

Análisis Numérico
Curso 2018–2019
Prácticas
Hoja 2. Métodos monopaso

1 Estructura general de las prácticas para resolver ecuaciones de la forma $\mathbf{x}'(t) = \mathbf{f}(t, \mathbf{x}(t))$.

Se considera el problema de valor inicial

$$\begin{cases} x_1'(t) = x_2(t), & t \in [0, 2\pi] \\ x_2'(t) = -9x_1(t) + 8\sin(t), & t \in [0, 2\pi] \\ x_1(0) = 0 \\ x_2(0) = 4. \end{cases}$$

- a) Escribir, en la ventana de comandos de `Matlab`, la función $f(t, x)$ de la ecuación diferencial del problema de valor inicial anterior como una función anónima de `Matlab`:

```
f=@(t,x) [x(2); -9*x(1)+8*sin(t)]
```

- b) Resolver el problema de valor inicial anterior utilizando la función `ode45` de `MATLAB`, que resuelve numéricamente problemas de valor inicial mediante un método adaptativo de tipo Runge–Kutta. Concretamente, escribir

```
[t,x]=ode45(f,[0,2*pi],[0,4])  
[t,x]=ode45(f,[0:0.01:2*pi],[0,4])  
figure(1)  
subplot(2,1,1)  
plot(t,x(:,1))  
subplot(2,1,2)  
plot(t,x(:,2))  
figure(2)  
plot(x(:,1),x(:,2))
```

Comentarios acerca de los argumentos de entrada/salida de la función `ode45`:

- 1) El primer argumento de entrada (en el caso anterior `f`) es el nombre de la función f de la ecuación diferencial del problema de valor inicial. Dicha función puede venir dada como una función anónima (como en el apartado a)) o como un fichero de tipo función (véase la Observación del apartado d)).
- 2) El segundo argumento de entrada debe ser un vector fila v de los tipos $(1, 2)$ o $(1, p)$:
 - En el primer caso, el vector v determina los extremos izquierdo y derecho del intervalo en el que se quiere aproximar la solución del problema de valor inicial.
 - En el segundo caso, v es el vector de nodos en los que se quiere que el método devuelva los valores de la solución aproximada.
- 3) El tercer argumento de entrada es un vector fila de tipo $(1, n)$ (en el caso anterior $n=2$) con las condiciones iniciales del problema de valor inicial.
- 4) La función `ode45` tiene dos argumentos de salida:
 - El primero es un vector columna t de tipo $(m, 1)$ en el que aparecen los nodos en los que se quiere obtener la solución aproximada.
 - Si el segundo argumento de entrada de `ode45` es un vector v de tipo $(1, 2)$, el valor de m lo determina la propia función `ode45` automáticamente, como estime más adecuado.
 - Si dicho vector v es de tipo $(1, p)$, con $p \geq 3$, entonces $t=v^T$ (y, por tanto, $m=p$).
 - El segundo es una matriz x de tipo (m, n) de forma que en la columna j -ésima de x aparece la solución aproximada de la componente j -ésima de la solución del problema de valor inicial en los m nodos del primer argumento de salida (en otras palabras, la fila i -ésima de x es la solución aproximada correspondiente al nodo i -ésimo del primer argumento de salida).

- c) Repetir los apartados a) y b) para los siguientes problemas de valor inicial:

- 1) **Oscilador armónico:** Supongamos que un cuerpo de masa m está sujeto en el extremo de un muelle.

Si desplazamos la masa respecto de su posición de equilibrio y después la soltamos, a partir de la Segunda Ley de Newton y de la Ley de Hooke (que afirma que el muelle ejerce una fuerza de restitución proporcional al

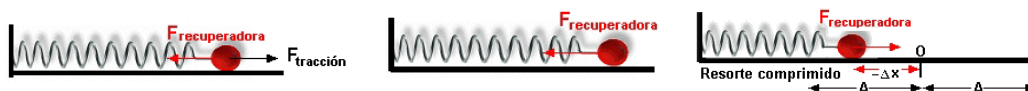


Figura 1: *Movimiento armónico simple*

alargamiento), en ausencia de rozamiento y de fuerzas externas, se tiene que el desplazamiento $x(t)$ respecto a la posición de equilibrio (*elongación*) en el instante de tiempo t es solución del problema de valor inicial

$$\begin{cases} mx''(t) = -kx(t), t \in [0, T] \\ x(0) = x_0 \\ x'(0) = v_0. \end{cases}$$

Considerar $k = m = 1$, $T = 10$, $x_0 = 1$, $v_0 = 0$. Explorar el comportamiento de las soluciones para otros datos iniciales.

II) **Ecuación de Van der Pol:** Describe el comportamiento de circuitos electrónicos no lineales

$$\begin{cases} x''(t) + \alpha (x^2(t) - \beta) x'(t) + x(t) = 0, t \in [0, T] \\ x(0) = x_0 \\ x'(0) = v_0. \end{cases}$$

Considerar $\alpha = \beta = 1$, $T = 10$, $x_0 = 0'1$, $v_0 = 0'2$. Explorar, también, el comportamiento de las soluciones para otros datos iniciales.

- d) Crear un fichero de tipo script llamado `datos.m` que contenga, en líneas distintas, ciertos datos que permiten caracterizar el problema de valor inicial cuya solución se quiere aproximar y cualquier otro dato necesario para implementar el método numérico que se vaya a utilizar. En particular, para utilizar el método implementado en Matlab en la función `ode45` y otros que implementaremos más adelante, introduciremos en el fichero `datos.m` la siguiente información:

```
f=@(t,x) [x(2);-9*x(1)+8*sin(t)]; intervalo=[0,2*pi]; x0=[0,4]; N=1000;
% k=1; m=1; f=@(t,x) [x(2);-(k/m)*x(1)]; intervalo=[0,10]; x0=[1,0]; N=1000;
```

Comentar en `datos.m` todas las líneas (usando para ello el símbolo `%`) excepto la correspondiente a la práctica que se desee ejecutar.

§ **Observación:** En lugar de definir la función f en el fichero `datos.m` como función anónima, también se pueden utilizar funciones definidas en ficheros externos con dos argumentos de entrada y uno de salida:

- El primer argumento de entrada es un número real y el segundo es un vector columna de tipo $(n, 1)$ o un vector fila de tipo $(1, n)$ (en el caso anterior $n=2$).
- El argumento de salida es otro vector columna de tipo $(n, 1)$.

Así, por ejemplo, podemos considerar la función $f(t, x)$ de la ecuación diferencial del problema de valor inicial escrita en un fichero de tipo función, con nombre `funcpvi.m` (por ejemplo), con el siguiente contenido:

```
function f=funcpvi(t,x)
f1=x(2);
f2=-9*x(1)+8*sin(t);
f=[f1;f2];
```

En este caso, introduciríamos, en el fichero `datos.m`, la siguiente información:

```
f=@funcpvi; intervalo=[0,2*pi]; x0=[0,4]; N=1000;
```

Nótese que, en la asignación de la función `f`, escribimos el nombre del fichero que contiene la función precedido del símbolo `@`.

- e) Crear un fichero tipo script con nombre `graficas.m` que, al ejecutarlo tras obtener la solución aplicando la función `ode45` de Matlab como en el apartado b), dibuje:

- Si la ecuación diferencial es escalar: la gráfica de la solución.
- Si la ecuación diferencial es en $\mathbb{R}^2$ o en $\mathbb{R}^3$: la gráfica de cada una de las componentes, en una misma ventana, usando los comandos `subplot`, `plot` y que, tras una pausa, dibuje en otra ventana la trayectoria de la solución.

Opcional: Utilizar el comando `title` y/o `legend` para indicar la curva de cada subventana.

Código de colores: Para unificar la notación, se utilizará el siguiente código de colores:

- Las gráficas para problemas escalares y las trayectorias para problemas 2D y 3D, serán de color rojo.
- Para problemas 2D, las componentes se dibujarán, respectivamente, en rojo y verde.
- Para problemas 3D los colores de las componentes serán, respectivamente, en rojo, verde y azul.

- f) Crear un fichero tipo script con nombre `testode45.m` que lea los datos del fichero `datos.m`, resuelva con la función `ode45` y dibuje utilizando el fichero `graficas.m`

Como regla general, en cada uno de los ficheros que ejecutan los programas principales (la mayor parte de los cuales se llaman `test***.m`), introducir en la primera línea la sentencia `datos` para que, de esta forma, se lea en primer lugar el contenido de `datos.m`

Para evitar conflictos con las variables utilizadas, emplear, adecuadamente, la sentencia `clear all`

2 Método de Euler (explícito).

- a) Crear un fichero de tipo función, de nombre `meuler.m`, que implemente el *Método de Euler*. Para tratar de mantener una estructura de argumentos parecida a la de la función `ode45`, el fichero `meuler.m` empezará de la siguiente forma:

```
function [t,x]=meuler(f,intervalo,x0,N)

% La función meuler resuelve un problema de valor inicial de la forma
% x'=f(t,x) en [t0,T]
% x(t0)=x0,
% con x0 en R^n, mediante el método de Euler (explícito).
%
% ENTRADA:
% f: nombre de la función (definida en formato anónimo o como fichero de tipo función de Matlab)
% del problema que se quiere resolver, con dos argumentos de entrada: el primero es un
% número real y el segundo es un vector columna de tipo (n,1)
% intervalo: [t0,T], donde está planteado el sistema de ecuaciones diferenciales
% x0: vector inicial de tipo (1,n)
% N: número de subintervalos
%
% SALIDA:
% t: vector columna de abscisas donde se va a aproximar la solución de tipo (N+1,1)
% x: matriz de ordenadas de la solución aproximada de tipo (N+1,n)
```

Una forma de testar la función `meuler` que acabamos de implementar es mediante el siguiente apartado b).

- b) Crear un fichero de tipo script, de nombre `testmeuler.m` que lea los datos del fichero `datos.m`, resuelva con la función `meuler` y dibuje la salida del algoritmo `meuler` haciendo una llamada al fichero `graficas.m`

3 Método de Euler modificado. Repetir la Práctica 2 para el *Método de Euler modificado* (ficheros `meulermmod.m` y `testmeulermmod.m`).

<u>Paso 1</u>	$x_0 \simeq \xi_0$
<u>Paso 2</u>	Para $i = 0, 1, \dots, N - 1$
	$\begin{cases} F_1 = f(t_i, x_i) \\ F_2 = f\left(t_i + \frac{h}{2}, x_i + \frac{h}{2} F_1\right) \end{cases}$
	$x_{i+1} = x_i + hF_2$

4 Método de Euler mejorado. Repetir la Práctica 2 para el *Método de Euler mejorado* (ficheros `meulermmej.m` y `testmeulermmej.m`).

<u>Paso 1</u>	$x_0 \simeq \xi_0$
<u>Paso 2</u>	Para $i = 0, 1, \dots, N - 1$
	$\begin{cases} F_1 = f(t_i, x_i) \\ F_2 = f(t_{i+1}, x_i + hF_1) \end{cases}$
	$x_{i+1} = x_i + \frac{h}{2} (F_1 + F_2)$

5 Un Método de Runge–Kutta de orden 3. Repetir la Práctica 2 para el *Método de Runge–Kutta* de orden 3 (ficheros `mrk3.m` y `testmrk3.m`) dado por

<u>Paso 1:</u> $x_0 \simeq \xi_0$ <u>Paso 2:</u> Para $i = 0, 1, \dots, N - 1$ $\begin{cases} F_1 = f(t_i, x_i) \\ F_2 = f\left(t_i + \frac{h}{2}, x_i + \frac{h}{2} F_1\right) \\ F_3 = f\left(t_i + \frac{3h}{4}, x_i + \frac{3h}{4} F_2\right) \\ x_{i+1} = x_i + \frac{h}{9} (2F_1 + 3F_2 + 4F_3) \end{cases}$
--

6 Método de Runge–Kutta de orden 4 (clásico). Repetir la Práctica 2 para el siguiente *Método de Runge–Kutta* de orden 4 (ficheros `mrk4.m` y `testmrk4.m`).

<u>Paso 1</u> $x_0 \simeq \xi_0$ <u>Paso 2</u> Para $i = 0, 1, \dots, N - 1$ $\begin{cases} F_1 = f(t_i, x_i) \\ F_2 = f\left(t_i + \frac{h}{2}, x_i + \frac{h}{2} F_1\right) \\ F_3 = f\left(t_i + \frac{h}{2}, x_i + \frac{h}{2} F_2\right) \\ F_4 = f(t_i + h, x_i + hF_3) \\ x_{i+1} = x_i + \frac{h}{6} (F_1 + 2F_2 + 2F_3 + F_4) \end{cases}$

7 Crear un fichero de tipo función, de nombre `testmet.m`, que añada como variable de entrada el nombre de una función con un método ya implementado, ejecute este último y dibuje la solución con el programa `graficas.m` creado anteriormente. Esta función debe permitir ejecutar en MATLAB lo siguiente: `testmet(@meuler)`, `testmet(@meulerm)`, `testmet(@mrk3)`,...

8 Crear un fichero de tipo función, de nombre `comp2met.m`, que tomando como dato dos métodos ya implementados, los ejecute y dibuje, siguiendo el código establecido de colores, de la siguiente forma:

- Si la ecuación diferencial es escalar: la gráfica de la solución por el primero de los métodos; tras una pausa y en otra ventana, la diferencia entre las soluciones obtenidas por ambos métodos y, tras una pausa y en otra ventana, la norma (infinito) de la diferencia entre las soluciones obtenidas por ambos métodos, utilizando el comando `legend` para mostrar su valor.
- Si la ecuación diferencial es en $\mathbb{R}^2$ o en $\mathbb{R}^3$: la gráfica de todas las componentes en la misma ventana por el primero de los métodos; tras una pausa, dibujar en otra ventana la diferencia entre las componentes respectivas de las soluciones obtenidas por ambos métodos; tras una pausa, dibujar la trayectoria de la solución por el primer método y, tras una pausa y en otra ventana, la norma (infinito) de la diferencia entre las soluciones obtenidas por ambos métodos, utilizando el comando `legend` para mostrar su valor.

Indicación: El fichero `comp2met.m` debe comenzar como `function comp2met(met1,met2)`, siendo `met1` y `met2` las variables asociadas a los nombres de las funciones que implementan los dos métodos que se quieren utilizar. Considerar, también, las sentencias utilizadas en el fichero `graficas.m`. Esta función debe permitir ejecutar en MATLAB lo siguiente: `comp2met(@meuler,@mrk3)`, `comp2met(@mrk4,@meulerm)`,...

9 Crear un fichero de tipo función, de nombre `comp2ode45.m`, que tomando como dato uno de los métodos, lo compare con la función `ode45` de MATLAB, siguiendo las indicaciones de la Práctica 8. Esta función debe permitir ejecutar en MATLAB lo siguiente: `comp2ode45(@meuler)`, `comp2ode45(@mrk3)`,...

10 Crear dos ficheros de tipo función, de nombres `solexac1.m` y `solexac2.m`, que tengan como parámetro de entrada un vector (columna) `t` de tiempos de tipo `(m,1)`, con `m` arbitrario, y devuelva una matriz de tipo `(m,2)` con la solución

exacta, evaluada en los nodos del vector t , de los problemas de primer orden correspondientes a:

$$\text{a) } \begin{cases} x'(t) = -0.1x(t) + 2y(t), & t \in [0, 10] \\ y'(t) = -2x(t) - 0.1y(t), & t \in [0, 10] \\ x(0) = 0 \\ y(0) = 1. \end{cases} \quad \text{Solución exacta: } x(t) = e^{-0.1t} \sin(2t), \quad y(t) = e^{-0.1t} \cos(2t).$$

$$\text{b) } \begin{cases} x''(t) + 2x(t) = \cos(3t), & t \in [0, 10] \\ x(0) = 1 \\ x'(0) = 0. \end{cases} \quad \text{Solución exacta: } x(t) = \frac{8}{7} \cos(\sqrt{2}t) - \frac{1}{7} \cos(3t).$$

11 En el caso de que se conozca la solución exacta del problema de valor inicial, crear un fichero de tipo función, de nombre `comp2solexac.m` que, utilizando como entradas el nombre del fichero con uno de los métodos implementados y el nombre del fichero con la solución exacta, lo ejecute, evalúe la solución exacta de la ecuación en los mismos nodos y dibuje, siguiendo el código establecido de colores, de la siguiente forma:

- a) Si la ecuación diferencial es escalar: la gráfica de la solución exacta; tras una pausa y en otra ventana, la diferencia entre las soluciones exacta y aproximada y, tras una pausa y en otra ventana, la norma (infinito) de la diferencia entre la solución exacta y la aproximada, utilizando el comando `legend` para mostrar su valor.
- b) Si la ecuación diferencial es en $\mathbb{R}^2$ o en $\mathbb{R}^3$: la gráfica de todas las componentes de la solución exacta en la misma ventana; tras una pausa, dibujar en otra ventana la diferencia entre las componentes respectivas de las soluciones exacta y aproximada; tras una pausa, dibujar la trayectoria de la solución exacta y, tras una pausa y en otra ventana, la norma (infinito) de la diferencia entre las soluciones exacta y aproximada, utilizando el comando `legend` para determinar su valor.

Esta función debe permitir ejecutar en MATLAB lo siguiente:

```
comp2solexac(@meuler,@solexac1), comp2solexac(@mrk4,@solexac2),...
```

12 Utilizar las Prácticas 8, 9 y 11 para comparar entre diversos métodos numéricos, así como con la solución exacta, cuando se les aplica a la resolución de los problemas de valor inicial de la Práctica 10.

13 Ecuación del péndulo.

El desplazamiento angular $\theta(t)$ de un péndulo de longitud L , con respecto a la vertical, es solución del problema de valor inicial

$$\begin{cases} mL\theta''(t) + 2L\beta\theta'(t) + mg\sin(\theta(t)) = F, & t \in [0, T] \\ \theta(0) = \theta_0 \\ \theta'(0) = w_0. \end{cases}$$

Se consideran los valores $m = 1$, $g = 9.8 \text{ m/s}^2$ y $T = 10$.

- Suponiendo que $F = 0$ y $L = 1 \text{ m}$ explorar, utilizando el *Método de Runge-Kutta* clásico de orden 4, el comportamiento de las soluciones para diversas elecciones de los datos iniciales. Considerar, en primer lugar, el valor $\beta = 0$ y, luego, los valores $\beta = 0.25$ y $\beta = 1.5$.
- Para cada uno de los datos iniciales escogidos, comparar los valores encontrados con los obtenidos al resolver el problema linealizado

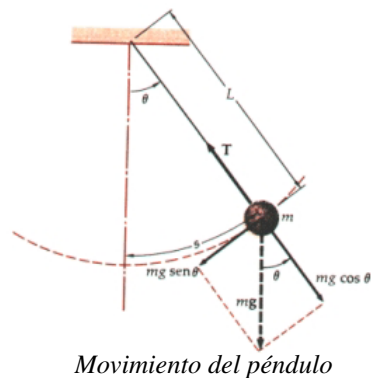
$$\begin{cases} mL\theta''(t) + 2L\beta\theta'(t) + mg\theta(t) = 0, & t \in [0, T] \\ \theta(0) = \theta_0 \\ \theta'(0) = w_0. \end{cases}$$

Nótese que la similitud de las soluciones y trayectorias únicamente se da cuando los datos iniciales son pequeños. ¿Qué ocurre si se parte con el péndulo en posición vertical, es decir, si $\theta_0 = \pi$ y $w_0 = 0$?

- Considérese ahora el valor $\beta = 0.5$. Comprobar que si $F = 1$ entonces

$$\left(\hat{\theta}_0, \hat{w}_0\right) = \left(\arcsen\left(\frac{1}{g}\right), 0\right)$$

es un punto de equilibrio del péndulo. Tomar datos iniciales próximos a este punto de equilibrio, variar F según los valores $F = 0.9$, $F = 1$ y $F = 1.1$ y observar el cambio de comportamiento.



14 Para cada uno de los siguientes *problemas autónomos* explorar, mediante el *Método de Runge–Kutta* clásico de orden 4, el comportamiento de las soluciones en el intervalo $[0, 100]$, para diversas elecciones de los datos iniciales.

- a) **Sistemas depredador–presa 1 (Lotka–Volterra):** Si $x(t)$ e $y(t)$ denotan la población de presas y depredadores, respectivamente, en el instante t , el modelo matemático más simple que gobierna su evolución, es

$$\begin{cases} x'(t) = ax(t) - bx(t)y(t) \\ y'(t) = -cy(t) + dx(t)y(t) \end{cases}$$

donde los coeficientes a, b, c, d son no negativos. Tomar los casos $a = b = c = d = 1$ y $a = 3, b = 0'2, c = 0'6, d = 5$ y datos iniciales $x(0), y(0) > 0$.

- b) **Sistema depredador–presa 2:** Un modelo de depredador–presa más completo, en el que se tiene en cuenta la saturación de presas en ausencia de depredadores y viceversa, es

$$\begin{cases} x'(t) = ax(t) - bx(t)y(t) - ex^2(t) \\ y'(t) = -cy(t) + dx(t)y(t) - fy^2(t) \end{cases}$$

donde los coeficientes a, b, c, d, e, f son no negativos. Tomar los valores $a = b = c = d = 1, e = 0'4$ y $f = 0'02$ y datos iniciales $x(0), y(0) > 0$.

- c) **Ecuación de Van der Pol:** Esta ecuación describe el comportamiento de circuitos electrónicos no lineales

$$x''(t) + \alpha(x^2(t) - \beta)x'(t) + x(t) = 0.$$

Considerar, en primer lugar, $\alpha = 1$ y tomar datos iniciales próximos al origen para $\beta = -0'2, \beta = 0$ y $\beta = 0'2$. Fijar, a continuación, $\beta = 1$ y aumentar el valor de α desde $\alpha = 1$ hasta $\alpha = 8$.

- d) **Ecuación de Duffing:** Esta ecuación describe el movimiento de una varilla bajo efectos magnéticos

$$x''(t) + \alpha x'(t) + x^3(t) - x(t) = 0.$$

Tomar, en primer lugar, $\alpha = 0$ y explorar las soluciones en torno a los 3 equilibrios del problema. Tomar, a continuación, $\alpha = 1$ y estudiar el cambio en el comportamiento de las soluciones.

- e) **Sistema de Lorenz:** En 1963, tratando de analizar el comportamiento impredecible del tiempo meteorológico, el meteorólogo E. N. Lorenz obtuvo el siguiente sistema de ecuaciones no lineales

$$\begin{cases} x'(t) = \sigma(y(t) - x(t)) \\ y'(t) = \rho x(t) - y(t) - x(t)z(t) \\ z'(t) = x(t)y(t) - \beta z(t) \end{cases}$$

Tomar $\sigma = 10, \beta = \frac{8}{3}$ y dato inicial $(0, 5, 75)$ e ir aumentando desde $\rho = 0'1$ a $\rho = 30$. Observar la dinámica en los valores intermedios $\rho = 1, \rho = 13'962$ y $\rho = 24'74$. Tomar $\rho = 100'5$ y $N = 10000$ para observar una solución periódica. Mover ρ entre $99'524$ y $100'795$ (por ejemplo $\rho = 99'65$) y observar el cambio de dinámica.

15 Oscilador armónico forzado. Se considera el siguiente problema de valor inicial gobernado por la ecuación del *oscilador armónico forzado*

$$\begin{cases} x''(t) + 2\beta x'(t) + a^2x(t) = A \cos(wt) \\ x(0) = x'(0) = 0. \end{cases}$$

Explorar, utilizando el *Método de Runge–Kutta* clásico de orden 4, el comportamiento de las soluciones en el intervalo $[0, 10]$ para las siguientes elecciones de los parámetros:

- a) Caso sin rozamiento: $\beta = 0$. Tomar los valores $A = 1, a = 10$ y $w = 12$. Ir disminuyendo el valor de w hasta que $w = a$. Continuar disminuyendo a un poco más. ¿Qué ocurre cuando $w = a$?
- b) Caso con rozamiento: $\beta > 0$. Tomar los valores $\beta = 1, A = 1, a = 10$. Comenzar con $w = 8$ e ir aumentando los valores hasta $w = 12$. ¿Qué se observa en las soluciones a medida que transcurre el tiempo? ¿Qué se observa cuando $w \simeq 9'8995$? Repetir lo anterior tomando $\beta = 15$. ¿Qué diferencia hay en las soluciones?