2.1 Resolución de un sistema de ecuaciones algebraicas

•Se plantea resolver el siguiente problema: sistema de n ecuaciones algebraicas con n incógnitas, que puede incluir ecuaciones no lineales, con la siguiente forma general.

$f_1(x_1,\ldots,x_n)=0$

$f_2(x_1,\ldots,x_n)=0$

.

.

.

$f_n(x_1,\ldots,x_n)=0$

Encontrar el vector $(x_1,\ldots\ldots,x_n)$ que hace que todas las ecuaciones sean igual a cero. Matlab tiene una sentencia para resolver esto: *fsolve.m*. Utilizamos una forma mejorada de este que es **csolve.m**

2.1 Resolución de un sistema de ecuaciones algebraicas

- Sintáxis:

[x,rc]=csolve('nombrefunción',x0,'gradfun',crit,itmax,p1,p2,…)

Nombre función = es un fichero función que calcula los valores de f_i.

•Sintáxis de 'nombrefunción':

```
function y=nombrefuncion(x)
y(1,:)=f1(x(1,:), x(2,:),…,x(n,:));

y(n,:)=fn(x(1,:),…,x(n,:));
```

La variable x es una matriz con n filas (n = nº de incógnitas) y con un número variable de columnas. Cada columna es una iteración que hace el programa.

$$
\begin{pmatrix}
x_{10} & x_{11} & x_{12} & \text{------} & x_{1n} \\
x_{20} & x_{21} & x_{22} & \text{------} & x_{2n} \\
x_{30} & x_{31} & x_{32} & \text{------} & x_{3n} \\
x_{40} & x_{41} & x_{42} & \text{------} & x_{4n} \\
\vdots & \vdots & \vdots & & \vdots \\
x_{n0} & x_{n1} & x_{n2} & \text{------} & x_{nn}
\end{pmatrix}
\qquad
\begin{pmatrix}
y_{10}\,(x_{10}\ x_{20}\text{-----}\ x_{n0}) & y_{11}\,(x_{11}\ x_{21}\text{-----}\ x_{n1}) & y_{1n}\,(x_{1n}\ x_{2n}\text{-----}\ x_{nn}) \\
y_{20}\,(x_{10}\ x_{20}\text{-----}\ x_{n0}) & y_{21}\,(x_{11}\ x_{21}\text{-----}\ x_{n0}) & y_{2n}\,(x_{1n}\ x_{2n}\text{-----}\ x_{nn}) \\
y_{30}\,(x_{10}\ x_{20}\text{-----}\ x_{n0}) & y_{31}\,(x_{11}\ x_{21}\text{-----}\ x_{n1}) & y_{3n}\,(x_{1n}\ x_{2n}\text{-----}\ x_{nn}) \\
y_{40}\,(x_{10}\ x_{20}\text{-----}\ x_{n0}) & y_{41}\,(x_{11}\ x_{21}\text{-----}\ x_{n1}) & y_{4n}\,(x_{1n}\ x_{2n}\text{-----}\ x_{nn}) \\
\vdots & \vdots & \vdots \\
y_{n0}\,(x_{10}\ x_{20}\text{-----}\ x_{n0}) & y_{n1}\,(x_{11}\ x_{21}\text{-----}\ x_{n1}) & y_{nn}\,(x_{1n}\ x_{2n}\text{-----}\ x_{nn})
\end{pmatrix}
$$

Cuando se logre que sustituyendo la columna de valores de x (desde x_1 hasta x_n) la columna de valores de las funciones se aproximen todas a cero, habré llegado a la solución.

2.1 Resolución de un sistema de ecuaciones algebraicas

[x,rc]=csolve('nombrefunción',x0,'gradfun',crit,itmax,p1,p2,…)

- x0: es un vector columna con la primera estimación de x $(x_{10},\ldots,x_{n0})$

[x,rc]=csolve('nombrefunción',x0,'gradfun',crit,itmax,p1,p2,…)

- 'gradfun': función que calcula el gradiente de 'nombrefunción'. Sirve para acelerar la llegada del comando a la solución del sistema de ecuaciones. No es absolutamente necesario emplearlo (en vez de escribir grandfun, se escribe []).

[x,rc]=csolve('nombrefunción',x0,'gradfun',crit,itmax,p1,p2,…)

- crit: número pequeño positivo que cumple $\sum_{i=1}^{i=n} |y_n| < \text{crit}$ en la solución (y de esta manera se determina que se ha llegado a la solución).

[x,rc]=csolve('nombrefunción',x0,'gradfun',crit,itmax,p1,p2,…)

- itmax: el número máximo de iteraciones permitidas al comando.

[x,rc]=csolve('nombrefunción',x0,'gradfun',crit,itmax,p1,p2,…)

- p1,p2,…: parámetros adicionales que se introducen en esta posición para calcular $y_1,\dots,y_n$. (Se pueden introducir con una sentencia global y no sería necesario introducirlos aquí).

2.1 Resolución de un sistema de ecuaciones algebraicas

[x,rc]=csolve('nombrefunción',x0,'gradfun',crit,itmax,p1,p2,…)

•rc: indicador del grado de acercamiento a una solución verdadera.

rc = 0, se ha llegado a una solución verdadera

rc = 4, se ha llegado al máximo número de interacciones.

•x: vector columna con la solución del sistema de ecuaciones si rc=0

El cálculo de la matriz [y(1,:); y(2,:);……; y(n,:)] es muy complicado cuando las ecuaciones f_1,………,f_n son complejas……

DIFICULTAD EN LAS OPERACIONES ELEMENTO A ELEMENTO.

El cálculo de la matriz [y(1,:); y(2,:);…….; y(n,:)] es muy complicado cuando las ecuaciones $f_1,\ldots\ldots\ldots,f_n$ son complejas…….

DIFICULTAD EN LAS OPERACIONES ELEMENTO A ELEMENTO.

$$\begin{pmatrix} x_{10} & x_{11} & x_{12} & \text{- - - - - -} & x_{1n} \\ x_{20} & x_{21} & x_{22} & \text{- - - - - -} & x_{2n} \\ x_{30} & x_{31} & x_{32} & \text{- - - - - -} & x_{3n} \\ x_{40} & x_{41} & x_{42} & \text{- - - - - -} & x_{4n} \\ , & , & , & & , \\ , & , & , & & , \\ , & , & , & & , \\ , & , & , & & , \\ x_{n0} & x_{n1} & x_{n2} & \text{- - - - - -} & x_{nn} \end{pmatrix} \qquad \begin{pmatrix} y_{10}(x_{10}\ x_{20}\text{-----}\ x_{n0}) & y_{11}(x_{11}\ x_{21}\text{-----}\ x_{n1}) & y_{1n}(x_{1n}\ x_{2n}\text{-----}\ x_{nn}) \\ y_{20}(x_{10}\ x_{20}\text{-----}\ x_{n0}) & y_{21}(x_{11}\ x_{21}\text{-----}\ x_{n0}) & y_{2n}(x_{1n}\ x_{2n}\text{-----}\ x_{nn}) \\ y_{30}(x_{10}\ x_{20}\text{-----}\ x_{n0}) & y_{31}(x_{11}\ x_{21}\text{-----}\ x_{n1}) & y_{3n}(x_{1n}\ x_{2n}\text{-----}\ x_{nn}) \\ y_{40}(x_{10}\ x_{20}\text{-----}\ x_{n0}) & y_{41}(x_{11}\ x_{21}\text{-----}\ x_{n1}) & y_{4n}(x_{1n}\ x_{2n}\text{-----}\ x_{nn}) \\ , & , & , \\ , & , & , \\ , & , & , \\ , & , & , \\ y_{n0}(x_{10}\ x_{20}\text{-----}\ x_{n0}) & y_{n1}(x_{11}\ x_{21}\text{-----}\ x_{n1}) & y_{nn}(x_{1n}\ x_{2n}\text{-----}\ x_{nn}) \end{pmatrix}$$

2.1 Resolución de un sistema de ecuaciones algebraicas

- Para facilitar el cálculo de las funciones $f_1,\ldots,f_n$ a partir de la matriz x(1,:),…,x(n,:), se extrae cada columna $x_1,\ldots x_n$ por separado, y se calcula:

$$
\begin{gathered}
y_1=f_1(x_1,\ldots,x_n)\\
y_2=f_2(x_1,\ldots,x_n)\\
.\\
.\\
.\\
y_n=f_n(x_1,\ldots,x_n)
\end{gathered}
$$

- Construimos la función ‘nombrefunción’ :

```
function y=nombrefuncion(x)
filas=size(x,1) %número de filas de la matriz x
columnas=size(x,2) %número de columnas de la matriz x
```

El comando size(matriz, número) calcula el número de filas de la matriz cuando el número es 1, y el número de columnas cuando es 2.

2.1 Resolución de un sistema de ecuaciones algebraicas

- Se busca calcular una matriz y a partir de una matriz x con la siguiente forma:

$$\begin{bmatrix} x_{1,0} & \cdot & \cdot & \cdot & x_{1,columnas} \\ \cdot & & & & \cdot \\ \cdot & & & & \cdot \\ \cdot & & & & \cdot \\ x_{filas,0} & \cdot & \cdot & \cdot & x_{filas,columnas} \end{bmatrix} \Rightarrow \begin{bmatrix} y_{1,0} & \cdot & \cdot & \cdot & y_{1,columnas} \\ \cdot & & & & \cdot \\ \cdot & & & & \cdot \\ \cdot & & & & \cdot \\ y_{filas,0} & \cdot & \cdot & \cdot & y_{filas,columnas} \end{bmatrix}$$

- El siguiente paso es extraer una por una cada columna de la matriz x:

```
for j=1:columnas
   for i=1:filas
          v(i) =x(i,j);
   end
```

2.1 Resolución de un sistema de ecuaciones algebraicas

- $(v_1,\ldots,v_n)$ contiene $(x_1,..,x_n)$ para calcular $(y_1,\ldots,y_n)$'

Con este ciclo, el vector $(v_1,\ldots,v_n)$ contiene el conjunto de valores con el que hay que calcular $(y_1,\ldots,y_n)$.

```
y(1,j)=f1(v(1),…,v(n));
.
.
.
y(n,j)=fn(v(1),…,v(n));

%Se ha calculado la columna j de la matriz deseada

end %vuelve a calcular la siguiente columna de y
```

principal.m

```
n=lenght(x);  %número de ecuaciones (o de incógnitas)
x0=ones(1,n) %Suponemos x(1)=1, x(2)=1, …….x(n)=1
x0=x0'  %Necesario para csolve

[x,rc]=csolve('nombrefuncion',x0,[ ],1e-7,10000)
x   %vector solución
rc
```

funcion.m

```
function y=nombrefuncion(x)
filas=size(x,1);
columnas=size(x,2);
for j=1:columnas
    for i=1:filas
    v(i)=x(i,j);
    end
y(1,:)=f1(x(1,:), x(2,:),…,x(n,:));
.
.
y(n,:)=fn(x(1,:),…,x(n,:));
end
```

LAS FUNCIONES DE $f(x_1, x_2 \ldots x_n)$

2.1 Resolución de un sistema de ecuaciones algebraicas

- Ejemplo: resolver el siguiente sistema de ecuaciones:

$$3 + x_1^2 x_2 + x_2 - 7x_3 = 0$$

$$x_1 - x_3^3 = 0$$

$$x_2 - x_3 = 0$$

funcion.m

```
function y=funcion(x)
filas=size(x,1);
columnas=size(x,2);
for j=1:columnas
    for i=1:filas
    v(i)=x(i,j);
    end
y(1,j)=3+v(1)^2*v(2)+v(2)-7*v(3);
y(2,j)=v(1)-v(3)^3;
y(3,j)=v(2)-v(3);
end
```

2.1 Resolución de un sistema de ecuaciones algebraicas

principal.m

```
n=3;  %número de ecuaciones (o de incógnitas)
x0=ones(1,n) %Suponemos x(1)=1, x(2)=1, x(3)=1
x0=x0'  %Necesario para csolve

[x,rc]=csolve('funcion',x0,[ ],1e-7,10000)
x   %vector solución
rc
```

PROGRAMA

```
n=3
x0=ones(1,n)
x0=x0'
[x,rc]=csolve('funcionparacsolve',x0,[],1e-7,10000)
```

FUNCION
csolve

FUNCION

```
function y=funcionparacsolve(x)
filas=size(x,1)
columnas=size(x,2)
for j=1:columnas
    for i=1:filas
        v(i)=x(i,j)
    end
    y(1,j)=3+v(1)^2*v(2)+v(2)-7*v(3);
    y(2,j)=v(1)-v(3)^3;
    y(3,j)=v(2)-v(3);
end
```

INFORMATICA APLICADA A LA INGENIERÍA QUIMICA: 2ª parte MATLAB

PROGRAMA

```
n=3
x0=ones(1,n)
x0=x0'
[x,rc]=csolve('funcionparacsolve',x0,[],1e-7,10000)
```

FUNCION
csolve

FUNCION

```
function y=funcionparacsolve(x)
filas=size(x,1)
columnas=size(x,2)
for j=1:columnas
    for i=1:filas
        v(i)=x(i,j)
    end
    y(1,j)=3+v(1)^2*v(2)+v(2)-7*v(3);
    y(2,j)=v(1)-v(3)^3;
    y(3,j)=v(2)-v(3);
end
```