MÉTODOS NUMÉRICOS
GUÍA DE LABORATORIO NRO. 11
ECUACIONES DIFERENCIALES ORDINARIAS

OBJETIVOS:

Utilizar comandos y funciones para cálculo de ecuaciones diferenciales ordinarias.
Resolver problemas mediante el uso del MATLAB.

MEDIOS Y MATERIALES EDUCATIVOS:

Guía de laboratorio, computadora, software de Matlab, tutoriales y manuales de Matlab, apuntes, Internet y flash memory.

INFORME:

Realizar un informe del laboratorio realizado, puede ser individual o de un máximo de dos estudiantes.

TAREA 1. MÉTODO DE EULER

Analizar el siguiente programa, y verificar su correcto funcionamiento con tres funciones:

Esta función permite la construcción de las aproximaciones a la solución del problema inicial $Y'=f(X,Y)$ con $Y(a)=Y_0$ en $[a,b]$ dadas por:

$$Y_{k+1} = Y_k + h\, f(X_k,Y_k) \quad \text{para } k=0, 1,\ldots\ldots, N-1$$

```
function E=Euler(f,a,b, Y_X0,N)
%Datos
%  - f es la función, almacenada como
%  una cadena de caracteres de 'f'
%  - a y b son los extremos derecho e izquierdo del intervalo
%  - Y_X0 es la condición inicial Y(X0)
%  M es el número de pasos
%  Resultado
%  - E=[X' Y'] siendo X el vector de las abscisas e
%    Y el vector de las ordenadas
h=(b-a)/N
X=zeros(1,N+1);
```

```
Y=zeros(1,N+1);
X=a:h:b;
Y(1)=Y_X0;
for j=1:N
   Y(j+1)=Y(j)+h*feval(f,X(j),Y(j))
end
E=[X' Y'];
```

TAREA 2. RESOLUCIÓN DE PROBLEMAS DE CAUCHY PARA ECUACIONES DIFERENCIALES ORDINARIAS DE PRIMER ORDEN

MATLAB dispone de toda una familia de funciones para resolver (numéricamente) ecuaciones diferenciales:

ode45, ode23, ode113
ode15s, ode23s, ode23t, . . .

Cada una de ellas implementa un método numérico diferente, siendo adecuado usar unas u otras en función de las dificultades de cada problema en concreto. Para problemas no demasiado «difíciles» será suficiente con la función ode45 ó bien ode23.
Exponemos aquí la utilización de la función ode45, aunque la utilización de todas ellas es similar, al menos en lo más básico. Para más detalles, consúltese la documentación.
Para dibujar la gráfica de la solución (numérica) se usará la orden:

ode45(odefun, [t0,tf], y0)

donde:

odefun es un manejador de la función que evalúa el segundo miembro de la ecuación, f(t; y). Puede ser el nombre de una función anónima dependiente de dos variables, siendo t la primera de ellas e y la segunda, o también una M-función, en cuyo caso se escribiría @odefun. Ver los ejemplos a continuación.
[t0,tf] es el intervalo en el que se quiere resolver la ecuación, i.e. t0= t0, tf= tf .
y0 es el valor de la condición inicial, y0= y0.

Ejercicio 1. Calcular la solución del Problema de Cauchy:

$$\begin{cases} y' = 5y \ \ en \ \ [0,1] \\ y(0) = 1 \end{cases}$$

Comparar (gráficamente) con la solución exacta de este problema, $y = e^{5t}$

Solución.

i. Se comienza por definir una función anónima que represente el segundo miembro de la ecuación f(t, y):

*f = @(t,y) 5*y;*

ii. Se calcula y dibuja la solución numérica:

ode45(f, [0,1], 1)

Observar que no se obtiene ningún resultado numérico en salida, únicamente la gráfica de la solución.

iii. Para comparar con la gráfica de la solución exacta, se dibuja en la misma ventana, sin borrar la anterior:

```
hold on
t = linspace(0,1);
plot(t, exp(5*t), 'r')
shg
```

iv. Para hacer lo mismo utilizando una M-función en vez de una función anónima, se tendría que escribir, en un fichero (el nombre odefun es sólo un ejemplo):

```
function [dy] = odefun(t,y)
% Segundo miembro de la ecuacion diferencial
dy = 5*y;
```

y guardar este texto en un fichero de nombre odefun.m (el mismo nombre que la función).

Este fichero debe estar en la carpeta de trabajo de MATLAB.

Después, para resolver la ecuación, usaríamos ode45 en la forma:

ode45(@odefun, [0,1], 1)

Si se desea conseguir los valores t_k e y_k de la aproximación numérica para usos posteriores, se debe usar la función ode45 en la forma:

[t, y] = ode45(odefun, [t0,tf], y0);

obteniéndose así los vectores t e y de la misma longitud que proporcionan la aproximación de la solución del problema:

$y(k) \approx y(t(k))$, k = 1, 2, ……., length(y):

Si se utiliza la función ode45 en esta forma, no se obtiene ninguna gráfica.

Ejercicio 2. Calcular el valor en t = 0:632 de la solución del problema

$$\begin{cases} y' = t\, e^{\frac{t}{y}} \\ y(0) = 1 \end{cases}$$

Solución.

Se calcula la solución en el intervalo [0; 1] y luego se calcula el valor en t = 0.632 utilizando una interpolación lineal a trozos:

```
f = @(t,y) t.*exp(t/y);
[t,y] = ode45(f, [0,1], 1);
v = interp1(t, y, 0.632)
```

Obsérvese que t e y, las variables de salida de ode45, son dos vectores columna de la misma longitud.

Ejercicio 3. Calcular (aproximaciones de) los valores de la solución del problema

$$\begin{cases} y' = -2\,y \\ y(0) = 10 \end{cases}$$

en los puntos: 0, 0.1, 0.2, . . . , 1.9, 2. Comparar (gráficamente) con la solución exacta $y = 10\,e^{-2t}$.

Solución.

```
f = @(t,y) -2*y;
t = 0:0.1:2;
[t,y] = ode45(f, t, 10);
plot(t, y, 'LineWidth', 1.1)
hold on
plot(t, 10*exp(-2*t), 'r.')
legend('Sol. aproximada', 'Sol. exacta')
shg
```

Ejercicio 4. Calcular el instante en que la solución del problema

$$\begin{cases} y' = -0.5(10\,t - \ln(y+1)) \\ y(0) = 1 \end{cases}$$

alcanza el valor y = 1:5.

Solución.

i. Comenzamos por visualizar la gráfica de la solución, para obtener, por inspección, una primera aproximación:

```
f = @(t,y) 0.5*(10*t-log(y+1));
ode45(f, [0,1], 1)
grid on
shg
```

Se ve que el valor y = 1.5 se produce para un valor de t en el intervalo [0.5; 0.6].

ii. Se utilizará la función fzero para calcular la solución de la ecuación y(t) = 1.5 escrita en forma homogénea, es decir, y(t) - 1:5 = 0, dando como aproximación inicial (por ejemplo) t = 0:5. Para ello, necesitamos construir una función que interpole los valores que nos devuelva ode45. Se usará interpolación lineal a trozos:

[ts,ys] = ode45(f, [0,1], 1);
fun = @(t) interp1(ts, ys, t) - 1.5;
fzero(fun, 0.5)

Se obtendra el valor t ≈ 0:5292.

TAREA 3. RESOLUCIÓN DE ECUACIONES NO LINEALES

En el caso de ecuaciones polinómicas, es posible calcular, de una vez, todas sus soluciones, gracias al comando,

s = roots(p)

que calcula todas las raíces (reales y complejas) de un polinomio y puede ser utilizado para calcular las soluciones reales.

Ejercicio 1. Calcular las soluciones de la ecuación polinómica:

$$x^3 - 9x^2 - x + 5 = 0$$

Solución.

i. Se comienza introduciendo el polinomio p que aparece en el primer miembro de la ecuación homogénea anterior. Recuerda que se introduce como un vector fila cuyas componentes son los coeficientes del polinomio, ordenados de mayor a menor grado. En este caso

p=[1,-9,-1,5];

Recuerda que hay que incluir los coeficientes nulos, si los hay.

ii. Se calcula las raíces

roots(p)

Se obtiene las raíces: 9.0494, -0.7685 y 0.7190 que, puesto que son todas reales, son las soluciones de la ecuación.

Recordar también, siempre, que estamos realizando cálculos numéricos con la computadora y que, por consiguiente, todo lo que calculemos es aproximado.

Ejercicio 2. El comando fzero.

Los algoritmos numéricos para resolver el problema

Hallar $x \in [a; b]$ tal que $f(x) = 0$

cuando la ecuación $f(x) = 0$ es no lineal (en el caso lineal su resolución es inmediata) son en general algoritmos iterados.

Esto significa que, a partir de un punto o intervalo iniciales (dependiendo del algoritmo), se construye una sucesión de aproximaciones (calculadas una a partir de la anterior) cuyo límite es la solución buscada. Estos algoritmos, cuando convergen, lo hacen a una de las soluciones de la ecuación. Si ésta tuviera más de una solución, habría que utilizar el

algoritmo una vez para cada una, cambiando el punto inicial.
En la práctica, lógicamente, sólo se efectúan un número finito de iteraciones y el algoritmo se detiene cuando se verifica algún criterio, previamente establecido.
MATLAB dispone de la función

solucion=fzero(funcion,xcero)

donde:
funcion es un manejador de la función que define la ecuación, f. Puede ser el nombre de una función anónima dependiente de una sola variable, o también un manejador de una M-función, en cuyo caso se escribiría @funcion.
xcero es un valor «cercano» a la solución, a partir del cual el algoritmo iterado de búsqueda de la solución comenzará a trabajar.
solucion es el valor (aproximado) de la solución encontrado por el algoritmo.

Resolver la ecuación:

$$x + \ln\left(\frac{x}{3}\right) = 0$$

tiene una solución cerca de x = 1. Calcularla.

Solución.

i. Comienza por definir una función anónima que evalúe la expresión del primer miembro:
   fun = @(x) x + log(x/3);
ii. A continuación usa el comando fzero tomando x=1 como valor inicial:
   fzero(fun,1)
   se obtiene el resultado 1.0499.

También se podría haber usado una M-función para definir la función, en lugar de una función anónima. Mostramos a continuación cómo se haría.

i. Se escribe, en un archivo de nombre mifuncion.m, las órdenes siguientes:
   function [y] = mifuncion(x)
   y = x + log(x/3);
   Se graba el archivo y se lo cierra. Recordar que el archivo tiene que llamarse igual que la M-función.
ii. En la ventana de comandos, para calcular el cero de la función se escribe:

fzero(@mifuncion,1)

Ejercicio 3. Analizar e implementar el script de Matlab para el método de la bisección, y aplicarla para resolver la ecuación:

$$\frac{1}{2}\, e^{\frac{x}{3}} - \sin(x) = 0$$

```
function biseccion(f,a,b,tol)
f=inline(f);
% inline convierte a f en una función que depende de x
n=ceil(log((b-a)/tol)/log(2));
% ceil toma el entero mayor cercano obtenido por la cota de error del método
fprintf('\n it. a b x f(x) \n')
for i=1:n
x=(a+b)/2;
fprintf('%3.0f %10.10f %10.10f %10.10f %10.10f \n',i,a,b,x,f(x))
% muestra en cada paso los valores de la iteración, de a, de b, de x y de f(x)
% la instrucción %10.10f significa dejar 10 espacios y colocar el número con 10 decimales
% la instrucción \n se emplea para cambiar a línea nueva
if f(a)*f(x)<0
b=x;
else
a=x;
end
end
fprintf('\n La aproximación de la raíz es: %3.10f \n\n',x)
```

Utilizarlo con el comando:

*biseccion('0.5*exp(x/3)-sin(x)',0.5,1,0.000001)*

TAREA 4. OPTIMIZACIÓN

Ejercicio 1. sdv Devuelve los valores singulares de la matriz A en orden descendente.

[U, S, D] = sdv(A) realiza una descomposición en valor singular de la matriz A, de modo que A = U*S*V'.

Calcule los valores singulares de una matriz de rango completo para:

```
A = [1  0  1; -1  -2  0; 0  1  -1]
```

Solución.

```
s = svd(A)
```

Ejercicio 2. Encontrar la descomposición del valor singular de la matriz rectangular A:

```
A = [1  2; 3  4; 5 6;  7  8]
```

Explicar los resultados.

Solución.

```
[U, S, V] = svd(A)
```