Práctica 4

Métodos Numéricos para problemas de valor inicial

4.1. Introducción

Los métodos analíticos de integración de ecuaciones diferenciales sólo son útiles para resolver una pequeña parte de las ecuaciones diferenciales que aparecen en problemas reales. Este hecho justifica, junto al avance de la capacidad de computación, el interés práctico por los métodos que se denominan aproximados.

Los métodos numéricos son los métodos aproximados de uso más extendido en la resolución de problemas reales científicos y técnicos. Se caracterizan porque proporcionan una solución aproximada dada mediante una tabla de valores.

En esta práctica consideramos algunos de los métodos numéricos para obtener un valor aproximado de la solución $y(t)$ del problema de valor inicial

$$\left\{ \begin{array}{rcl} y' & = & f(t,y), \\ y(t_0) & = & y_0, \end{array} \right. \tag{4.1}$$

donde $y \in \mathbb{R}^p$. Esto es, $y = (y_1, \ldots, y_p)$, $f = (f_1, \ldots, f_p)$ y

$$\left\{ \begin{array}{rcl} y_1' & = & f_1(t, y_1, \ldots, y_p), \\ & \vdots & \\ y_p' & = & f_p(t, y_1, \ldots, y_p), \end{array} \right.$$

$$y_1(t_0) = y_{1,0}, \ldots, y_p(t_0) = y_{p,0},$$

y con la hipótesis de que la función f es tal que el problema tiene solución única en un intervalo que contiene a t_0.

Matlab dispone de funciones para resolver ecuaciones diferenciales. Para ello debemos definir el campo vectorial (la función $f(t, y)$) en un fichero aparte. Algunos de los comandos más utilizados en Matlab son:

- ode23: Utiliza una combinación de métodos Runge-Kutta de órdenes 2 y 3 explícitos y paso de integración variable
- ode45: Utiliza una combinación de métodos Runge-Kutta de órdenes 4 y 5 explícitos y paso de integración variable
- ode113: Utiliza métodos multipaso de órdenes 1 a 13 explícitos y paso de integración variable

Ilustramos su uso con un ejemplo.

Ejemplo El modelo de Lotka-Volterra se suele utilizar para describir la evolución de dos poblaciones que interaccionan entre ellas como, por ejemplo, zorros (cuya población denotamos por $z(t)$) y conejos (cuya población denotamos por $c(t)$). Este sencillo modelo de evolución temporal viene dado por el siguiente sistema de ecuaciones diferenciales no lineales y acopladas (en miles de individuos y tomando que la variación de individuos es contínua)

$$\begin{aligned} \frac{dc}{dt} &= c(1-z) \\ \frac{dz}{dt} &= z(c-2). \end{aligned}$$

Este sistema no tiene solución analítica, aunque sí es fácil obtener una magnitud que depende de las variables c y z, y que se mantiene constante a lo largo del tiempo: $I(c, z) = z - 2\log(z) + c - \log(c)$. Suponemos que inicialmente las poblaciones son $c(0) = 4,\ z(0) = 1$ (luego, $I(c, z) = 5 - \log(4)$, y tiene que mantenerse este valor fijo para todos los valores de c y z posteriores) y reescribimos el sistema de la siguiente forma genérica

$$\begin{aligned} y_1' &= y_1(1-y_2) \\ y_2' &= y_2(y_1-2). \end{aligned}$$

con $(y_1(0), y_2(0)) = (4, 1)$. Para resolver este problema utilizando un método numérico, en primer lugar nos construiremos un fichero que llamaremos

fVolt.m, que contiene a la función f, y al que llamaremos cada vez que necesitemos evaluar la parte derecha de las ecuaciones diferenciales. Este fichero puede ser, por ejemplo:

```
                     fVolt.m
% Modelo de Lotka-Volterra para el sistema depredador-presa
  function Ydot = fVolt(t, Y)
  global cont
  cont=cont+1;
% Parte derecha de la ecuación diferencial
  Yd( 1) = Y(1)*(1-Y(2));
  Yd( 2) = Y(2)*(Y(1)-2);
  Ydot = [Yd(1) ; Yd(2)];
```

Hemos introducido una variable, *cont*, que podemos utilizar para ver cuántas veces ha evaluado el algoritmo numérico esta función. En problemas más complejos, esto sería un indicador del coste computacional del método numérico.

En un nuevo fichero ponemos:

```
%                y' = f(t,y)
% Métodos ode23, ode45 y ode113
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
global cont
cont = 0;
% Tiempo inicial, final y número de divisiones
t0=0; tf=20;
% Condiciones iniciales
y0 = [4 ; 1];
% Magnitud conservada
I0 = y0(1) - 2*log(y0(1)) + y0(2) - log(y0(2));
% Options
options = odeset('AbsTol',1e-8,'RelTol',1e-5);

[t,y] = ode23('fVolt',[t0,tf],y0,options);
%[t,y] = ode45('fVolt',[t0,tf],y0,options);
%[t,y] = ode113('fVolt',[t0,tf],y0,options);

[y1,y2] = meshgrid(0:.2:4.5,0:.2:3);
Dy1Dt = y1.*(1-y2); Dy2Dt = y2.*(y1-2);
```

```
quiver(y1,y2,Dy1Dt,Dy2Dt,1.5)

error = y(:,1) - 2*log(y(:,1)) + y(:,2) - log(y(:,2)) - I0;
figure(1)
hold on
plot(y(:,1),y(:,2),'r^','MarkerSize',4')
axis equal,
axis([0 4.5 0 3.2])
hold off

figure(2)
hold on
plot(log10(t),log10(error),'r^','MarkerSize',4')
```

En este fichero se integra en el intervalo $t \in [0, 20]$ y se muestra la solución junto con el campo de vectores. Se utiliza el método `ode23` pero podemos elegir también los otros. En `options` podemos introducir una lista de opciones que dispone el Matlab. En el ejemplo hemos elegido el error absoluto y el relativo. Si los reducimos en un orden de magnitud debemos esperar que los errores se reduzcan aproximadamente en esta magnitud.

Cada uno de los diferentes métodos numéricos suelen ser útiles sólo para ciertos tipos de problemas y por eso Matlab dispone de muchos métodos implementados para ser utilizados en problemas distintos. En esta práctica vamos a estudiar el método de Euler, que es el más sencillo, para entender su funcionamiento. Nos debe de ayudar a entender el funcionamiento de los métodos numéricos más elaborados y que estudiaremos en la siguiente práctica.

4.2. Método de Euler

El método de Euler es el más simple de los métodos numéricos de resolución de ecuaciones diferenciales. No es muy utilizado debido a que el error que se comete al aplicarlo, crece considerablemente con el número de iteraciones. El método de Euler es un caso particular del método del desarrollo en serie de Taylor, donde nos quedamos en el primer orden del paso de integración, h,

$$y(t_0 + h) \approx y(t_0) + hy'(t_0).$$

A partir de la ecuación diferencial $y'(t_0) = f(t_0, y_0)$, obtenemos

$$y_1 = y_0 + hf(t_0, y_0) \ .$$

Análogamente, utilizando y_1 como condición inicial, tenemos en el segundo paso,

$$y_2 = y_1 + hf(t_1, y_1),$$

y, en general,

$$y_{i+1} = y_i + hf\left(t_i, y_i\right), \qquad i = 0, 1, 2, \ldots$$

con $t_i = t_0 + ih$.

El error que se comete en la primera iteración es,

$$\begin{aligned}
\varepsilon_1 &= \mid y(t_1) - y_1 \mid = \mid y(t_1) - y_0 - hf(t_0, y_0) \mid = \\
&= \mid y(t_0) + hy'(t_0) + \frac{1}{2}h^2 y''(t_0) + \cdots - y_0 - hf(t_0, y_0) \mid = \\
&= O(h^2) \ .
\end{aligned}$$

A este error se le llama *error local* o *error de truncamiento* del método. El error local o de truncamiento del método de Euler es $O\left(h^2\right)$ mientras que el error global es $O\left(h\right)$. Esta reducción del error local al error global en un grado de h es típica de las técnicas numéricas para problemas de valor inicial.

El error aumentará a medida que aumente el número de iteraciones (segú aumente el tiempo de integración). El valor práctico de este método es limitado aunque resulta útil para obtener al menos una primera aproximación de la solución, para un valor de h suficientemente pequeño.

Para entender mejor su funcioanmiento, aplicamos el método de Euler al problema anterior utilizando el siguiente programa:

```
%                  y' = f(t,y)
% Método de Euler
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
global cont
cont = 0;
% Tiempo inicial, final y numero de divisiones
t0=0; tf=20; Npasos = 2000;
% Condiciones iniciales
y = [4 ; 1];
% Magnitud conservada
I0 = y(1) - 2*log(y(1)) + y(2) - log(y(2));
```

```
% Paso de integracion
h = (tf-t0)/Npasos;
% EU almacena las soluciones obtenidas en cada instante
EU = [t0 y' eps]; t=t0;
for i2 = 1:Npasos;
  %%%% Metodo de Euler %%%%%%%%%%%%%%%%%%%%%%
  y = y + h*fVolt(t,y);
  %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
  t = t +h;
  In = y(1) - 2*log(y(1)) + y(2) - log(y(2));
  lerror = log10(abs(In-I0));
  EU = [EU ; t y' lerror];
end

figure(1)
hold on
plot(EU(:,2),EU(:,3),'*')

[y1,y2] = meshgrid(0:.2:4.5,0:.2:3);
Dy1Dt = y1.*(1-y2);
Dy2Dt=y2.*(y1-2);
quiver(y1,y2,Dy1Dt,Dy2Dt,1.5)

figure(2)
hold on
plot(log10(EU(:,1)),EU(:,4),'*')
```

Ejercicios. Práctica 4

Considera el sistema de ecuaciones de primer orden

$$\left\{\begin{array}{rcl} x' & = & v_x \\ v_x' & = & -\dfrac{x}{(x^2+y^2)^{3/2}} \\ y' & = & v_y \\ v_y' & = & -\dfrac{y}{(x^2+y^2)^{3/2}} \end{array}\right.$$

que sirve para describir la trayectoria de un satélite alrededor de la Tierra (en caso de ser perfectamente esférica). El sistema tiene la siguiente magnitud conservada (la energía):

$$H = \frac{1}{2}(v_x^2 + v_y^2) - \frac{1}{\sqrt{x^2+y^2}}. \tag{4.2}$$

Tomando como condiciones iniciales

$$x = 1 - e, \quad y = 0, \quad v_x = 0, \quad v_y = \sqrt{(1+e)/(1-e)}$$

obtenemos una trayectoria cerrada elíptica con excentricidad, e, y de periodo $T = 2\pi$. Tomar $e = 0{,}5$ y resolver numéricamente el sistema para $t \in [0, 100]$.

1. Definir la nueva función vectorial (de dimensión 4) y preparar un nuevo fichero adaptando los métodos ode23, ode45 y ode113.

2. Resuelve el sistema utilizando el método de Euler con los siguientes pasos de integración: $h = \frac{1}{10}, \frac{1}{100}, \frac{1}{1000}$, indicando cómo disminuye el error.