Universidad Rey Juan Carlos

Práctica número 3. Curso 2011-2012

Las prácticas propuestas aquí están realizadas con la ayuda del programa de simulación Matlab. Las prácticas también se podrán realizar con el programa libre Octave disponible para los sistemas operativos Linux y Windows.

Las prácticas consisten en la elaboración de ficheros de comando Matlab para la resolución de problemas. Los "scripts" tienen que ir acompañados por una memoria explicativa y por las figuras correspondientes a cada problema.

En esta práctica se estudian los sistemas dinámicos continuos con Matlab.

1. Ecuaciones Diferenciales Ordinarias lineales

Vamos en esta sección a integrar ecuaciones diferenciales ordinarias (EDO) con Matlab usando los esquemas de integración proporcionados por este software. El tipo de ecuaciones diferenciales que vamos a estudiar es el siguiente:

$$\frac{d\mathbf{x}}{dt} = f(\mathbf{x}, t) \tag{1}$$

Donde $\mathbf{x}$ es un vector de dimensión n. Tomemos un ejemplo sencillo donde $n = 2$ y una función f lineal:

$$\begin{aligned} \tfrac{dx_1}{dt} &= ax_1 + bx_2 + c \\ \tfrac{dx_2}{dt} &= dx_1 + ex_2 + f \end{aligned} \tag{2}$$

Podemos calcular la solución numérica facilmente con Matlab gracias a un algoritmo que integra el sistema. Para ello necesitamos primero escribir la función f:

```
function y=f(t,x)

%vector columna
y=zeros(2,size(x,2));

y(1,:)=a*x(1,:)+b*x(2,:)+c;
y(2,:)=d*x(1,:)+e*x(2,:)+f;
```

La función define la ecuación diferencial. Vamos ahora a calcular una solución con una condición inicial dada. Para obtener esta solución usamos la función `ode23` que implementa un esquema de integración numérica adaptado a la mayoría de los problemas de EDO.

```
[t,y]=ode23(@f,[0 2],[0.1 0.1]);
figure(1);
plot(t,y);
```

El comando de la primera línea integra la ecuación diferencial para $t \in [0, 2]$ con las condiciones iniciales $x_1(0) = 0{,}1$ y $x_2(0) = 0{,}1$. Por ejemplo podemos integrar el sistema para los parámetros: $a = 1$, $b = -1$, $c = 0$, $d = -1$, $e = 1$, $f = 0$. Siendo un sistema líneal, podemos realizar un analisís de su comportamiento. El sistema tiene como único punto fijo el origen (0,0). Encontrar numéricamente este punto fijo puede resultar complicado, para ayudarnos tenemos la función `fminsearch` que permite hallar mínimos de funciones de n variables. La estabilidad de este punto fijo se obtiene calculando los autovalores de la matriz jacobiana asociada al sistema. Esta matriz aparece claramente si escribimos el sistema (2) en forma matricial:

$$\begin{pmatrix} \dot{x}_1 \\ \dot{x}_2 \end{pmatrix} = \begin{pmatrix} a & b \\ d & e \end{pmatrix} \begin{pmatrix} x_1 \\ x_2 \end{pmatrix} + \begin{pmatrix} c \\ f \end{pmatrix} \tag{3}$$

Para calcular los autovalores de la matriz jacobiana tenemos la función de Matlab `eig`. Siendo A la matriz cuadrada que queramos estudiar, se obtiene los autovalores y autovectores de la forma siguiente:

```
[V,D] = eig(A)
```

D es una matriz diagonal que contiene los autovalores y V una matriz que contiene los autovectores. Estos autovectores corresponden justamente con las variadades (estable o inestables) del sistema. Dado un vector director $\mathbf{v}$ y un punto (x_1, x_2) podemos representar una recta correspondiente en el plano de la forma siguiente:

```
recta=(V(:,1)/norm(V(:,1)))*(-1:1);
recta=recta+[x1 ; x2]*ones(1, length(recta));
plot(recta(1,:),recta(2,:));
```

El programa siguiente representa las variedades del sistema junto con una trayectoria y un punto fijo del sistema en el plano de fases:

```
% Solución de la ecuación diferencial.
plot(y(:,1),y(:,2),'g');
axis([-1 1 -1 1]);
hold on;
plot(0,0,'.','MarkerSize',20)

% Calculo de los autovalores del sistema
A=[a b; d e];
[V,D]=eig(A)

% Primera variedad
vv=V(:,1)*(-3:3)/(norm(V(:,1)));
vv=vv+[0;0]*ones(1,length(vv));
% Se pinta de un color u otro según tenemos una variedad estable o no.
if D(1)>=0,
    plot(vv(1,:),vv(2,:),'r');
else
    plot(vv(1,:),vv(2,:),'b');
end

% Segunda variedad
vv=V(:,2)*(-3:3)/(norm(V(:,2)));
vv=vv+[0;0]*ones(1,length(vv));
if D(4)>=0,
    plot(vv(1,:),vv(2,:),'r');
else
    plot(vv(1,:),vv(2,:),'b');
end

hold off
```

Para mejorar la interpretación del sistema podemos añadir la dirección local del flujo con flechas. Necesitamos la función `quiver` de Matlab:

```
% Representamos el flujo del plano de fase
[xx yy]=meshgrid(-1:0.2:1,-1:0.2:1);

% formamos las coordenadas en vectores columnas:
cc=[xx(:)' ; yy(:)'];

% evaluamos el flujo en los puntos elegidos
arrows=f(0,cc);

% calculamos la norma por si queremos normalizar los vectores
norma=sqrt(arrows(1,:).^2+arrows(2,:).^2);

quiver(xx(:)',yy(:)',arrows(1,:),arrows(2,:));
```

Realizar las siguientes tareas:

1. Representar en el espacio de las fases: una o varias trayectorias, las variedades asociadas al punto fijo y la dirección del flujo para varios parámetros a, b,d ye.

2. EDO no lineales

El estudio de sistemas no lineales en el plano resulta muy interesante. Como ejemplo estudiaremos el modelo de predador presa:

$$\begin{aligned} \frac{dx_1}{dt} &= x_1(\alpha - cx_2) \\ \frac{dx_2}{dt} &= x_2(\gamma x_1 - \delta) \end{aligned} \tag{4}$$

Existen dos puntos fijos para el sistema: $(0,0)$ y $(\delta/\gamma, \alpha/c)$. Por otro lado el jacobiano del sistema se calcula facilmente:

$$J = \begin{pmatrix} \alpha - cy & -cx \\ \gamma y & -\delta + \gamma x \end{pmatrix} \tag{5}$$

De forma similar al sistema lineal podemos representar trayectorias en el espacio de las fases con el campo de vectores. Sin embargo es más complicado representar las variedades estables e inestables asociadas a los puntos fijos:

```
% Solución de la ecuación diferencial.
plot(y(:,1),y(:,2),'g');
axis([0 3 0 6]);
hold on;
plot(0,0,'.','MarkerSize',20)
% Representamos el flujo del plano de fase
[xx yy]=meshgrid(0:0.2:3,0:0.2:6);

% formamos las coordenadas en vectores columnas:
cc=[xx(:)' ; yy(:)'];

% evaluamos el flujo en los puntos elegidos
arrows=f(0,cc);

% calculamos la norma por si queremos normalizar los vectores
norma=sqrt(arrows(1,:).^2+arrows(2,:).^2);

quiver(xx(:)',yy(:)',arrows(1,:),arrows(2,:));
```

Otra fuente de información interesante para el análisis de estos sistemas son las nulclinas, es decir las curvas definidas por $\dot{x} = 0$ y $\dot{y} = 0$. En nuestro sistemas, las dos nuclinas del sistema son:

$$\begin{aligned} x_2 &= \frac{\alpha}{c} \\ x_1 &= \frac{\delta}{\gamma} \end{aligned} \tag{6}$$

En este caso son dos rectas que coinciden con los ejes. La representación es muy elemental. En otros casos las nulclinas pueden llegar a tener formas complejas.

Realizar las siguientes tareas:

1. Representar en el espacio de las fases: una o varias trayectorias y la dirección del flujo eligiendo los parámetros de manera a observar un ciclo límite.
2. Representar lo mismo que en el punto 1 añadiendo las nulclinas del sistema.

Repetir para el siguiente sistema dinámico de FitzHugh-Nagumo:

$$\begin{aligned} \frac{dx_1}{dt} &= x_1 - \frac{x_1^3}{3} - x_2 + I \\ \frac{dx_2}{dt} &= 0{,}08(x_1 + 0{,}7 - 0{,}8x_2) \end{aligned} \tag{7}$$

3. Sistemas Caóticos

En esta sección vamos a representar el atractor de Lorenz en tres dimensiones. Las ecuaciones del sistema son las siguientes:

$$\begin{aligned} \frac{dx}{dt} &= \sigma(y - x) \\ \frac{dy}{dt} &= x(\rho - z) - y \\ \frac{dz}{dt} &= xy - \beta z \end{aligned} \tag{8}$$

con los siguientes parámetros: $\rho = 28$, $\sigma = 10$ y $\beta = 8/3$. El planteamiento es idéntico en tres dimensiones que en dos. Se integra el sistema con la función `ode23`. Pero para representar la trayectoria en tres dimensiones usamos la función `plot3`.

```
[t,y]=ode23(@f,[0 2],[0.1 0.1 0.1]);
plot3(y(:,1),y(:,2),y(:,3));
```

Realizar las siguientes tareas:

1. Representar en el atractor de Lorenz para dos condiciones iniciales muy cercanas usando colores diferentes para cada trayectoria.