton discretizado**. Entón, debes:

- a) Escribir un ficheiro, `newton_dis_datos.m`, cos datos do método:

```
niter=... ;           % número máximo de iteracións
funcion='f' ;         % función que define o sistema non linear
x0=[...;...; ... ] ;  % iterante inicial
n=... ;               % orde do sistema
h=... ;               % paso de discretización da matriz xacobiana
tol=... ;             % valor de tolerancia do test de parada
```

- b) Escribir unha `function`, nun ficheiro `f.m`, para a función  $F$  que define o sistema:

```
function z=f(x)
z=[...;...; ... ] ;   % expresión da función non linear F(x)
return
```

- c) Programar o método, tomando como modelo o seguinte script:

```
% Metodo de Newton discretizado para un sistema non linear
clc
clear all
newton_dis_datos

for it=1:niter

    fx=feval(funcion,x0);

% Matriz xacobiana discretizada
    for j=1:n
        x0(j)=x0(j)+h;
        fxh=feval(funcion,x0);
        x0(j)=x0(j)-h;
        df(:,j)=(fxh-fx)/h;
    end

% Test sobre a matriz xacobiana discretizada
    detdf=det(df);
    if(abs(detdf)<1.e-12)
        disp(['det(df)= ',num2str(detdf)])
        disp('matriz xacobiana discretizada singular')
        return
    end

% Resolucion do sistema
    dx=df\fx;
    x1=x0-dx;
```

```

% Test de parada
error=norm(dx);
if(error<=tol) % test error absoluto
% if(error<=tol*norm(x0)) % test error relativo
% if(error<=tol*(norm(x0)+1)) % test error absoluto/relativo
disp('Converxencia alcanzada')
disp(['A solucion e: ',num2str(x1')])
disp(['Obtivose na iteracion: ',num2str(it)])
residuo=norm(feval(funcion,x1));
disp(['0 residuo e: ',num2str(residuo)])
return
else
x0=x1;
end
end

disp('Superouse o numero maximo de iteracions sen converxencia')
disp(['0 ultimo iterante calculado e: ',num2str(x1')])
residuo=norm(feval(funcion,x1));
disp(['0 residuo e: ',num2str(residuo)])

```

d) Valida o método programado cos sistemas:

$$\begin{cases} x_1^2 - x_2 = 0 \\ -x_1 + x_2^2 = 0 \end{cases}$$

con solucións:  $x_1 = x_2 = 0$ ;  $x_1 = x_2 = 1$ .

$$\begin{cases} x_1^2 - x_2 + 0.25 = 0 \\ -x_1 + x_2^2 + 0.25 = 0 \end{cases}$$

con solución:  $x_1 = x_2 = 0.5$ .

$$\begin{cases} x_1^2 + x_2^2 - 2 = 0 \\ -x_1 + 2x_2^2 - 1 = 0 \end{cases}$$

con solución:  $x_1 = 1$ ,  $x_2 = \pm 1$ .