Práctica de SDC

5

TEMA

Esquema

Campos de direcciones
Soluciones
Órbitas
Planos de fase

Aplicación sobre el sistema de Lorenz

Ideas clave

5.1. ¿Cómo estudiar este tema?

Para estudiar este tema lee las **Ideas clave** que encontrarás a continuación.

Este tema es un guion de prácticas a realizar con SciLab. Para ello se presenta cómo descargar el programa, las principales características y una serie de operadores necesarios. A continuación, se introducen los conceptos de los temas 1 a 4 para poder obtener soluciones desde el punto de vista numérico que se recomienda seguir paso a paso.

Aunque SciLab se ha presentado en diversas lecciones magistrales y se ha utilizado para la generación de los gráficos de los temas anteriores, en este tema el alumno será capaz de generar dichas ilustraciones.

5.2. Introducción a Scilab

En esta introducción se asume que el alumno ya ha trabajo con *software* destinado al análisis numérico y a la representación gráfica de funciones y sistemas, por lo que nos centraremos en determinar cómo se ordenan determinadas instrucciones bajo los requisitos de este *software*.

El programa es de acceso libre y se puede descargar desde www.scilab.org. Para cualquier consulta adicional que quede fuera del material que a continuación se proporciona existen bastantes tutoriales *online* para su consulta. Asimismo, también se pueden encontrar códigos en SciLab equivalentes a otras herramientas de análisis numérico que no son gratuitas.

Representación de gráficas

» Caso 2D:

- o Representaciones 2D: plot.
- o Campos de direcciones: fchamp.

» Caso 3D:

- o Representación en 3D sobre una superficie: plot3d.
- o Representación en 3D de una curva paramétrica: parameter3d.

Propiedades de las figuras

Es importante acceder a los parámetros de cada una de las figuras para poder ajustar determinados atributos. Para ello, una vez generada la figura, hay que acceder a Editar > Propiedades de la figura (ver figura 1).

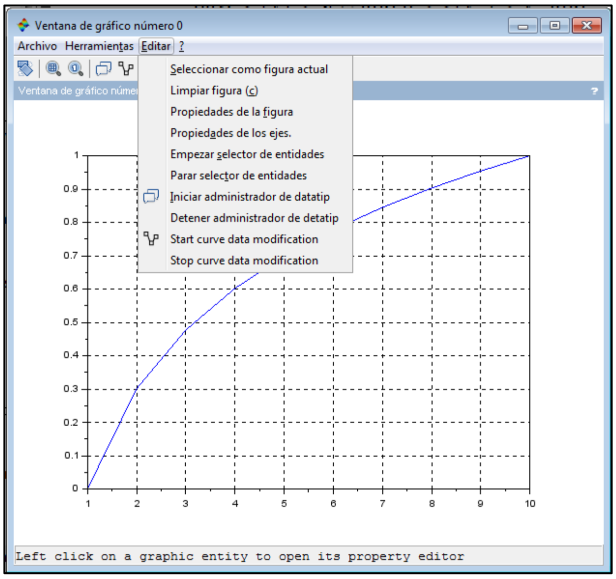


Figura 1. Propiedades de la figura.

Propiedades de los ejes

Del mismo modo se accede a las propiedades de los ejes, bien con Editar > Propiedades de los ejes, o bien desde la pantalla que presenta los atributos de la figura, clicando sobre Axes, como indica la figura 2.

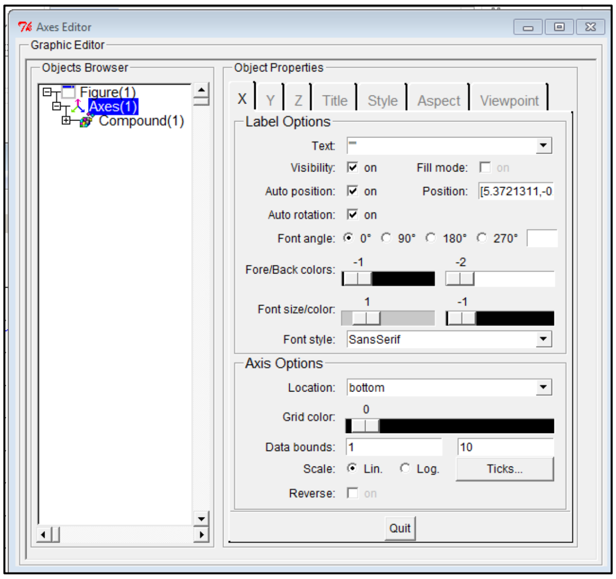


Figura 2. Propiedades de los ejes.

Útil para:

» *Axis Options > Data bounds* establecer los límites inferior y superior de representación de cada una de las variables.
» *Label Options > Text* etiquetar los ejes.
» *Aspect>Isoview* permite que los ejes tengan la misma longitud.

Propiedades del gráfico

Las propiedades de la curva o la superficie también se pueden cambiar. Esto es accesible a través del enlace que cuelga de Axes, obteniendo la pantalla de la figura 3:

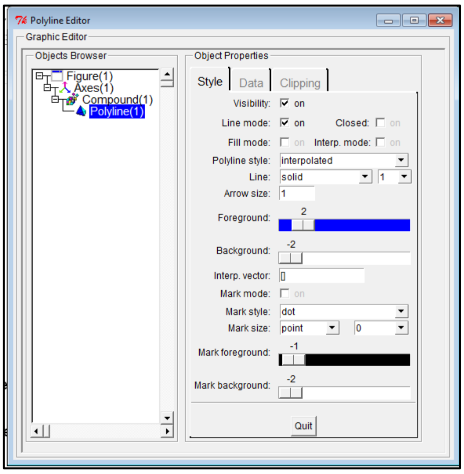


Figura 3. Propiedades del gráfico

Útil para:

» *Foreground* cambia el color.
» *Polyline style* > *Arrowed* inserta flechas en lugar de líneas.
» *Arrow size* define el tamaño de la flecha.

A todos estos atributos se puede acceder a través de la línea de comandos con a=gca();

A partir de las propiedades de sus hijos se puede acceder a estas características.

Scripts

Desde la consola, accedemos a Aplicaciones > SciNotes.

SciNotes es un editor de texto que permite introducir el programa, *script*… a guardar para poder ejecutarlo desde la consola cuantas veces sea necesario. Se le pueden dar parámetros de entrada y salida para que trabaje como una función.

Cada vez que se guarda, para poder utilizarla hay que ejecutarla. Esto se hace desde la consola. En la parte izquierda de la pantalla se encuentra el Navegador de archivos.

Clicando con el botón derecho sobre el archivo y dándole a Ejecutar.

Si es la primera vez que guardas la función, es posible que no la encuentres en el Navegador de archivos. Vuelve a entrar sobre el directorio de trabajo (por ejemplo, dándole a la flecha de *Parent Directory* y volviendo a entrar en la carpeta de trabajo) y ya debería aparecer el fichero.sci que has generado.

Otras cuestiones

- **Constantes matemáticas: s**e puede acceder a determinadas constantes matemáticas, si las precedemos del símbolo %. Por ejemplo, %pi, %i, %eps, %e.
- **Funciones útiles:**
    - Spec: obtiene los valores y vectores propios de una matriz.
    - Exp: función exponencial.
    - Log: logaritmo neperiano.
    - log10: logaritmo decimal.
    - Sqrt: raíz cuadrada.
    - Xgrid: inserta una malla en un gráfico.
    - Roots: calcula las raíces de un polinomio.
    - Ode: resuelve ecuaciones diferenciales.

5.3. Representaciones gráficas de sistemas dinámicos continuos

Ecuaciones diferenciales

» **Campo de direcciones:** en relación con las ecuaciones diferenciales, veremos la representación del campo de direcciones. Como introdujimos en el tema 2, el siguiente *script* generaría el campo de direcciones de la ecuación diferencial:

$$x' = x - t$$

```
1 deff("[xdot]=ed(t,xv)",["xd1=1";"xd2=xv(2)-xv(1)";"xdot=[xd1;xd2]"])
2 tf=-4:0.5:4;
3 xf=-4:0.5:4;
4 fchamp(ed,0,tf,xf)
```

La línea 1 define la ecuación diferencial. El vector *xdot* contiene las derivadas de t y de x, respectivamente, mientras que el vector xv contiene las variables t y x, respectivamente. Así, para este sistema, tendríamos:

$$xdot = \begin{bmatrix} t \\ x \end{bmatrix}' = \begin{bmatrix} 1 \\ x - t \end{bmatrix} = \begin{bmatrix} 1 \\ xv(2) - xv(1) \end{bmatrix}, xv = \begin{bmatrix} t \\ x \end{bmatrix}$$

A continuación se define la malla de puntos (líneas 2 y 3) en la que se quiere obtener el campo de direcciones. En este caso se define sobre la malla $[t, x] = [-4,4] \times [-4,4]$, con una separación entre puntos de 0.5.

Por último, en la línea 4 se realiza la llamada a la función fchamp con los parámetros que se indica. Esta llamada da lugar a la figura 4.

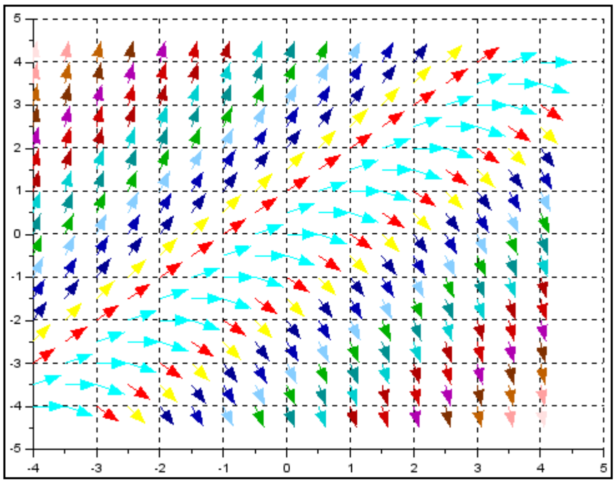


Figura 4. Campo de direcciones de $x' = x - t$.

Los atributos de la figura 4 han sido modificados respecto de la configuración por defecto. Se le han dado un tamaño de flecha y un color diferentes (Arrow size = 2, Colored: on).

Además, le hemos introducido la malla de fondo para visualizar mejor la posición de cada flecha a través de la instrucción xgrid.

Soluciones

La solución a la ecuación diferencial $x' = x - t$ es $x(t) = -e^t + t + 1$. La representación de la función es muy sencilla. Para ello hay que definir los valores que va a tomar la variable t y a continuación escribir la función en la consola para, posteriormente, ejecutar el ploteado.

```
t=linspace(-2,2,101);
x=-exp(t)+t+1;
plot(x,y)
```


La función linspace(-2,2,101) genera 101 puntos equiespaciados entre los valores -2 y 2. Al ejecutar el plot obtenemos la figura 5.

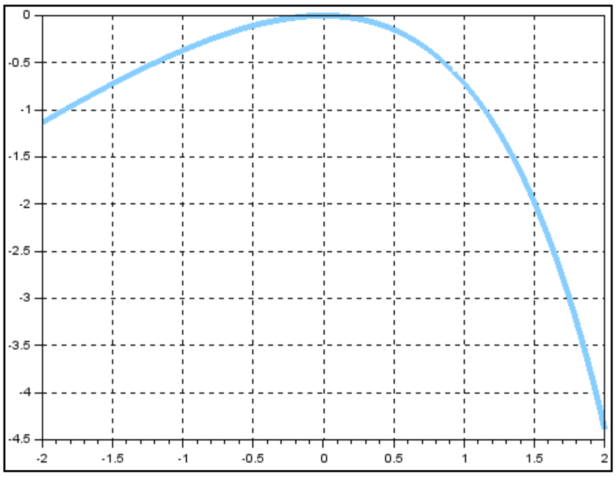


Figura 5. Solución de la ecuación diferencial $x' = x - t$

En este caso hemos modificado los atributos anchura de la línea (*line*=5.0) y color (*foreground*=12), además de introducir el mallado de fondo (*xgrid*).

Sistemas de ecuaciones diferenciales

En cuanto a los sistemas de ecuaciones diferenciales, hay que renombrar algunas de las definiciones anteriores, así como introducir algunas nuevas. Por simplicidad a la hora de introducir y visualizar los conceptos trabajaremos con sistemas de dos ecuaciones diferenciales, siendo extrapolable a los sistemas de tres ecuaciones en cuanto a la representación gráfica.

» **Campo de direcciones**: en el caso de ecuaciones diferenciales representábamos las variables t y x. Cuando estamos ante un sistema, las variables que entran en juego son $t, x,$ e y. Los sistemas sobre los que hemos estudiado los fenómenos dinámicos eran autónomos, de forma que la variable t no aparecía en ninguna ecuación diferencial. Este hecho nos va a ayudar a representar el campo de direcciones.

  Al tratarse de sistemas lineales o sistemas no lineales que han sido linealizados se expresarán como una matriz de coeficientes. De esta manera, el siguiente código es una función que tiene como entrada la matriz `A`.


```
1 function campoDirecciones2D(A)
2 ("[xdot]=cd2d(t,x)",["xd1=A(1,1)*x(1)+A(1,2)*x(2)";...
3 "xd2=A(2,1)*x(1)+A(2,2)*x(2)";"xdot=[xd1;xd2]"]);
4 tf=-5:.5:5; xf=tf;
5 fchamp(cd2d,0,tf,xf);
6 grafico=get("hdl");
7 grafico.colored="on";
8 ejes=get("current_axes");
9 ejes.isoview="on";
10 xgrid
11 endfunction
```

La línea 1 declara la función y será el nombre al que hay que acceder desde el Navegador de archivos. Las líneas 2 y 3 son similares a la del caso del campo de direcciones de una ecuación diferencial. La diferencia reside en que está en función de los valores de la matriz `A`. La línea 4 especifica en qué puntos queremos obtener el campo de direcciones y la línea 5 lo representa.

Las líneas 6 a 10 modifican determinados atributos del gráfico.

Para que represente el campo de direcciones hay que ejecutar previamente el programa, clicando con el botón secundario sobre el nombre de la función en el Navegador de archivos y dándole a Ejecutar. En la consola introducimos la matriz `A=[1 2;0 1]` y escribimos `c(A)`, resultando la figura 6.

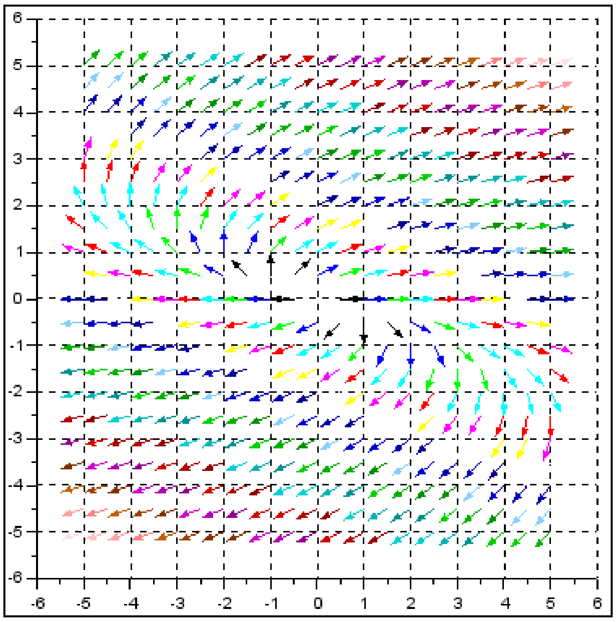


Figura 6. Campo de direcciones del sistema $x' = x + 2y, y' = y$.

Soluciones

Las soluciones aportan la resolución gráfica del sistema sobre el que estamos trabajando. Como estamos ante un sistema en el que hay que representar la evolución de las variables $x(t)$ e $y(t)$ necesitaríamos una representación 3D; sin embargo, este tipo de representaciones siempre dan lugar a confusiones, por lo que se evitan en la medida de lo posible.

La solución que se aporta es la superposición de las gráficas de $x(t)$ e $y(t)$. Para el caso del sistema descrito por $X' = \begin{bmatrix} 1 & 2 \\ 0 & 1 \end{bmatrix} X$, la solución se representa en la figura 7.

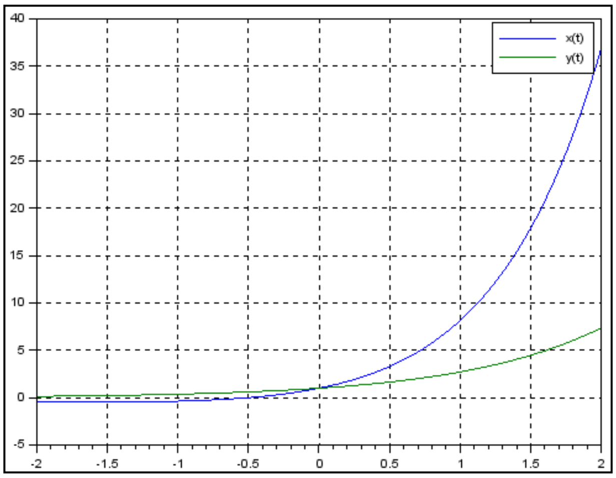


Figura 7. Solución $x(t) = e^t + 2te^t$ (azul), $y(t) = e^t$ (verde)

El código utilizado es:

```
t=linspace(-2,2,101);
x=exp(t)+2*t.*exp(t);
y=exp(t);
plot(t,[x;y])
xgrid
legend('x(t)','y(t)')
```

Órbitas

Si las soluciones representan la variación de una de las variables con el tiempo, las órbitas representan la misma variación pero de las dos variables a la vez. De nuevo, nos encontramos con el problema de la representación de tres magnitudes. En este caso, vamos a representar en el plano XY la evolución de las variables x e y, dándoles direccionalidad en forma de vector. De este modo, estaremos representando implícitamente la evolución de t.

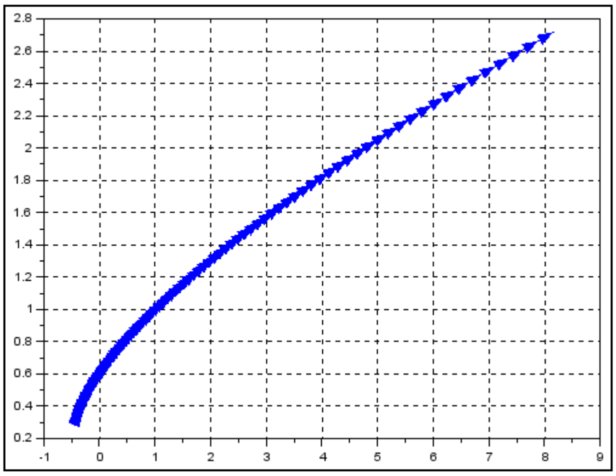


Figura 8. Órbita de $\boldsymbol{x' = x + 2y, y' = y}$

La figura 8 ilustra la órbita solución de $X' = \begin{bmatrix} 1 & 2 \\ 0 & 1 \end{bmatrix} X$. Como se puede observar, se representan los puntos (x, y) con flechas para identificar la dirección que sigue con el tiempo. En este caso, la trayectoria tendería hacia el infinito.

A continuación presentamos el código asociado a la representación de órbitas:

```
1 t=linspace(-2,2,101);
2 x=exp(t)+2*t.*exp(t);
3 y=exp(t);
4 plot(x,y)
5 figura=get("current_axes"); linea=figura.children(1).children;
6 linea.polyline_style=4;
7 xgrid
```

Las líneas 5 y 6 seleccionan el gráfico que hay dentro de la figura y modifican su representación para que aparezcan las flechas (modo 4 de polyline style).

Plano de fase

Hasta ahora solo hemos representado una de las soluciones del sistema de ecuaciones diferenciales. Como sabemos, la solución general será cualquier solución independientemente de las condiciones iniciales, que cumple las ecuaciones del sistema.

El plano de fase representa varias órbitas del sistema para diferentes valores de esas constantes no identificadas.

Dado el sistema $X' = \begin{bmatrix} -1 & 2 \\ 0 & 1 \end{bmatrix} X$, la solución analítica es:

$$\begin{bmatrix} x \\ y \end{bmatrix} = C_1 e^{-t} \begin{bmatrix} 1 \\ 0 \end{bmatrix} + C_2 e^{t} \begin{bmatrix} 1 \\ 1 \end{bmatrix}$$

Una práctica habitual para representar diferentes trayectorias es combinar valores de C_1 y C_2 como se identifica en la tabla 1.

C_1	-1	-1	-1	0	0	0	1	1	1
C_2	-1	0	1	-1	0	1	-1	0	1

Tabla 1. Combinaciones de $\boldsymbol{C_1}$ y $\boldsymbol{C_2}$ para representar el plano de fase

En este caso, representaríamos en un plano de fase hasta 9 órbitas diferentes. Para este sistema, el plano de fase asociado sería el que ilustra la figura 9.

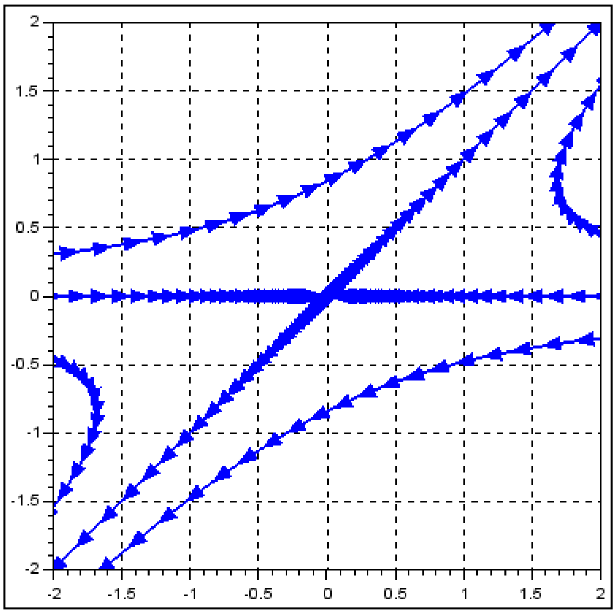


Figura 9. Plano de fase del sistema $\boldsymbol{x' = -x + 2y, y' = y}$

El código del plano de fase incluye un bucle, en el que en cada iteración se calcula y representa la órbita para un valor diferente de C_1 y C_2.

```
t=linspace(-2,2,101);
C1=[-1 -1 -1 0 0 0 1 1 1]; C2=[-1 0 1 -1 0 1 -1 0 1];
v1=[1 0]; v2=[1 1];
ind=0;
while ind<=length(C1)
      x=C1(ind)*exp(-t)*v1(1)+C2(ind)*exp(t)*v2(1);
      y=C1(ind)*exp(-t)*v1(2)+C2(ind)*exp(t)*v2(2);
      plot(x,y)
      figura=get("current_axes");
      linea=figura.children(1).children; linea.polyline_style=4;
      xgrid
      ind=ind+1;
end
```

5.4. El sistema de Lorenz

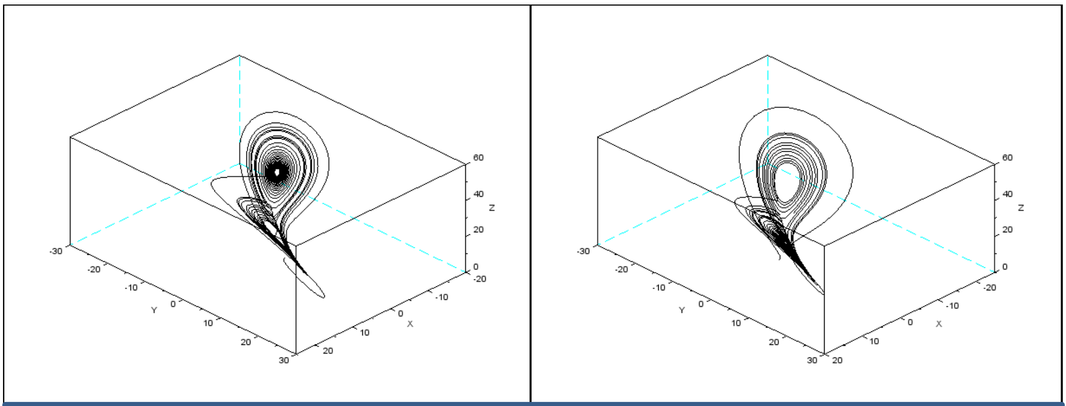
Suena el despertador. Tengo que preparar la comida para llevármela al trabajo. Hoy volveré a comer ensalada de pasta, que cuesta poco de cocinar. Pongo un poco de agua en la olla, le añado sal y espero a que hierva para echar la pasta.

En ese momento las partículas del fondo de la olla que más próximas están a la fuente de calor, en este caso es la base de la olla, tienden a subir. Cuando la diferencia entre las temperaturas es muy grande empieza a suceder el fenómeno de convección en forma de circulación circular.

El movimiento resultante puede ser descrito por un sistema de ecuaciones diferenciales que implican una cantidad innumerable de variables. Edward Norton Lorenz (1917-2008) simplificó este problema asumiendo que todas las variables permanecerían constantes salvo tres: la intensidad del movimiento de convección (x) y las variaciones horizontal (y) y vertical (z) de temperaturas. Esta simplificación incluiría tres parámetros: el número de Prandtl(σ), el número de Rayleigh(ρ) y el tamaño del sistema (β), siendo los tres parámetros positivos y cumpliendo que $\sigma > \beta + 1$.

$$\begin{aligned} x' &= \sigma(y-x) \\ y' &= \rho x - y - xz \\ z' &= xy - \beta z \end{aligned}$$

En la siguiente figura se puede observar la solución al sistema para dos puntos iniciales diferentes y un determinado valor de los parámetros.



$\sigma = 10, \rho = 28, \beta = 8/3$ $(x, y, z)_0 = (0,2,0)$	$\sigma = 10, \rho = 28, \beta = 8/3$ $(x, y, z)_0 = (0,-2,0)$

Figura 10. Soluciones del sistema de Lorenz bajo unos parámetros con diferentes condiciones iniciales

Sobre este sistema tendrás que realizar el trabajo que se propone al final del tema.

5.5. Referencias bibliográficas

Hirsch, M. W., Smale, S. y Devaney, R. L. (2004). *Differential equations, dynamical systems, and an introduction to chaos*. Elsevier.

Lo + recomendado

No dejes de leer…

Ordinary differential equations with Scilab

En el documento al que puedes acceder a través del enlace se encuentra una extensa explicación de las ecuaciones diferenciales ordinarias. En el capítulo 5 se presentan diferentes herramientas de aplicación computacional con SciLab para la resolución de las ecuaciones diferenciales ordinarias de cualquier tipo, ampliando los tipos de ecuaciones que se han visto a lo largo de este tema.

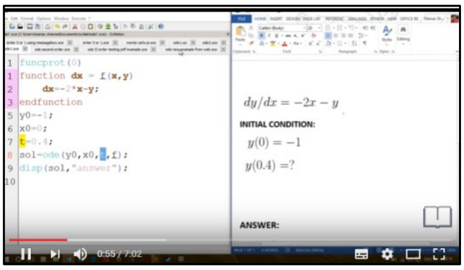
Ordinary Differential Equations with Scilab
WATS Lectures

Accede al artículo desde el aula virtual o a través de la siguiente dirección web:
http://www.math.univ-metz.fr/~sallet/ODE_Scilab.pdf

No dejes de ver…

How to solve an ODE using SCILAB

Además del método utilizado a lo largo de este tema para la resolución computacional de ecuaciones diferenciales ordinarias, existen otros métodos bastante utilizados. En el tutorial que se muestra en el vídeo que se enlaza se presenta una forma alternativa de resolver ecuaciones diferenciales ordinarias con SciLab.



Accede al vídeo desde el aula virtual o a través de la siguiente dirección web:
https://www.youtube.com/watch?v=ogC78S3FY8Q

+ Información

Recursos externos

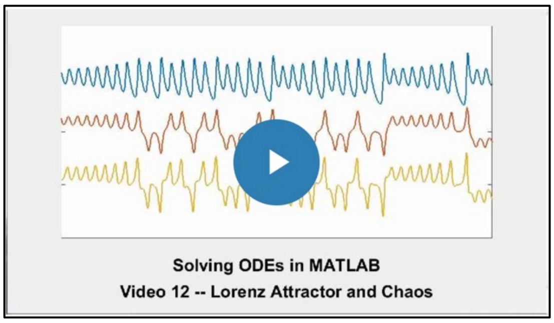
SciLab

SciLab es un *software* de computación numérica que utilizaremos para representar de manera gráfica algunos de los sistemas dinámicos bajo estudio, así como para obtener soluciones numéricas de dichos sistemas. Si has utilizado MATLAB®, su funcionamiento es muy similar.

Accede a la página desde el aula virtual o a través de la siguiente dirección web:
http://www.scilab.org/download/latest

Atractor de Lorenz y caos

El atractor caótico de Lorenz es un ejemplo de sistema no lineal de tres ecuaciones diferenciales. En este vídeo se fijan valores para los parámetros, y se muestra que las soluciones del sistema orbitan caóticamente alrededor de los dos puntos críticos inestables que tiene el sistema. Su visualización te ayudará en la elaboración de la actividad propuesta para este tema.



Accede a la página desde el aula virtual o a través de la siguiente dirección web:
https://es.mathworks.com/support/search.html/videos/solving-odes-in-matlab-12-lorenz-attractor-and-chaos-117656.html?fq=asset_type_name:video%20category:matlab/ordinary-differential-equations&page=1

Test

1. ¿Cómo establecer los límites de los ejes de un gráfico?

A. *Text*.

B. *Data bounds*.

C. *Isoview*.

D. *Tight axes*.

2. ¿Cómo cambiar el color de la línea de un gráfico?

A. *Foreground*.

B. *Background*.

C. *Fill mode*.

D. *Mark mode*.

3. ¿Cómo introducir el número imaginario?

A. %i.

B. $i.

C. ii.

D. //i.

4. ¿Cuál de las siguientes instrucciones genera un vector de 100 elementos entre -1 y 1?

A. 1:1/100:1

B. linspace (-1,1,100).

C. linspace (-1,1).

D. 1:2/99:1.

5. ¿Qué instrucción introduce un mallado en el gráfico?

A. mesh.

B. xgrid.

C. grid.

D. interp.

6. El campo de direcciones se representa a partir del comando:

A. fchamp.

B. plot.

C. surf.

D. quiver.

7. Una órbita representa la variación:

A. De una variable.

B. De dos variables.

C. De tres variables.

D. Ninguna de las anteriores es correcta.

8. El plano de fase representa un conjunto de:

A. Ecuaciones.

B. Soluciones.

C. Órbitas.

D. Ninguna de las anteriores es correcta.

9. El sistema de Lorenz describe el fenómeno de:

A. Biyección.

B. Inducción.

C. Atracción.

D. Convección.

10. En el sistema de Lorenz se tiene que cumplir:

A. $\sigma + \beta > 1$.

B. $\sigma - 1 > \beta$.

C. $\sigma - \beta < 1$.

D. $\sigma + 1 > \beta$.

