

INFORMATICA APLICADA A LA INGENIERÍA QUIMICA: 2ª parte MATLAB

2.3 Resolución de sistemas de ecuaciones diferenciales ordinarias aplicadas a sistemas de interés en I.Q.

- Un ingeniero químico debe estar interesado en cómo influyen las variables de operación en un determinado proceso:

Condiciones óptimas de reacción

Diseño de equipos

POR LA COMPLEJIDAD NO ES POSIBLE LA “INTUICION FISICA”

- Es necesario manejar relaciones matemáticas entre las variables implicadas, para predecir y explicar cuantitativamente el efecto de las variables
- Un gran número de modelos de interés para los ingenieros químicos están constituidos por sistemas de ecuaciones diferenciales
- En primer lugar, se abordan las ecuaciones diferenciales ordinarias



2.3 Resolución de sistemas de ecuaciones diferenciales ordinarias aplicadas a sistemas de interés en I.Q.

• Hay dos tipos de ecuaciones diferenciales ordinarias:

(i) de valor inicial

(ii) con condiciones de contorno

ODE de valor inicial son aquellas en las que el valor de las variables dependientes se especifica para un único valor de la variable independiente.

ODE con condiciones de contorno son aquellas en las que el valor de las variables dependientes se especifica para dos valores de la variable independiente.



2.3 Resolución de sistemas de ecuaciones diferenciales ordinarias aplicadas a sistemas de interés en I.Q.

- MATLAB tiene varias funciones para resolver ecuaciones diferenciales ordinarias de valor inicial
- Los problemas de valor inicial tienen la forma:

$$\frac{dy_1}{dt} = f_1(t, y_1, \dots, y_n)$$

M M

$$\frac{dy_n}{dt} = f_n(t, y_1, \dots, y_n)$$

Condición inicial

$$t = t_0 \quad y_1 = y_{10}$$

M

$$y_n = y_{n0}$$



INFORMATICA APLICADA A LA INGENIERÍA QUIMICA: 2ª parte MATLAB

2.3 Resolución de sistemas de ecuaciones diferenciales ordinarias aplicadas a sistemas de interés en I.Q.

>>help funfun

- La mejor es 'ode15s' (la más eficiente).

Sintáxis:

$[t,y]=ode15s('nombrefunción',vectort,vectoryini,opciones)$

$t=[t_0 \ t_1 \dots t_{final}]'$

t es un vector columna en el que se guardan los tiempos (o los valores de la variable independiente que sea) en los que se desea conocer la solución. (Matlab lo dispone en columna)



2.3 Resolución de sistemas de ecuaciones diferenciales ordinarias aplicadas a sistemas de interés en I.Q.

$[t,y]=ode15s('nombrefunción',vectort,vectoryini,opciones)$

$$\begin{array}{l} \frac{dy_1}{dt} = f_1(t, y_1, \dots, y_n) \\ \quad \quad \quad M \quad \quad \quad M \\ \frac{dy_n}{dt} = f_n(t, y_1, \dots, y_n) \end{array}$$

y = matriz con la solución en los tiempos deseados:

$$\begin{array}{l} \left[\begin{array}{cccc} y_1^{t_0} & \cdot & \cdot & \cdot & y_n^{t_0} \\ \cdot & & & & \cdot \\ \cdot & & & & \cdot \\ \cdot & & & & \cdot \\ y_1^{t_{\text{final}}} & \cdot & \cdot & \cdot & y_n^{t_{\text{final}}} \end{array} \right] \begin{array}{l} \longrightarrow \text{Vector solución a } t_0 \\ \\ \longrightarrow \text{Vector solución a } t_{\text{final}} \end{array} \end{array}$$



2.3 Resolución de sistemas de ecuaciones diferenciales ordinarias aplicadas a sistemas de interés en I.Q.

`[t,y]=ode15s('nombrefunción',vectort,vectoryini,opciones)`

'nombrefunción'=función en la que se calculan las derivadas temporales de las variables dependientes.

Estructura:

`function dydt=nombrefuncion(t,y)`

`dydt(1)=f1(t,y(1),...,y(n));`

`⋮`

`dydt(n)=fn(t,y(1),...,y(n));`

El argumento de salida de este fichero debe ser un vector columna con estos términos

`dydt=[f1(t,y(1),...,y(n));.....;dydt(n)=fn(t,y(1),...,y(n))]`



INFORMATICA APLICADA A LA INGENIERÍA QUIMICA: 2ª parte MATLAB

2.3 Resolución de sistemas de ecuaciones diferenciales ordinarias aplicadas a sistemas de interés en I.Q.

$[t,y]=ode15s(\text{'nombrefunción'}, \text{vectort}, \text{vectoryini}, \text{opciones})$

$\text{vectort} = [t_0 \dots t_{\text{final}}]$ Vector fila con los tiempos en los que se desea la solución. Aporta sus elementos al vector de salida t.

$\text{vectoryini} = [y_{10} \dots y_{n0}]$ Vector fila con los valores iniciales de las variables dependientes.

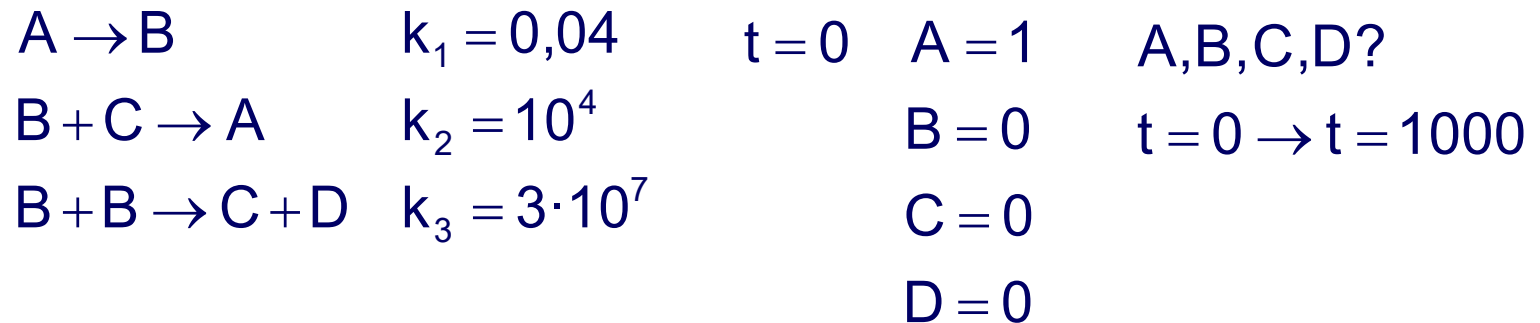
$\text{opciones} =$ vector opcional obtenido con 'odeset'



2.3 Resolución de sistemas de ecuaciones diferenciales ordinarias aplicadas a sistemas de interés en I.Q.

Ejemplo: Reacciones simultáneas:

Evolución de la concentración de varias especies químicas que intervienen en varias reacciones simultáneas.



Nomenclatura:

$A = y(1)$
$B = y(2)$
$C = y(3)$
$D = y(4)$

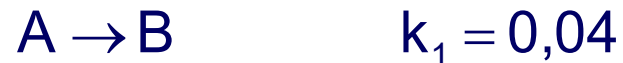
SE DESEA CONOCER LAS
CONCENTRACIONES DE A, B, C y D
ENTRE $t=0$ y $t=1000$.



INFORMATICA APLICADA A LA INGENIERÍA QUIMICA: 2ª parte MATLAB

2.3 Resolución de sistemas de ecuaciones diferenciales ordinarias aplicadas a sistemas de interés en I.Q.

Preparamos la función **chem.m**



$$\frac{dA}{dt} = -k_1 A + k_2 B C$$

$$\frac{dB}{dt} = k_1 A - k_2 B C - k_3 B^2$$

$$\frac{dC}{dt} = -k_2 B C + k_3 B^2$$

$$\frac{dD}{dt} = +k_3 B^2$$

```
function dydt=chem(t,y)
```

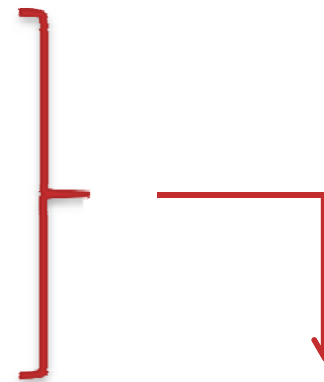
```
dydt(1)=-0.04*y(1)+1e4*y(2)*y(3);
```

```
dydt(2)=0.04*y(1)-1e4*y(2)*y(3)-3e7*y(2)^2;
```

```
dydt(3)=3e7*y(2)^2-1e4*y(2)*y(3);
```

```
dydt(4)=3e7*y(2)^2;
```

```
dydt=[-0.04*y(1)+1e4*y(2)*y(3); 0.04*y(1)-1e4*y(2)*y(3)-3e7*y(2)^2;  
3e7*y(2)^2-1e4*y(2)*y(3); 3e7*y(2)^2;]
```



INFORMATICA APLICADA A LA INGENIERÍA QUIMICA: 2ª parte MATLAB

2.3 Resolución de sistemas de ecuaciones diferenciales ordinarias aplicadas a sistemas de interés en I.Q.

Preparamos el programa:

`principalchem.m`

```
t= linspace(0,1000,100); %para obtener curvas de 100 puntos
```

```
t=vectort;
```

```
y=[1 0 0 0]; %[A0 B0 C0 D0]
```

```
y=vectoryini;
```

```
[t,y]=ode15s('chem',vectort,vectoryini)
```

```
plot.....
```

- Después de ejecutar 'principalchem' los resultados se guardan en t e y
- Para pasar argumentos adicionales a la función con las derivadas temporales, se usa la estructura 'global'



INFORMATICA APLICADA A LA INGENIERÍA QUIMICA: 2ª parte MATLAB

2.3 Resolución de sistemas de ecuaciones diferenciales ordinarias aplicadas a sistemas de interés en I.Q.

principalchem.m

```
t=linspace(0,1000,100); %para obtener curvas de 100 puntos
vectort=t;
vectoryini=[1,0,0,0]; %[A0 B0 C0 D0]
options=odeset('RelTol',1e-12,'AbsTol',1e-12,'OutputFcn','odeplot')
[t,y]=ode15s('chem',vectort,vectoryini,options)
plot(t,y(:,1),t,y(:,2),t,y(:,3),t,y(:,4)) %Elijo que quiero representar
```

```
function dydt=chem(t,y)
dydt=[-0.04*y(1)+1e4*y(2)*y(3);0.04*y(1)-
1e4*y(2)*y(3)-3e7*y(2)^2;3e7*y(2)^2-
1e4*y(2)*y(3);3e7*y(2)^2;]
```



2.3.1 Funcionamiento de ode15s

- Para adaptar ode15s a un problema, hay que conocer cómo funciona
- Para ello, se modifica chem.m

```
function dydt=...
```

```
t
```

```
y
```

```
pause
```

Estas líneas hacen que el programa se pare cada vez que ODE15s llama a la función 'chem'

Esquema de funcionamiento



2.3.1 Funcionamiento de ode15s

Esquema de funcionamiento



t es un escalar, que comienza siendo el valor inicial, y cada vez que pulsamos aumenta en una pequeña cantidad, que no tiene que ver con el vector de tiempos vectort y es un vector columna en el que cada elemento va cambiando ligeramente con cada pulsación.

Con ese tiempo y el valor de y se calculan el vector de derivadas con respecto al tiempo, que es la variable de salida de la función.



2.3.2 Uso de 'odeset' y representación gráfica de los resultados

`[t,y]=ode15s('nombrefunción',vectort,vectoryini,opciones)`

El programa construido anteriormente emplea las opciones por defecto del integrador ODE15s

- Para modificar las opciones de 'ode15s', se emplea el comando 'odeset'. Su sintáxis es:

`opciones=odeset('nombre1',valor1,'nombre2',valor2,...)`

'nombre1' y 'nombre2' = nombres asignados a las opciones modificables

'valor1' y 'valor2' = valores asignados a cada una de las opciones

- Sólo analizaremos las opciones:
- 'Reltol' (tolerancia relativa),
- 'AbsTol' (tolerancia absoluta), y
- 'OutputFcn' (controla la salida de datos de la función)



2.3.2 Uso de 'odeset' y representación gráfica de los resultados

Para cambiar estas opciones, se añade a principalchem.m

Para hacer cambios en las tolerancias y visualizar como se obtienen los resultados durante la integración, añadimos la siguiente línea antes de llamar a ODE15s en principalchem.m:

```
opciones=odeset('RelTol',1e-12,'AbsTol',1e-12,'OutputFcn','odeplot');
```

•Se incluye después del vectoryini:

```
[t,y]=ode15s('chem',vectort,vectoryini,opciones);
```



INFORMATICA APLICADA A LA INGENIERÍA QUIMICA: 2ª parte MATLAB

2.3.2 Uso de 'odeset' y representación gráfica de los resultados

- Para representar gráficamente los resultados de forma controlada, usamos 'plot':

```
plot(t,y(:,1))
```

```
hold on
```

```
plot(t,y(:,2))
```

```
hold off
```

hold on mantiene abierta la figura y hold off cierra la figura

- Con semilogy se cambia a escala semilogarítmica

```
semilogy(t,y(:,1))
```

```
hold on
```

```
semilogy(t,y(:,2))
```

```
hold off
```

- Con loglog se obtiene escala doble logarítmica



2.3.2 Uso de 'odeset' y representación gráfica de los resultados

- Para exportar los resultados y representarlos con otro programa, se emplea el comando 'save'. Sintáxis:

`save 'nombrefichero' matriz –ASCII`

Se guardan los datos en un fichero ASCII (nombre fichero) asociado a una matriz de matlab de donde se leen los datos.

Se añade a `principalchem.m`

`datosorigin=[t y];`

`save 'datosorigin.dat' datosorigin –ASCII (Lo crea en el workspace)`



2.3.2 Uso de 'odeset' y representación gráfica de los resultados

- Se guardan los datos como (con chem):

$$\begin{bmatrix} t_0 & y_1^0 & y_2^0 & y_3^0 & y_4^0 \\ - & - & - & - & - \\ - & - & - & - & - \\ - & - & - & - & - \\ t_{\text{final}} & y_1^{\text{final}} & y_2^{\text{final}} & y_3^{\text{final}} & y_4^{\text{final}} \end{bmatrix}$$



INFORMATICA APLICADA A LA INGENIERÍA QUIMICA: 2ª parte MATLAB

2.3 Resolución de sistemas de ecuaciones diferenciales ordinarias aplicadas a sistemas de interés en I.Q.

principalchem.m

```
t=linspace(0,1000,100); %para obtener curvas de 100 puntos
vectort=t;
vectoryini=[1,0,0,0]; %[A0 B0 C0 D0]
options=odeset('RelTol',1e-12,'AbsTol',1e-12,'OutputFcn','odeplot')
[t,y]=ode15s('chem',vectort,vectoryini, options)
plot(t,y(:,1),t,y(:,2),t,y(:,3),t,y(:,4)) %Elijo que quiero representar
datosoriginexcell=[t y];
save datosoriginexcell.dat
```

```
function dydt=chem(t,y)
dydt=[-0.04*y(1)+1e4*y(2)*y(3);0.04*y(1)-
1e4*y(2)*y(3)-3e7*y(2)^2;3e7*y(2)^2-
1e4*y(2)*y(3);3e7*y(2)^2;]
```



2.4 Resolución de sistemas de ecuaciones diferenciales ordinarias con las funciones ode23 y ode45.

La nomenclatura es muy similar. Lo vemos con un ejemplo:

Resolución de un problema de valor inicial de un sistema de ecuaciones diferenciales de primer orden con ode23.

$$\begin{aligned}y_1 &= 2x_1 + x_2 + 5x_3 + e^{-2t} \\y_2 &= -3x_1 - 2x_2 - 8x_3 + 2e^{-2t} - \cos(3t) \\y_3 &= 3x_1 + 3x_2 + 2x_3 + \cos(3t)\end{aligned}$$

Valores de t entre 0 y $\pi/2$

$$x_1(0) = 1, \quad x_2(0) = -1, \quad x_3(0) = 0$$



PROGRAMA Y FUNCION

FUNCION: contiene las tres expresiones matemáticas.

```
function F=sistemaf(t,x);  
F(1,1)=2*x(1)+x(2)+5*x(3)+exp(-2*t);  
F(2,1)=-3*x(1)-2*x(2)-8*x(3)+2*exp(-2*t)-cos(3*t);  
F(3,1)=3*x(1)+3*x(2)+2*x(3)+cos(3*t);
```

PROGRAMA:

```
[t,x]=ode23('sistemaf', [0 pi/2], [1;-1;0]);  
plot(t,x(:,1),'+',t,x(:,2),'x',t,x(:,3),'o')  
hold  
grid
```

En la representación gráfica los valores de x_1 , x_2 y x_3 se representan con los símbolos +, x, y o respectivamente.



Añadimos al Programa:

```
title('Example 2')  
text(0.3,14,'-+-x_1')  
text(0.3,10,'-x-x_2')  
text(0.3,-12,'-o-x_3')  
xlabel('time')  
hold off
```

Añadimos título

Coordenadas de variable y símbolo

Etiqueta eje x

