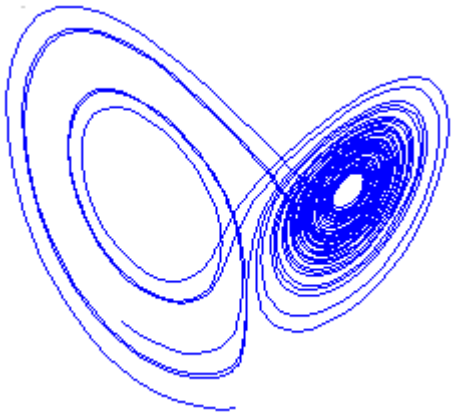
Práctica 9
Ecuaciones diferenciales ordinarias



Héctor Garbisu Arocha
Curso 2015/16
Métodos Numéricos para la Computación
Grado en Ingeniería Informática
Escuela de Ingeniería Informática
Universidad de Las Palmas de Gran Canaria

Índice

1. EDO de 2º orden como sistema de ecuaciones

En el primer ejercicio resolveremos un problema clásico utilizando el solucionador de ecuaciones diferenciales de Matlab. En concreto, y para todos los demás ejercicios, usaremos el algoritmo llamado ode45, que es el de Dorman-Prince.

Para ello, se divide la ecuación diferencial de 2º orden en dos ecuaciones de pimer orden.

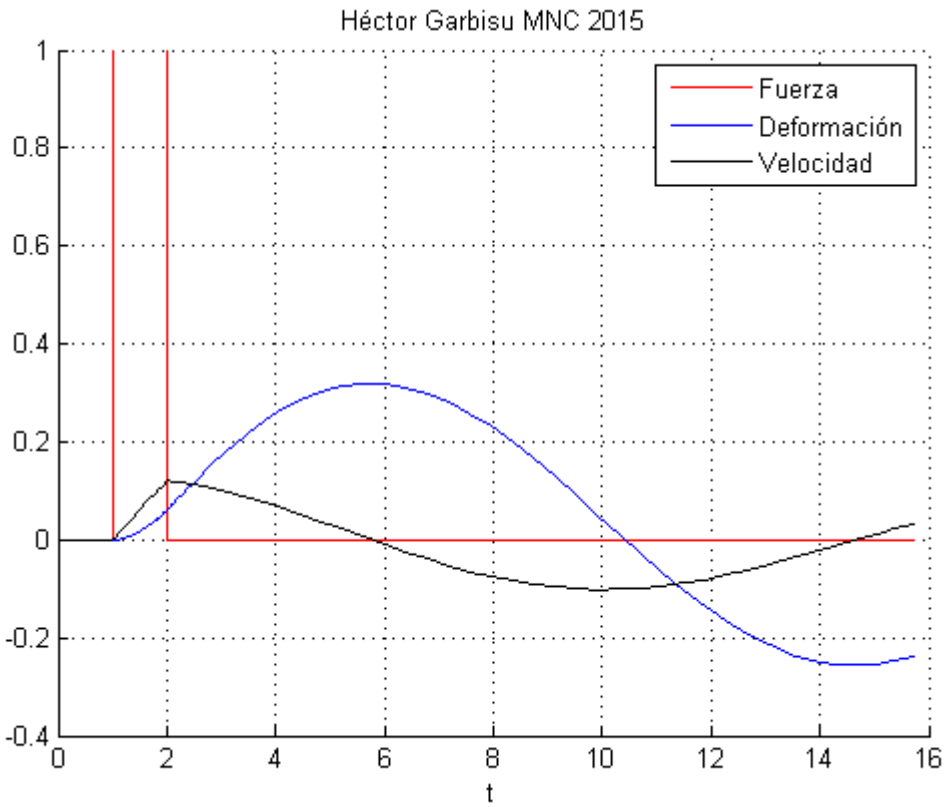
```
    dx(1,1) = x(2);
    dx(2,1) = (f(t)-(r)*x(2)-k*x(1))/m;
```

Y ode 45 resuelve el sistema de ecuaciones, dando como soluciones los valores de la velocidad (x') y posición (x) del resorte, a lo largo de unos puntos que se han ido calculando dinámicamente.

```
    [t,s] = ode45(@(u,v)sfunc(u,v,m,r,k,f),[rango(1),rango(2)],[x0;dx0]);
    x=s(:,1);
    v=s(:,2);
```

Representación gráfica del sistema para los siguientes valores:

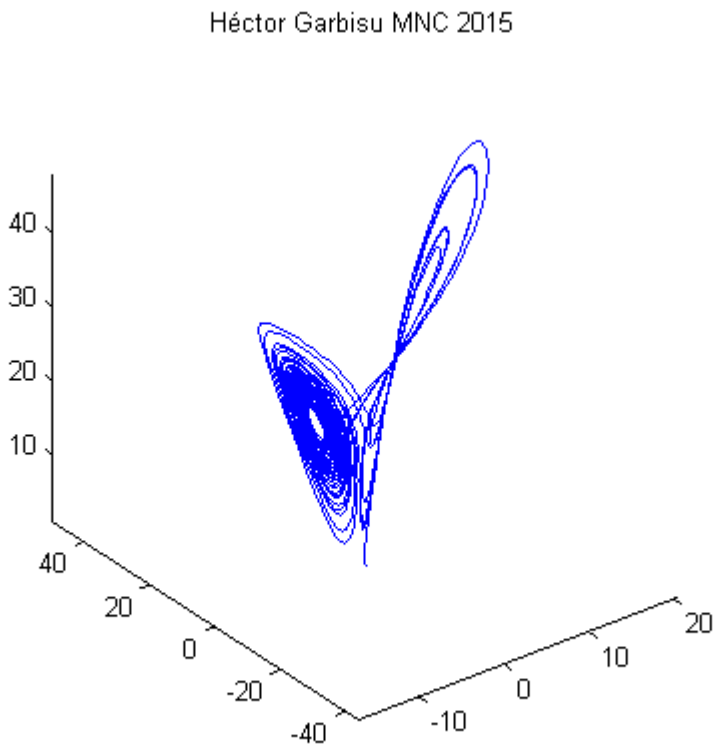
```
m = 8;
b = 0.4;
k = 1;
```



2. Atractor de Lorenz

Este sistema de ecuaciones presenta un comportamiento caótico.
Lo simularemos, dibujando las soluciones.

```
    a = 10;
    b = 28;
    c  = 8/3;
    %vecot columna para los resultados
    dX(1,1) = a*(X(2)-X(1));
    dX(2,1) = X(1).*(b-X(3))-X(2);
    dX(3,1) = X(1).*X(2) -c*X(3);
```



3. Tiro parabólico

Un problema de cinética en el que se desprecian las fuerzas de rozamiento se puede modelar fácilmente como un sistema de ecuaciones.
Para ello, las variables asociadas a la velocidad, x' e y' tendrán como valor inicial la descomposición canónica de la velocidad inicial.
La posición inicial es 0,0.

```
    [t,x] = ode45(@proyectil,[0,T],[0,v*cosd(theta),0,v*sind(theta)]);
```

Las soluciones del sistema dependerán únicamente de la velocidad inicial y el ángulo de tiro.

```
function dx = proyectil(t,x)
    g = 9.81;
    dx = zeros(4,1);
    dx(1,1) = x(2); %velocidad horizontal inicial
    dx(2,1) = 0;    %aceleraciónn horizontal nula
    dx(3,1) = x(4); %velocidad vertical inicial
    dx(4,1) = -g;   %acelaración vertical constante
end
```

```
for theta = 30:5:60
    [t,X] = TiroParabolico(T,v,theta);
    x = X(:,1); %pos x
    y = X(:,3); %pos y
    plot(x,y);
end
```