

Métodos Numéricos

Departamento de Matemática Aplicada II

Escuela Técnica Superior de Ingeniería

Universidad de Sevilla

Lección 6: Ecuaciones diferenciales

Índice

1. Ecuaciones diferenciales ordinarias: problemas de valores iniciales	2
1.1. Error global y orden de convergencia	4
1.2. El método de Euler	5
1.3. Métodos de Runge-Kutta explícitos	7
1.4. Pares encajados de métodos de Runge Kutta: la orden ode45	10
1.5. Eventos	13
2. Ecuaciones diferenciales ordinarias: problemas de contorno	16
2.1. El método de disparo lineal	17
2.2. El método de disparo no lineal	19
3. Apéndice A: Introducción a las ecuaciones en derivadas parciales	21
4. Soluciones de los proyectos	25

Las ecuaciones diferenciales son una de las herramientas matemáticas más usadas en el modelado de problemas técnicos y científicos. Puesto que la gran mayoría de las ecuaciones diferenciales que aparecen en dichos modelos no pueden resolverse analíticamente, la integración numérica de ecuaciones diferenciales suele considerarse la parte central del análisis numérico.

En esta lección vamos a centrarnos principalmente en las ecuaciones diferenciales ordinarias (en el apéndice A aparece una breve introducción a las ecuaciones en derivadas parciales) y vamos a estudiar y analizar métodos numéricos para resolver dos grandes tipos de problemas ligados a este tipo de ecuaciones: los problemas de valores iniciales y los problemas de contorno.

1. Ecuaciones diferenciales ordinarias: problemas de valores iniciales

Consideremos un sistema de $m \geq 1$ ecuaciones diferenciales de primer orden

$$\begin{cases} y_1'(t) &= f_1(t, y_1(t), \dots, y_m(t)), \\ \vdots & \vdots \\ y_m'(t) &= f_m(t, y_1(t), \dots, y_m(t)), \end{cases}$$

donde $f_j : I \times \mathbb{R}^m \rightarrow \mathbb{R}$, $j = 1, \dots, m$, e I es cierto intervalo de la recta real.

El sistema anterior suele escribirse, en notación más compacta, como

$$y'(t) = f(t, y(t))$$

o bien

$$y' = f(t, y),$$

donde

$$y(t) = \begin{bmatrix} y_1(t) \\ \vdots \\ y_m(t) \end{bmatrix} \in \mathbb{R}^m, \quad f(t, y(t)) = \begin{bmatrix} f_1(t, y_1(t), \dots, y_m(t)) \\ \vdots \\ f_m(t, y_1(t), \dots, y_m(t)) \end{bmatrix} \in \mathbb{R}^m.$$

El **problema de valores iniciales** surge al fijar $t_0 \in I$ e $y_0 \in \mathbb{R}^m$ y plantear la existencia de soluciones del problema (denominado de Cauchy)

$$\begin{cases} y'(t) = f(t, y(t)), \\ y(t_0) = y_0. \end{cases} \quad (1)$$

En la mayoría de las aplicaciones prácticas, la función f suele ser bastante regular, lo que implica que el problema anterior (1) tiene siempre solución y es única.

Ejemplo 1.1 ► Problema de valores iniciales

Consideremos el problema de valores iniciales

$$\begin{cases} y' = y + t, \\ y(0) = 1. \end{cases} \quad (2)$$

La función que define la ecuación diferencial, $f(t, y) = y + t$, es suficientemente regular (para ser más precisos, es una función diferenciable de todo orden en $\mathbb{R}^2$), luego el problema (2) tiene solución única.

Es fácil comprobar que la ecuación diferencial lineal $y' = y + t$ tiene por solución general la familia de curvas

$$y(t) = ce^t - 1 - t, \quad c \in \mathbb{R}.$$

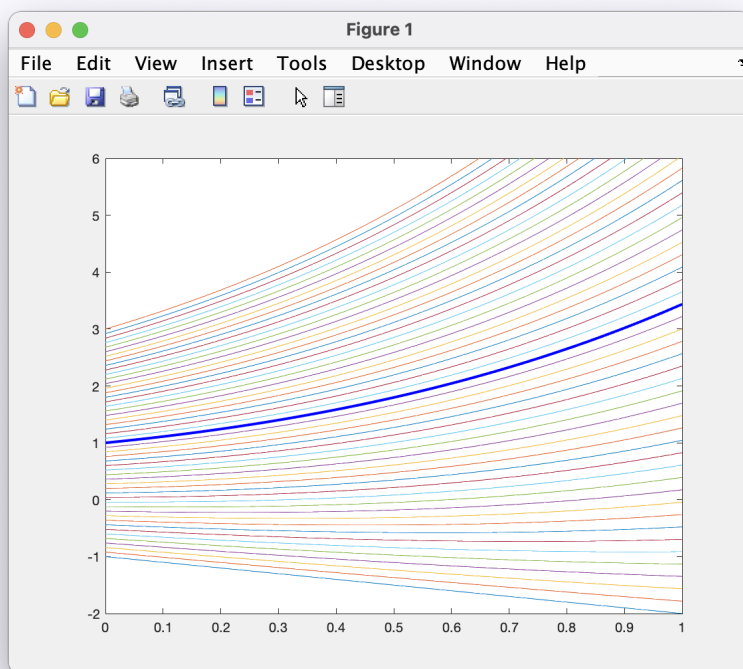
Imponiendo la condición inicial,

$$1 = y(0) = c - 1,$$

se obtiene $c = 2$. La solución única del problema de valores iniciales (2) es, por tanto,

$$y(t) = 2e^t - 1 - t.$$

En la siguiente imagen se muestran varias curvas de la solución general, correspondientes a diversos valores del parámetro c . Se ha distinguido con trazo resaltado la solución del problema de valores iniciales.



Si llamamos $Y_{(t_0, y_0)}(t)$ a la solución de (1), en general solo podemos asegurar que dicha solución está definida a la derecha en un intervalo de la forma $[t_0, t_0 + T) \subset I$, donde $0 < T = T(t_0, y_0)$ puede ser un valor muy pequeño. Otro tanto ocurre al resolver el problema hacia la izquierda (tiempos menores que t_0).

Sea $t_f \in (t_0, t_0 + T)$ fijo. Dada una partición Δ del intervalo $[t_0, t_f]$, sus diferentes puntos se denominan **instantes** o nodos temporales y forman un conjunto creciente de valores,

$$\Delta \equiv t_0 < t_1 < \dots < t_n = t_f,$$

La diferencia

$$h_k := t_{k+1} - t_k, \quad k = 0, \dots, n-1,$$

se denomina **paso** k -ésimo de la partición y

$$h_\Delta := \max_k h_k,$$

diámetro de la partición. Si todos los h_k son iguales, se dice que Δ es una **partición de paso fijo**.

La resolución numérica (a la derecha) del problema (1) consiste en obtener aproximaciones de la solución $Y_{(t_0, y_0)}$ en particiones del intervalo $[t_0, t_f]$ de diámetro arbitrariamente pequeño.

Es importante observar que, puesto que $t_n = t_f$, cuando el diámetro de la partición h_Δ tiende a cero, el número de instantes $n = n(\Delta)$ tiende a infinito.

1.1. Error global y orden de convergencia

El objetivo de los distintos métodos numéricos es, para cierta partición Δ , obtener aproximaciones de la solución en todos los instantes de Δ . Llamamos y_k a la aproximación de la solución en el instante t_k . Observando el problema (1), vemos que y_0 no es una aproximación de $Y_{(t_0, y_0)}(t_0)$, es exactamente dicho valor.

Dada una partición

$$\Delta \equiv t_0 < t_1 < \dots < t_n = t_f,$$

fijemos cierto método numérico M . Este método genera un conjunto de valores

$$y_0, y_1, \dots, y_n.$$

Se denomina **error global** del método respecto a (1) y Δ al valor

$$E_\Delta^M := \max\{\|y_k - Y_{(t_0, y_0)}(t_k)\| : k = 1, \dots, n\}.$$

Tal se comentó antes, el error en el instante inicial t_0 es nulo; de ahí que no se incluya en la expresión anterior. Como podemos ver, el error global mide la diferencia entre las aproximaciones numéricas del método y los valores de la solución exacta sobre los puntos que definen Δ .

Al analizar los distintos métodos numéricos, será fundamental analizar cómo evoluciona el error global frente al diámetro de la partición, es decir, E_Δ^M visto como $E^M(h_\Delta)$. En concreto, se dirá que el método M es de **orden de convergencia** (o, simplemente orden) p si para todo problema suficientemente regular (1) se tiene que

$$E^M(h_\Delta) \leq Ch_\Delta^p,$$

donde C es una constante positiva que no depende de la partición. Cuando $p = 1$, se dice simplemente que el método es convergente. La propia definición muestra que cuanto mayor sea el orden, mejor es el método considerado, pues obtiene cierto nivel de precisión dado a un menor coste computacional. En otras palabras, el orden de convergencia es esencialmente una medida de la eficiencia del método.

Puesto que, siempre que el método M sea convergente,

$$\lim_{h_\Delta \rightarrow 0} E^M(h_\Delta) = 0,$$

puede pensarse que para obtener en un ordenador estimaciones muy precisas de las soluciones de (1) con dicho método M , basta tomar h_Δ suficientemente pequeño. Esto dista de ser cierto, puesto que no tiene en cuenta los inevitables errores de redondeo. De hecho, lo que se detecta en la práctica es el denominado «dilema del incremento» que, esencialmente, afirma que existe un valor crítico $h_M^* > 0$ tal que si, al implementar M en un ordenador, tomamos $h_\Delta < h_M^*$, no solo no se gana precisión sino que se pierde; es más, a medida que h_Δ es mucho menor que h_M^* , las estimaciones son cada vez peores. Como es lógico, el valor crítico $h_M^* > 0$ depende del método M y su implementación en un código, pero también del ordenador que se esté utilizando.

Para la orden `ode45` de MATLAB® (que se considera la orden por defecto a aplicar en la resolución de problemas de valores iniciales y tiene orden de convergencia cinco), el valor crítico suele rondar en torno a 10^{-5} . Asimismo, para la orden `ode23` (con orden de convergencia tres), el valor crítico se sitúa habitualmente en torno a 10^{-7} . Este fenómeno es un hecho general: a medida que un método tiene orden de convergencia más alto, aumenta el valor crítico para el dilema del incremento.

1.2. El método de Euler

Posiblemente el método numérico más sencillo para abordar problemas de valores iniciales sea el **método de Euler**.

Consideremos de nuevo la solución $y := Y_{(t_0, y_0)}$ del problema (1) en $[t_0, t_f]$ y la partición

$$\Delta \equiv t_0 < t_1 < \cdots < t_n = t_f.$$

Trivialmente,

$$y(t_{k+1}) = y(t_k) + \int_{t_k}^{t_{k+1}} y'(t) dt,$$

para todo $k = 0, 1, \dots, n-1$.

Ahora bien, (1) implica que

$$y'(t) = f(t, y(t)),$$

luego

$$y(t_{k+1}) = y(t_k) + \int_{t_k}^{t_{k+1}} f(t, y(t)) dt, \quad (3)$$

para todo $k = 0, 1, \dots, n-1$.

El método de Euler surge de aproximar la integral anterior mediante la regla de cuadratura conocida como regla del rectángulo¹. En concreto,

$$\int_{t_k}^{t_{k+1}} f(t, y(t)) dt \approx f(t_k, y(t_k))(t_{k+1} - t_k) \approx f(t_k, y_k)h_k.$$

Por tanto, el método de Euler para la partición Δ se define como

$$y_{k+1} = y_k + h_k f(t_k, y_k),$$

donde $k = 0, 1, \dots, n-1$.

Puesto que la generación del método de Euler se ha basado fundamentalmente en la regla del rectángulo, parece natural conjeturar que si se emplean reglas de cuadratura más eficientes para estimar integrales, deberían obtenerse métodos numéricos más eficientes para la resolución de problemas de valores iniciales. Esto es así y fundamenta los desarrollos de métodos numéricos para la resolución de los problemas de valores iniciales.

Ejemplo 1.2 ► Método de Euler modificado

Veamos cómo se genera el denominado **método de Euler modificado**, que se basa en la regla de cuadratura del punto medio.

En concreto,

$$\begin{aligned} \int_{t_k}^{t_{k+1}} f(t, y(t)) dt &\approx (t_{k+1} - t_k) f\left(\frac{1}{2}(t_k + t_{k+1}), y\left(\frac{1}{2}(t_k + t_{k+1})\right)\right) \\ &= h_k f\left(t_k + \frac{h_k}{2}, y\left(t_k + \frac{h_k}{2}\right)\right) \\ &\approx h_k f\left(t_k + \frac{h_k}{2}, y_k + \frac{h_k}{2} f(t_k, y_k)\right), \end{aligned}$$

donde el método de Euler nos ha permitido realizar la aproximación^a

$$y\left(t_k + \frac{h_k}{2}\right) \approx y_k + \frac{h_k}{2} f(t_k, y_k).$$

En consecuencia, el método de Euler modificado para la partición Δ viene dado por

$$y_{k+1} = y_k + h_k f\left(t_k + \frac{h_k}{2}, y_k + \frac{h_k}{2} f(t_k, y_k)\right),$$

donde $k = 0, 1, \dots, n-1$.

^aEsta aproximación se puede obtener también mediante el polinomio de Taylor de primer orden de la función y centrado en t_k , esto es,

$$y\left(t_k + \frac{h_k}{2}\right) \approx y(t_k) + \frac{h_k}{2} y'(t_k) \approx y_k + \frac{h_k}{2} f(t_k, y_k).$$

¹La regla del rectángulo se origina al sustituir el integrando por su valor en el extremo inferior del intervalo de integración. Esto es,

$$\int_a^b f(x) dx \approx (b-a)f(a)$$

Como veremos más adelante, el método de Euler es convergente de orden uno y el de Euler modificado, de orden dos. Estamos, por consiguiente, ante un claro ejemplo de la estrategia mencionada anteriormente.

1.3. Métodos de Runge-Kutta explícitos

Los métodos de Euler y Euler modificado son dos casos particulares de los denominados **métodos de Runge-Kutta explícitos** (en lo sucesivo, métodos *(RK)*). Todos los métodos de esta familia parten de la igualdad (3), aproximan la integral por alguna regla de cuadratura, cambian adecuadamente los valores $y(t_k)$ por los y_k y, finalmente, generan la correspondiente recurrencia numérica para determinar el valor de y_{k+1} .

Estos métodos están completamente estandarizados. En concreto, los denominados métodos de Runge-Kutta explícitos de s **etapas** se definen mediante el siguiente proceso de recurrencia:

$$\left\{ \begin{array}{lcl} y_{k+1} & = & y_k + h_k(b_1 k_1 + \cdots + b_s k_s), \\ k_1 & = & f(t_k, y_k), \\ k_2 & = & f(t_k + c_2 h_k, y_k + h_k a_{21} k_1), \\ k_3 & = & f(t_k + c_3 h_k, y_k + h_k(a_{31} k_1 + a_{32} k_2)), \\ \vdots & & \vdots \\ k_s & = & f(t_k + c_s h_k, y_k + h_k(a_{s1} k_1 + \cdots + a_{s, s-1} k_{s-1})), \end{array} \right.$$

donde $k = 0, 1, \dots, n-1$.

Ejemplo 1.3 ► Etapas

El método de Euler consta de una única etapa,

$$y_{k+1} = y_k + h_k \underbrace{f(t_k, y_k)}_{k_1}, \quad k = 0, 1, \dots, n-1,$$

luego se expresa en la forma

$$\left\{ \begin{array}{lcl} y_{k+1} & = & y_k + h_k k_1, \\ k_1 & = & f(t_k, y_k). \end{array} \right.$$

En el método de Euler modificado se pueden distinguir dos etapas:

$$y_{k+1} = y_k + h_k \underbrace{f(t_k + \frac{h_k}{2}, y_k + \frac{h_k}{2} \underbrace{f(t_k, y_k)}_{k_1})}_{k_2}, \quad k = 0, 1, \dots, n-1.$$

Por consiguiente, se escribe en la forma

$$\left\{ \begin{array}{lcl} y_{k+1} & = & y_k + h_k k_2, \\ k_1 & = & f(t_k, y_k), \\ k_2 & = & f(t_k + \frac{h_k}{2}, y_k + \frac{h_k}{2} k_1). \end{array} \right.$$

La manera habitual en la que se presenta un método de s etapas es mediante los denominados **tableros de Butcher**. En concreto, el tablero asociado al método anterior de s etapas es

$$\begin{array}{c|cccc} 0 & 0 & & & \\ c_2 & a_{21} & & & \\ c_3 & a_{31} & a_{32} & & \\ \vdots & \vdots & \vdots & \ddots & \\ c_s & a_{s1} & a_{s2} & \dots & a_{s\ s-1} \\ \hline & b_1 & b_2 & \dots & b_{s-1} & b_s \end{array}$$

o, escrito en forma compacta,

$$\begin{array}{c|c} c & A \\ \hline & b \end{array}$$

Uno de los métodos de Runge-Kutta explícitos más conocidos (de hecho, uno de los primeros en formularse) es el denominado **método de Runge-Kutta clásico**²:

$$\left\{ \begin{array}{l} y_{k+1} = y_k + h_k \left(\frac{1}{6}k_1 + \frac{2}{6}k_2 + \frac{2}{6}k_3 + \frac{1}{6}k_4 \right), \\ k_1 = f(t_k, y_k), \\ k_2 = f\left(t_k + \frac{h_k}{2}, y_k + \frac{h_k}{2}k_1\right), \\ k_3 = f\left(t_k + \frac{h_k}{2}, y_k + \frac{h_k}{2}k_2\right), \\ k_4 = f(t_k + h_k, y_k + h_k k_3), \end{array} \right.$$

donde $k = 0, 1, \dots, n-1$.

Ejemplo 1.4 ► Tableros de Butcher

Los tableros de Butcher de los métodos de Euler, de Euler modificado y de Runge-Kutta clásico son, respectivamente,

$$\begin{array}{c|c} 0 & 0 \\ \hline & 1 \end{array} \quad \begin{array}{c|c} 0 & 0 \\ \frac{1}{2} & \frac{1}{2} \\ \hline & 0 & 1 \end{array} \quad \begin{array}{c|ccc} 0 & 0 & & \\ \frac{1}{2} & \frac{1}{2} & & \\ \frac{1}{2} & 0 & \frac{1}{2} & \\ 1 & 0 & 0 & 1 \\ \hline & \frac{1}{6} & \frac{1}{3} & \frac{1}{3} & \frac{1}{6} \end{array}$$

Como puede esperarse, el orden de convergencia de los métodos (RK) depende básicamente de la matriz A y de los vectores b y c . Por ejemplo, un método (RK) es, al menos, convergente de orden uno si, y solo si,

$$\sum_{j=1}^s b_j = 1.$$

²O de cuatro etapas. Este método se basa en la regla de cuadratura de Simpson.

Definiendo $c_1 := 0$ y $a_{i,j} := 0$ para $i \leq j$ ($i, j = 1, \dots, s$), un método (RK) es, al menos, convergente de orden dos si, y solo si,

$$i) \sum_{j=1}^s b_j = 1 \text{ (es convergente de orden 1),}$$

$$ii) \sum_{j=1}^s b_j c_j = \frac{1}{2}, \quad c_i = \sum_{j=1}^s a_{i,j}, \text{ con } i = 1, \dots, s.$$

En particular, para conseguir orden dos se necesitan al menos dos etapas. En efecto, si el número de etapas fuera uno, se tendría, a partir de la segunda identidad, que $b_1 c_1 = \frac{1}{2}$, pero $c_1 = 0$.

A medida que aumenta el orden de convergencia, aumenta el número de condiciones y su complejidad.

A partir de las expresiones anteriores, puede comprobarse que el método de Euler es convergente de orden uno y el de Euler modificado, de orden dos. El método de Runge-Kutta clásico es de orden cuatro.

Es interesante mencionar que para los métodos (RK) hay una barrera que limita el orden de convergencia con respecto al número de etapas. En concreto, el orden máximo posible de un método (RK) de s etapas es s . Más aún, la igualdad solamente puede conseguirse para $s = 1, 2, 3, 4$. Es la denominada primera barrera de Butcher. Por ejemplo, para conseguir un método (RK) de orden cinco hacen falta como mínimo seis etapas y se necesitan once etapas para conseguir uno de orden ocho.

Problema 1.1

1. *Elabora una función de MATLAB® que implemente el método de Euler con partición de paso fijo.*
2. *Elabora otra una función de MATLAB® que implemente el método de Runge-Kutta clásico, de nuevo con partición de paso fijo.*
3. *Considera el problema de valores iniciales*

$$\begin{cases} y'' + 8y' + 16y = 8e^{-2t}, \\ y(0) = 2, \\ y'(0) = 0, \end{cases}$$

donde $t \in [0, 3]$.

Puedes comprobar que la solución exacta de este problema es

$$y(t) = 4te^{-4t} + 2e^{-2t}.$$

Aproxima esta solución mediante los métodos de Euler y de Runge-Kutta clásico, tomando en ambos casos $h = 10^{-1}$. Repite los cálculos para $h = 10^{-2}, 10^{-3}, 10^{-4}$.

Calcula en todos los casos los errores cometidos y determina numéricamente el orden de convergencia de ambos métodos.

Proyecto 1.1 (solución en página 25)

El tablero de Butcher

0		0	
$\frac{1}{3}$		$\frac{1}{3}$	
$\frac{2}{3}$		0	$\frac{2}{3}$
		$\frac{1}{4}$	0 $\frac{3}{4}$

corresponde a un método de Runge-Kutta explícito de tres etapas que denotaremos por *rk3*.

1. Comprueba analíticamente que este método es, al menos, convergente de orden dos.
2. Elabora una función de MATLAB® que implemente este método con partición de paso fijo.
3. Repite el apartado 3 del problema 1.1, pero aproximando ahora la solución mediante el método *rk3*. Determina numéricamente su orden de convergencia.

1.4. Pares encajados de métodos de Runge Kutta: la orden **ode45**

La eficiencia al implementar un método numérico para resolver problemas de valores iniciales descansa fuertemente en poder adaptar el tamaño de los pasos h_k a la variabilidad de la solución del problema. En otras palabras, en zonas donde dicha solución varíe bruscamente es necesario tomar pasos de integración muy pequeños para no perder precisión; por el contrario, en zonas donde la solución sea casi constante, lo razonable sería poder elegir tamaños de paso relativamente grandes para eliminar cálculos innecesarios.

Hoy día, el procedimiento que se ha mostrado más eficaz siguiendo la filosofía que acabamos de mencionar es el de los denominados **pares encajados de métodos de Runge-Kutta**. Estos pares están formados por dos métodos (*RK*) de órdenes respectivos $\hat{p}$ y p distintos (supongamos $\hat{p} > p$) con la propiedad crucial de que los tableros de Butcher de ambos métodos son muy similares. En concreto, comparten las mismas etapas, lo que permite estimar la solución del método de orden $\hat{p}$ solo realizando la iteración, al aprovechar el cálculo previo de las etapas realizado con el método de orden p .

A grandes rasgos, y sin entrar en consideraciones muy técnicas, si asumimos que se han dado por válidos la aproximación de la solución en el instante t_k y el tamaño de paso h_k , la similitud o disparidad de las estimaciones calculadas con ambos métodos en el instante t_{k+1} es la base para definir el tamaño del paso siguiente,

$$h_{k+1} := t_{k+2} - t_{k+1}$$

y, por tanto, el instante t_{k+2} .

En concreto, si en el paso k -ésimo y con tamaño dado h_k ambas estimaciones en

$$t_{k+1} = t_k + h_k$$

son muy similares, se acepta dicho tamaño, se toma como aproximación la obtenida con el método de orden más alto y se define como posible siguiente tamaño de paso

$$h_{k+1} = \rho_k h_k,$$

para cierto $\rho_k \geq 1$. Si, por el contrario, ambas estimaciones difieren notablemente, se reduce el tamaño del paso h_k con cierto factor que tiene en cuenta las estimaciones realizadas en t_k , se obtienen nuevas estimaciones con dicho paso reducido y se realizan nuevas comparaciones. Siempre se imponen unos tamaños máximo y mínimo de paso.

Si los tableros de ambos métodos del par encajado vienen dados, respectivamente, por

$$\frac{c}{b} \mid \frac{A}{b} \quad y \quad \frac{c}{\hat{b}} \mid \frac{A}{\hat{b}}$$

el tablero del par encajado se representa mediante el símbolo

$$\frac{c}{b} \mid \frac{A}{\hat{b}}$$

La orden **ode45** está basada en un par encajado de métodos de Runge-Kutta de órdenes 4 y 5, respectivamente. Los tableros encajados de ambos métodos aparecen en la siguiente imagen.

0	0								
$\frac{1}{5}$	$\frac{1}{5}$								
$\frac{3}{10}$	$\frac{3}{40}$	$\frac{9}{40}$							
$\frac{4}{5}$	$\frac{44}{45}$	$-\frac{56}{15}$	$\frac{32}{9}$						
$\frac{8}{9}$	$\frac{19372}{6561}$	$-\frac{25360}{2187}$	$\frac{64448}{6561}$	$-\frac{212}{729}$					
1	$\frac{9017}{3168}$	$-\frac{355}{33}$	$\frac{46732}{5247}$	$\frac{49}{176}$	$-\frac{5103}{18656}$				
1	$\frac{35}{384}$	0	$\frac{500}{1113}$	$\frac{125}{192}$	$-\frac{2187}{6784}$	$\frac{11}{84}$			
b_i	$\frac{35}{384}$	0	$\frac{500}{1113}$	$\frac{125}{192}$	$-\frac{2187}{6784}$	$\frac{11}{84}$	0		
$\hat{b}_i$	$\frac{5179}{57600}$	0	$\frac{7571}{16695}$	$\frac{393}{640}$	$-\frac{92097}{339200}$	$\frac{187}{2100}$	$\frac{1}{40}$		

Además, MATLAB® también incorpora la orden **ode23**, basada en un par encajado de métodos de Runge-Kutta de órdenes respectivos 2 y 3. Esta orden puede ser una opción mejor que la anterior cuando se busquen estimaciones no muy precisas; técnicamente, cuando se impongan tolerancias no muy exigentes (*crude tolerances*).

Ambas órdenes admiten un vector de opciones denominado **odeset** que permite modificar ciertos aspectos de su ejecución. Para ello, se recomienda consultar los diferentes argumentos que puede tomar dicho vector de opciones, aunque, en particular, es de especial interés la posibilidad de cambiar los valores de las tolerancias asignados por defecto. Ambas manejan dos tolerancias, una para el error absoluto, representada por **AbsTol**, y otra para el error relativo, representada por **RelTol**.

Ejercicio 1.1

El sistema de ecuaciones diferenciales, conocido como Brusselator,

$$\begin{cases} u' = 1 + u^2v - 4u, \\ v' = 3u - u^2v, \end{cases}$$

es un modelo de cierta reacción química.

Considera las condiciones iniciales

$$u(0) = 1.5, \quad v(0) = 3.$$

Calcula, usando la orden de MATLAB® `ode45`, con las especificaciones por defecto, los valores de $u(2)$ y $v(3.5)$.

Calcúlalos de nuevo, pero en esta ocasión toma como tolerancias $\text{AbsTol}=10^{-9}$ y $\text{RelTol}=10^{-6}$.

Proyecto 1.2 (solución en página 27)

Considera el problema de valores iniciales

$$\begin{cases} y'' - \mu(1 - y^2)y' + y = 0, \\ y(0) = 2, \\ y'(0) = 0, \end{cases}$$

donde $\mu \in \mathbb{R}$.

Utilizando la orden de MATLAB® `ode45` con los argumentos del vector de opciones AbsTol y RelTol igual a 10^{-8} , obtén una representación gráfica de la solución en el intervalo temporal $[0, 20]$ y con $\mu = 0.2$. ¿Qué particularidad tiene dicha solución?

Proyecto 1.3 (solución en página 29)

Considera de nuevo el problema de valores iniciales dado en el ejercicio 1.1.

Calcula el valor de la integral

$$\int_0^{20} v^2(t) dt.$$

Observa que en su integrando aparece una de las componentes de la solución del problema de valores iniciales mencionado. El cálculo de la integral no debe realizarse, por tanto, por medio de órdenes asociadas a la cuadratura numérica, sino utilizando la orden de MATLAB® `ode45`.

Proyecto 1.4 (solución en página 30)

Determina un valor $\alpha \in \mathbb{R}$ de modo que la solución $y(t)$ del problema de valores iniciales

$$\begin{cases} y'' = \cos y, \\ y(0) = \alpha, \\ y'(0) = 0, \end{cases}$$

verifique $y'(1.2) = 0.5$. Para ello utiliza la orden de MATLAB® `ode45` y define cierta ecuación no lineal.

1.5. Eventos

En muchas ocasiones, al resolver problemas de valores iniciales, más que obtener la integración del problema en cierto intervalo temporal, lo que se busca es detectar aquellos instantes donde ocurre cierto **evento** (o eventos). Por ejemplo, al disparar un proyectil, una cuestión que podría interesar sería determinar el tiempo de impacto.

La orden `ode45` permite detectar dichos instantes, siempre que el evento pueda describirse mediante la aparición de un cero de cierta función vectorial. La sintaxis de dicha función es la siguiente:

$$[\text{value}, \text{isterminal}, \text{direction}] = \text{evento}(t, y)$$

donde **value**, **isterminal** y **direction** son vectores cuya componente k -ésima corresponde a la función evento k -ésima. Además:

- El argumento de salida **value** es el valor de la función evento.
- El argumento de salida **isterminal** igualado a 1 indica que la integración se acaba al detectar un cero de la función evento e igualado a 0, que la integración del sistema continúa.
- El argumento de salida **direction** puede tomar tres valores: 0 si se buscan todos los ceros de la función evento (valor por defecto); +1, si solo se buscan los ceros donde la función es creciente; -1, si solo se buscan los ceros donde la función es decreciente.

Una de las principales utilidades del argumento **direction** es la detección de máximos y mínimos relativos de funciones y que son soluciones de ciertos sistemas de ecuaciones diferenciales. En concreto, supongamos que para cierto instante t_0 se tiene que $y'(t_0) = 0$ e $y''(t_0) > 0$. Esto implica que t_0 es un mínimo relativo de y y, localmente en t_0 , la derivada y' es estrictamente creciente. Podemos capturar dicho punto t_0 con la opción de eventos consignando en el argumento **value** la expresión que corresponda a $y' = 0$ y como argumento **direction**, el valor 1. Análogamente, para máximos relativos del tipo $y'(t_0) = 0$ e $y''(t_0) < 0$, el razonamiento es similar, pero en esta ocasión hay que asignar al argumento **direction** el valor -1.

Ejemplo 1.5 ► Evento

Se deja caer una pelota desde cinco metros de altura. Nos planteamos determinar el tiempo que tarda la pelota en llegar al suelo asumiendo, primero, caída libre, esto es, que no hay ninguna fuerza de resistencia y, después, que dicha fuerza sí está presente y su valor es igual a dos veces el cuadrado de la velocidad de la pelota.

Si $y(t)$ denota la posición de la pelota con respecto al suelo en el instante de tiempo t y $g = 9.81\text{m/s}^2$ es la aceleración de la gravedad, la ecuación diferencial que verifica la aceleración $y''(t)$ es:

Caso I [Caída libre]: $y''(t) = -g$.

Caso II [Caída con resistencia del aire]: $y''(t) = -g + 2y'(t)^2$.

En ambas situaciones las condiciones iniciales son $y(0) = 5$, $y'(0) = 0$, pues la pelota parte del reposo desde 5 metros de altura.

Comenzamos analizando el primer caso.

Editor ► Caída libre

ejem15a.m

```
format long
opciones=odeset('events',@eventos);
[~,~,TE]=ode45(@fun,[0 Inf],[5 0],opciones);
disp('Tiempo de llegada al suelo:')
disp(TE)

function out=fun(~,y)
g=9.81;
out=[y(2);-g];
end

function [E,T,D]=eventos(~,y)
E=y(1);
T=1;
D=0;
end
```

Ejecutando el archivo obtenemos el tiempo de llegada al suelo.

Command Window

```
>> ejem15a
Tiempo de llegada al suelo
    1.009637554692332
```

Pasemos ahora al segundo caso.

Editor ► Caída con resistencia del aire

ejem15b.m

```
format long
opciones=odeset('events',@eventos);
[~,~,TE]=ode45(@fun,[0 Inf],[5 0],opciones);
disp('Tiempo de llegada al suelo:')
disp(TE)

function out=fun(~,y)
g=9.81;
out=[y(2);-g+2*y(2)^2];
end

function [E,T,D]=eventos(~,y)
E=y(1);
T=1;
D=0;
end
```

Ejecutando el archivo, observamos cómo la pelota tarda ahora más tiempo en llegar al suelo.

Command Window

```
>> ejem15b
Tiempo de llegada al suelo
2.414011614493581
```

Problema 1.2

El movimiento de una partícula en el plano está modelado por el sistema de ecuaciones diferenciales

$$\begin{cases} y_1'' = -2y_1 + y_2, \\ y_2'' = y_1 - 2y_2, \end{cases}$$

donde $[y_1(t), y_2(t)]^T$ indica la posición de la partícula en el instante t . La partícula, situada en el punto $[1, 0]^T$ en el instante inicial $t = 0$, parte del reposo. Escribe el problema de valores iniciales anterior como un sistema de 4 ecuaciones de primer orden.

1. Usando la orden de MATLAB® *ode45* calcula la posición de la partícula a los 6 segundos.
2. Representa gráficamente la órbita, esto es, la solución proyectada sobre el plano (y_1, y_2) , para $t \in [0, 100]$.
3. Calcula todos los valores de $t \in [0, 6]$ para los que se verifica $y_1(t) = -0.4$. Calcula asimismo los mínimos de la función y_1 en el intervalo $[0, 6]$. ¿Podrías obtener también los máximos de dicha función?

4. Es probable que hayas realizado el apartado anterior utilizando diferentes funciones evento. Haz de nuevo el apartado 3, pero esta vez utiliza una única función evento.

2. Ecuaciones diferenciales ordinarias: problemas de contorno

Los problemas de contorno más clásicos para ecuaciones diferenciales ordinarias vienen dados por una ecuación diferencial lineal de segundo orden para la que, en un intervalo $[a, b]$, se busca aquella solución que verifica ciertas condiciones en los puntos a y b (llamadas condiciones de contorno).

En concreto, el **problema de contorno lineal** se expresa como

$$\begin{cases} y''(t) = p(t)y'(t) + q(t)y(t) + r(t), \\ c_1y(a) + c_2y'(a) = \alpha, \quad d_1y(b) + d_2y'(b) = \beta, \end{cases}$$

donde $t \in [a, b]$, p, q, r son funciones continuas en $[a, b]$ y $c_1, c_2, d_1, d_2, \alpha, \beta$ son constantes reales. Observemos que si el problema fuera de valores iniciales, solo se impondrían condiciones en a .

En general, los problemas de contorno se formulan para sistemas de ecuaciones diferenciales de m ecuaciones, es decir,

$$y'(t) = f(t, y(t)),$$

donde $f : [a, b] \times \mathbb{R}^m \rightarrow \mathbb{R}^m$.

En este caso, se imponen $N_1 \geq 1$ condiciones en el punto a y $N_2 \geq 1$ en el punto b , de modo que $N_1 + N_2 = m$. Las condiciones de contorno en el extremo a (e igualmente en el extremo b) suelen ser, esencialmente, de tres tipos:

- i) Condición de tipo Dirichlet: $y(a) = \alpha$, donde $\alpha \in \mathbb{R}$.
- ii) Condición de tipo Neumann: $y'(a) = \alpha$, donde $\alpha \in \mathbb{R}$.
- iii) Condición mixta o de tipo Robbin: $c_1y(a) + c_2y'(a) = \alpha$, donde $c_1, c_2, \alpha \in \mathbb{R}$.

En cualquier caso, en problemas de contorno donde f es fuertemente no lineal, es habitual que las condiciones de contorno sean, de hecho, no lineales, en el sentido de que vengan dadas por cierta expresión del tipo

$$G(y(a), y(b), y'(a), y'(b)) = 0.$$

Respecto a la cuestión de existencia y unicidad de soluciones, la casuística de los problemas de contorno es bastante más compleja que la correspondiente a los problemas de valores iniciales. De hecho, no es difícil mostrar problemas de contorno, aparentemente muy sencillos, sin solución, o con un número finito de soluciones distintas, o incluso con un número infinito de soluciones distintas.

Ejemplo 2.1 ► Problema de contorno lineal

Consideremos el problema de contorno lineal

$$\begin{cases} y'' + y = t + 2a \operatorname{sen} t, \\ y(0) = 0, \quad y(\pi) = 0, \end{cases}$$

donde $t \in [0, \pi]$ y $a \in \mathbb{R}$. Vamos a comprobar que el valor de a determina el número de soluciones del problema. Es fácil deducir que la solución general de la ecuación homogénea asociada,

$$y'' + y = 0,$$

es

$$y(t) = \alpha_1 \operatorname{sen} t + \alpha_2 \cos t, \quad \alpha_1, \alpha_2 \in \mathbb{R},$$

y que una solución particular de la ecuación completa es

$$y(t) = t - at \cos t.$$

Por tanto, la solución general de la ecuación completa es

$$y(t) = t - at \cos t + \alpha_1 \operatorname{sen} t + \alpha_2 \cos t, \quad \alpha_1, \alpha_2 \in \mathbb{R}.$$

Vamos a imponer ahora las condiciones de contorno:

- i) **Condición de contorno en 0:** $0 = y(0) = \alpha_2$, es decir, $\alpha_2 = 0$.
- ii) **Condición de contorno en π :** $0 = y(\pi) = \pi + a\pi - \alpha_2 = (1 + a)\pi$, es decir, $(1 + a)\pi = 0$.

Se presentan dos casos:

- Si $a \neq -1$, no se verifica la segunda condición de contorno y, por tanto, **no existe solución**.
- Si $a = -1$, se verifica la segunda condición de contorno para todo $\alpha_1 \in \mathbb{R}$, de manera que **hay infinitas soluciones**, dadas por $y(t) = t + t \cos t + \alpha_1 \operatorname{sen} t$, para $\alpha_1 \in \mathbb{R}$.

2.1. El método de disparo lineal

Un ejemplo paradigmático de problema de contorno y que aparece reiteradamente en áreas técnicas y científicas es

$$\begin{cases} y'' = py' + qy + r, \\ y(a) = \alpha, \quad y(b) = \beta, \end{cases} \quad (4)$$

donde $t \in [a, b]$, p, q, r son funciones continuas en $[a, b]$ y α, β son constantes. Fijándonos exclusivamente en las condiciones de contorno, el problema (4) es un problema de tipo Dirichlet (en ambos extremos). Por otro lado, desde el punto de vista de la ecuación diferencial, el

problema (4) es un problema lineal regular. Si las funciones p, q, r son continuas únicamente en (a, b) , se habla de problemas lineales singulares.

Supongamos que el problema (4) tiene solución única y definida en todo $[a, b]$. La idea del **método de disparo lineal** es expresar y como combinación lineal de dos soluciones de sendos problemas de valores iniciales. En concreto, consideremos la familia uniparamétrica dada por

$$y(t) = y_p(t) + \gamma y_0(t), \quad \gamma \in \mathbb{R}, \quad (5)$$

siendo y_p e y_0 las soluciones respectivas de los problemas de valores iniciales

$$\begin{cases} y_p'' = py_p' + qy_p + r, \\ y_p(a) = \alpha, \\ y_p'(a) = 0, \end{cases} \quad \begin{cases} y_0'' = py_0' + qy_0, \\ y_0(a) = 0, \\ y_0'(a) = 1. \end{cases}$$

Por construcción, todos los miembros de la familia (5) verifican la ecuación diferencial del problema (4), así como la condición de contorno en el extremo a . El valor del parámetro γ se elige de forma que y verifique la condición de contorno en el extremo b , esto es, $y(b) = \beta$. En consecuencia,

$$\gamma = \frac{\beta - y_p(b)}{y_0(b)}.$$

Observemos que la hipótesis de que (4) tiene solución única implica que $y_0(b) \neq 0$.

Puede comprobarse que la función y es también solución del problema de valores iniciales

$$\begin{cases} y'' = py' + qy + r, \\ y(a) = \alpha, \\ y'(a) = \gamma. \end{cases}$$

Si pensamos en la solución de esta ecuación como la trayectoria que sigue un proyectil lanzado desde el punto (a, α) , lo que estamos pidiendo es que, al fijar la «elevación del cañón» mediante el valor $y'(a) = \gamma$, el proyectil llegue a su objetivo (b, β) ; de ahí el nombre de método de disparo.

Usando la orden `ode45` con las especificaciones por defecto, una posible implementación del método de disparo lineal aplicado al problema (4) podría ser:

Editor ► Método de disparo lineal

disparolineal.m

```
function [T,Y]=disparolineal(funcom,funhom,a,b,alpha,beta,h)
T=(a:h:b)';
[T,Y1]=ode45(funcom,T,[alpha;0]);
[T,Y2]=ode45(funhom,T,[0;1]);
Y=Y1(:,1)+((beta-Y1(end,1))/Y2(end,1))*Y2(:,1);
end
```

El argumento `funcom` representa la ecuación (completa) del problema (esto es, $y'' = py' + qy + r$) y el argumento `funhom`, la ecuación homogénea asociada (esto es, $y'' = py' + qy$). Asimismo, `h` indica el tamaño fijo de paso para obtener la partición donde se va a estimar la solución del problema de contorno.

Ejercicio 2.1

Considera el problema de contorno lineal

$$\begin{cases} y'' = 4y + 2t, \\ y(0) = 1, \quad y(3) = 2, \end{cases}$$

cuya solución exacta es

$$y(t) = -\frac{t}{2} + Ae^{2t} + Be^{-2t}, \quad \text{donde} \quad A = \frac{\frac{7}{2} - e^{-6}}{e^6 - e^{-6}}, \quad B = \frac{e^6 - \frac{7}{2}}{e^6 - e^{-6}}.$$

Tomando una partición de paso $h = 0.01$, haz una representación gráfica conjunta, en el intervalo $[0, 3]$, de las soluciones exacta y aproximada del problema. Calcula el error cometido sobre dicha partición.

2.2. El método de disparo no lineal

Consideremos el **problema de contorno general** con condiciones de Dirichlet

$$\begin{cases} y'' = f(t, y, y'), \\ y(a) = \alpha, \quad y(b) = \beta. \end{cases} \quad (6)$$

Supongamos que este problema tiene solución definida en todo $[a, b]$.

Siguiendo con el planteamiento del método de disparo lineal, consideremos la familia de problemas de valores iniciales

$$P(s) \equiv \begin{cases} y'' = f(t, y, y'), \\ y(a) = \alpha, \\ y'(a) = s, \end{cases}$$

donde s es un parámetro real.

Supongamos que existe un valor s_0 de modo que la solución de (6) es también la solución del problema de valores iniciales $P(s_0)$. Es más, lo que suele ocurrir en la práctica es que existe todo un intervalo I (conteniendo a s_0) donde, para todo $s \in I$, el problema $P(s)$ tiene solución definida en todo $[a, b]$.

Denotando por $y(t, s)$ la solución de $P(s)$, se tiene que $y(t, s_0)$ es la solución del problema de contorno (6). Puesto que $y(b, s_0) = \beta$, deducimos que el valor s_0 resuelve la ecuación no lineal

$$y(b, s) = \beta, \quad s \in I.$$

Esta observación es el punto de partida del **método de disparo no lineal**³. En concreto, se considera la función $F : I \rightarrow \mathbb{R}$, dada por

$$F(s) = y(b, s) - \beta$$

³Se trata, en realidad, de un método de disparo general, pero se le califica de «no lineal» al ser un procedimiento específicamente concebido para resolver problemas de contorno no lineales.

y se plantea la determinación de algún cero de F en I . Finalmente, la solución del problema de contorno original se obtiene a través del correspondiente problema de valores iniciales asociado a dicho cero.

No está de más subrayar que la evaluación funcional $F(s)$ requiere la resolución numérica del problema de valores iniciales $P(s)$. La estrategia que acabamos de exponer (para condiciones de tipo Dirichlet) puede adaptarse con facilidad a todo tipo de condiciones de contorno. Para resolver la ecuación no lineal anterior puede utilizarse cualquiera de los métodos vistos en la lección 5.

Siguiendo paso a paso las explicaciones teóricas previas, un código para resolver (6) mediante el método de disparo no lineal (combinando las órdenes de MATLAB® `fzero` y `ode45` con las especificaciones por defecto) sería

Editor ► Método de disparo no lineal

disparonolineal.m

```
function [s1,T,Y]=disparonolineal(s0,f,a,b,alpha,beta)
s1=fzero(@(s)g(s,f,a,b,alpha,beta),s0);
[T,Z]=ode45(f,[a b],[alpha s1]);
Y=Z(:,1);
end

function y=g(s,f,a,b,alpha,beta)
[~,Y]=ode45(f,[a b],[alpha s]);
y=Y(end,1)-beta;
end
```

Ejercicio 2.2

Usando el método de disparo no lineal, dibuja la solución del problema de contorno no lineal

$$\begin{cases} 8y'' = 32 + 2t^3 - yy', \\ y(1) = 17, \quad y(3) = \frac{43}{3}. \end{cases}$$

Asimismo, verifica gráficamente la bondad de la aproximación obtenida, sabiendo que el problema anterior tiene solución única dada por

$$y(t) = t^2 + \frac{16}{t}.$$

Proyecto 2.1 (solución en página 31)

El problema de Bratu

$$\begin{cases} y'' + \lambda e^y = 0, \\ y(0) = 0, \quad y(1) = 0, \end{cases}$$

tiene dos soluciones distintas para $\lambda \in (0, \lambda_0)$, donde $\lambda_0 \approx 3.5$, una única solución si $\lambda = \lambda_0$ y ninguna si $\lambda > \lambda_0$. Calcula las dos soluciones que existen para $\lambda = 1$ y dibújalas en una misma ventana gráfica.

3. Apéndice A: Introducción a las ecuaciones en derivadas parciales

Se denominan ecuaciones en derivadas parciales a aquellas ecuaciones diferenciales en las que la función incógnita depende de más de una variable independiente.

Vamos a considerar una de las ecuaciones en derivadas parciales más simple que existe: la ecuación de transporte, dada por

$$u_t(x, t) + c(x)u_x(x, t) = 0,$$

donde $c : \mathbb{R} \rightarrow \mathbb{R}$ es una función continua y no negativa en $[0, +\infty)$.

Consideremos el problema de valores iniciales

$$(P) \begin{cases} u_t(x, t) + c(x)u_x(x, t) = 0, & x \in \mathbb{R}, t > 0, \\ u(x, 0) = f(x), & x \in \mathbb{R}, \end{cases}$$

donde $f : \mathbb{R} \rightarrow \mathbb{R}$ es una función continua.

Supondremos que el problema anterior tiene solución única u definida y continua en $\mathbb{R} \times [0, +\infty)$ y con derivadas parciales de primer orden continuas en $\mathbb{R} \times (0, +\infty)$.

Veamos cómo resolver numéricamente el problema (P) en un rectángulo $[0, L] \times [0, T]$, donde $L, T > 0$. Para ello, aplicamos el denominado método de diferencias finitas a la banda $R := \mathbb{R} \times [0, T]$. El primer paso es establecer un mallado en R . En concreto, fijados dos números naturales n y m , ambos mayores que dos, definimos el paso espacial

$$h := \frac{L}{n}$$

y el paso temporal

$$k := \frac{T}{m}$$

y consideramos el mallado de R dado por los nodos

$$(x_i, t_j) := (ih, jk), \quad i \in \mathbb{Z}, \quad j = 0, 1, \dots, m.$$

Para cada j , el conjunto $\{(x_i, t_j) : i \in \mathbb{Z}\}$ suele denominarse fila j del mallado.

El objetivo es obtener aproximaciones de los valores de la solución en cada uno de los nodos que forman el mallado. Llamamos $w_{i,j}$ al valor aproximado de la solución en el punto (x_i, t_j) , esto es, $w_{i,j} \approx u(x_i, t_j)$. Conviene observar que la condición inicial nos indica que

$$w_{i,0} \approx u(x_i, 0) = f(x_i), \quad i \in \mathbb{Z}.$$

En otras palabras, conocemos la estimaciones de la fila cero del mallado. Además, denotemos $c_i = c(x_i)$, para $i \in \mathbb{Z}$.

El primer paso del método es aproximar las derivadas parciales mediante fórmulas de derivación numérica. La opción más habitual es elegir la fórmula de la diferencia regresiva⁴

⁴La fórmula de la diferencia regresiva para la derivada primera viene dada por

$$f'(x_0) \approx \frac{f(x_0) - f(x_0 - h)}{h},$$

donde f es una función derivable en x_0 y $h > 0$.

para estimar u_x y la fórmula de la diferencia progresiva para estimar u_t . Es decir,

$$u_x(x, t) \approx \frac{u(x, t) - u(x - h, t)}{h},$$

$$u_t(x, t) \approx \frac{u(x, t + k) - u(x, t)}{k}.$$

Evaluando u_x y u_t en los puntos del mallado, sustituyendo en (P) las fórmulas de diferencias anteriores y eliminando los términos de error se obtiene

$$\frac{w_{i,j+1} - w_{i,j}}{k} + c_i \frac{w_{i,j} - w_{i-1,j}}{h} = 0, \quad i \in \mathbb{Z}, \quad j = 0, 1, \dots, m-1.$$

Denotando $\lambda = \frac{k}{h}$, se deduce la ecuación

$$w_{i,j+1} = (1 - c_i \lambda) w_{i,j} + c_i \lambda w_{i-1,j}, \quad i \in \mathbb{Z}, \quad j = 0, 1, \dots, m-1,$$

conocida como esquema de diferencias finitas.

Esta ecuación nos permite calcular las estimaciones de la solución en los nodos de la fila $j + 1$ del mallado suponiendo conocidas las estimaciones de la fila j . Observemos que las estimaciones de la fila primera en los nodos $\{(ih, k) : i = 0, 1, \dots, n\}$ requieren estimaciones de la fila cero en los nodos $\{(ih, 0) : i = -1, 0, \dots, n-1\}$. Este fenómeno de desplazamiento de índices se repite entre dos filas consecutivas cualesquiera y hay que tenerlo en cuenta a la hora de programar el método.

Como es lógico, se pretende que a medida que h y k tiendan a cero, la diferencia entre los valores aproximados y los valores exactos tienda a cero. Técnicamente, este hecho se denomina convergencia del esquema de diferencias finitas y para garantizar que esto ocurre es necesario imponer que $C\lambda \leq 1$, donde $C = \max\{c(x) : x \in [0, L]\}$.

Ejemplo 3.1 ► Método de diferencias finitas para la ecuación de transporte

El siguiente código de MATLAB® implementa el método de diferencias finitas descrito anteriormente para la resolución del problema (P) en un rectángulo

$$[0, L] \times [0, T],$$

donde $L, T > 0$, con paso espacial h y paso temporal k .

Llamando u_a y u_e a las soluciones aproximada y exacta, respectivamente, de (P), la parte final del código muestra una animación conjunta de las funciones

$$x \mapsto u_a(x, t), \quad x \mapsto u_e(x, t),$$

en el intervalo $[0, L]$ para los distintos instantes temporales determinados por el mallado.

Los argumentos `fun` y `c` representan las funciones f y c , respectivamente, y los parámetros `A` y `B` se utilizan para encuadrar verticalmente las gráficas de la animación.

Editor ► Método de diferencias finitas

metdiffin.m

```
function U=metdiffin(fun,c,h,k,L,T,A,B)
x=0:h:L;
t=0:k:T;
nx=length(x);
nt=length(t);
lambda=k/h;
p=nt-1+nx;
V=zeros(p,nt);
xa=-(nt-1)*h:h:L;
V(:,1)=fun(xa);
for i=2:nt
    coef=lambda*c(xa(i:p))';
    V(i:p,i)=(1-coef).*V(i:p,i-1)+coef.*V(i-1:p-1,i-1);
end
U=V(nt:nt-1+nx,:);
for k=1:nt
    EX=x.^2.*exp(-2*t(k));
    plot(x,U(:,k),'b',x,EX,'r'),axis([0 L A B]),shg,pause(0.01)
end
end
```

Consideremos el problema de valores iniciales para la ecuación de transporte

$$\begin{cases} u_t(x,t) + xu_x(x,t) = 0, & x \in \mathbb{R}, \quad t > 0, \\ u(x,0) = x^2, & x \in \mathbb{R}, \end{cases}$$

cuya solución exacta es

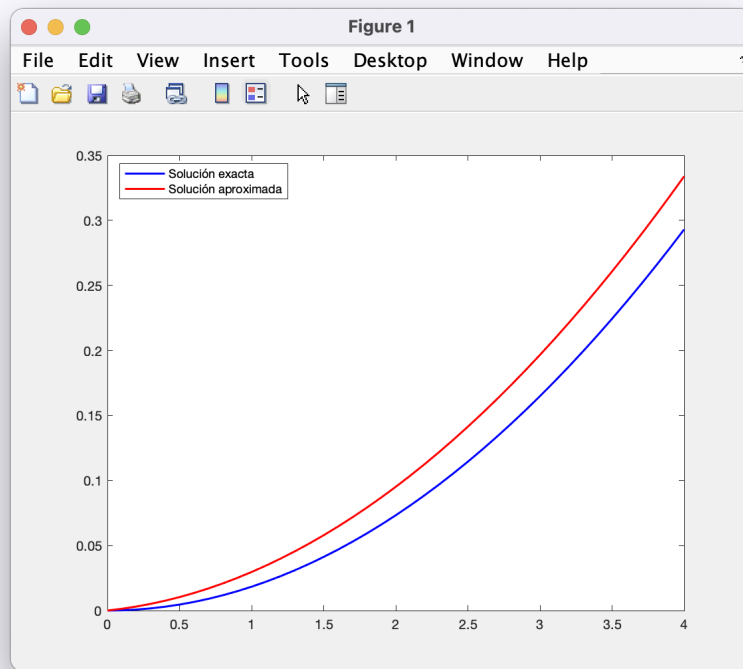
$$u(x,t) = x^2 e^{-2t}.$$

Tomando $L = 4$ y $T = 2$ y tamaños de paso espacial $h = 0.1$ y temporal $k = 0.005$ (de donde $n = 40$ y $m = 400$), vamos a hacer una representación gráfica conjunta de las soluciones exacta y aproximada para $t = 2$.

Observemos que $\lambda = 0.05$ y $C = \max\{x^2 : x \in [0, 4]\} = 16$, de donde $C\lambda = 0.8 \leq 1$. Por consiguiente, el esquema de diferencias finitas es convergente.

Command Window

```
>> f=@(x)x.^2;
>> c=@(x)x;
>> U=metdiffin(f,c,0.1,0.005,4,2,-0.5,2);
>> fex=@(x,t)x.^2.*exp(-2*t);
>> x=0:0.1:4;
>> plot(x,fex(x,2),'b',x,U(:,end),'r','LineWidth',1.5),shg,...
legend('Solución exacta','Solución aproximada','Location','Best');
```



4. Soluciones de los proyectos

Proyecto 1.1

1. Vamos a comprobar que el método `rk3` dado en el enunciado es convergente de orden dos, al menos.

Efectivamente, el método es convergente puesto que

$$b_1 + b_2 + b_3 = \frac{1}{4} + 0 + \frac{3}{4} = 1.$$

Su orden de convergencia es, al menos, dos, dado que, además de la condición anterior, se verifican

$$b_1 c_1 + b_2 c_2 + b_3 c_3 = \frac{1}{4} \cdot 0 + 0 \cdot \frac{1}{3} + \frac{3}{4} \cdot \frac{2}{3} = \frac{1}{2}$$

y

$$c_1 = 0 = a_{1,1}, \quad c_2 = \frac{1}{3} = a_{2,1}, \quad c_3 = \frac{2}{3} = a_{3,1} + a_{3,2} = 0 + \frac{2}{3}.$$

2. El siguiente código de MATLAB® implementa este método con partición de paso fijo.

Editor ► Método de Runge-Kutta de tres etapas

rk3.m

```
function [T,Y]=rk3(f,t0,tf,y0,h)
T=(t0:h:tf)';
n=length(T);
y0=y0(:)';
Y(1,:)=y0;
for k=1:n-1
    k1=f(T(k),Y(k,:))';
    k2=f(T(k)+h/3,Y(k,:)+h*k1/3)';
    k3=f(T(k)+2*h/3,Y(k,:)+2*h*k2/3)';
    Y(k+1,:)=Y(k,:)+h*(k1/4+3*k3/4);
end
end
```

3. Vamos a aproximar, mediante el método `rk3`, la solución del problema de valores iniciales

$$\begin{cases} y'' + 8y' + 16y = 8e^{-2t}, \\ y(0) = 2, \\ y'(0) = 0, \end{cases}$$

donde $t \in [0, 3]$. Tomaremos diferentes tamaños de paso, $h = 10^{-1}, 10^{-2}, 10^{-3}, 10^{-4}$, con el objeto de estudiar la evolución del error cuando h tiende a cero.

Puedes comprobar que la solución exacta del problema anterior es

$$y(t) = 4te^{-4t} + 2e^{-2t}.$$

En primer lugar, hacemos el cambio de variables

$$\begin{aligned}y_1 &= y, \\ y_2 &= y',\end{aligned}$$

para transformar la ecuación diferencial de segundo orden en un sistema de ecuaciones diferenciales de primer orden. El nuevo problema de valores iniciales es

$$\begin{cases} y_1' = y_2, \\ y_2' = -16y_1 - 8y_2 + 8e^{-2t}, \\ y_1(0) = 2, \\ y_2(0) = 0. \end{cases}$$

El siguiente código de MATLAB® aproxima la solución de este problema para los distintos tamaños de paso h antes mencionados, calcula el error cometido para cada h y muestra los resultados en una tabla.

Editor ► Convergencia y orden del método rk3

proyecto11.m

```
f=@(t,y)[y(2);-16*y(1)-8*y(2)+8*exp(-2*t)];
for k=1:4
    h=10^(-k);
    [T,Y]=rk3(f,0,3,[2 0],h);
    EX=4*T.*exp(-4*T)+2*exp(-2*T);
    err=max(abs(EX-Y(:,1)));
    fprintf('h=10^(-%1d)    error=%.4e \n',k,err)
end
```

Su ejecución en MATLAB® genera los siguientes resultados:

Command Window

```
>> proyecto11
h=10^(-1)    error=4.5361e-03
h=10^(-2)    error=3.0740e-06
h=10^(-3)    error=2.9553e-09
h=10^(-4)    error=2.9439e-12
```

Observa dos fenómenos:

- El primero es la convergencia. Fíjate en cómo, a medida que el tamaño de paso decrece, el error disminuye. Esto significa que podemos aproximar la solución del problema de valores iniciales por medio del método **rk3** con todo el nivel de precisión que queramos, sin otro límite que el que impone la precisión de la máquina.
- El segundo es el orden de convergencia, esto es, la «rapidez» con la que el error se aproxima a cero. Nota cómo, al dividir el tamaño paso por 10, el error se divide (aproximadamente) por 10^3 . En consecuencia, el error es del orden de h^3 , cuando h tiende a cero, es decir, la convergencia del método **rk3** es de orden 3.

Proyecto 1.2

En primer lugar, hacemos el cambio de variables

$$\begin{aligned}y_1 &= y, \\ y_2 &= y',\end{aligned}$$

para transformar la ecuación diferencial de segundo orden en un sistema de ecuaciones diferenciales de primer orden. El nuevo problema de valores iniciales consiste en el sistema de ecuaciones diferenciales

$$\begin{cases} y_1' = y_2, \\ y_2' = \mu(1 - y_1^2)y_2 - y_1, \end{cases}$$

junto con las condiciones iniciales

$$\begin{bmatrix} y_1(0) \\ y_2(0) \end{bmatrix} = \begin{bmatrix} 2 \\ 0 \end{bmatrix}.$$

El siguiente código de MATLAB® aproxima la solución de este problema para $\mu = 0.2$ y $t \in [0, 20]$ y la dibuja en los planos (t, y) e (y, y') (es decir, en el plano de las nuevas variables (y_1, y_2)).

Editor ► Ecuación diferencial con parámetro

proyecto12.m

```
f=@(t,y,mu)[y(2);mu*(1-y(1)^2)*y(2)-y(1)];

opciones=odeset('AbsTol',1e-8,'RelTol',1e-8);
[T,Y]=ode45(@(t,y)f(t,y,0.2),[0 20],[2 0],opciones);

figure
plot(T,Y(:,2)),shg

figure
plot(Y(:,1),Y(:,2)),shg
```

Fíjate en que la función f depende no solo de las variables t e y , sino también de una tercera variable, el parámetro μ . Cuando la orden `ode45` llama a f , es necesario especificar las variables utilizadas en la integración de la ecuación diferencial, esto es, t e y ,

$$\text{@}(t,y)f(t,y,0.2)$$

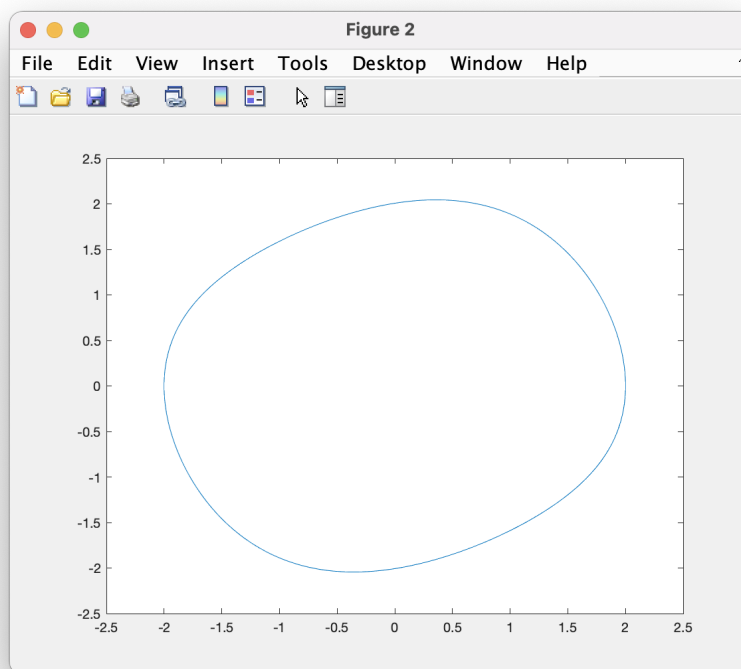
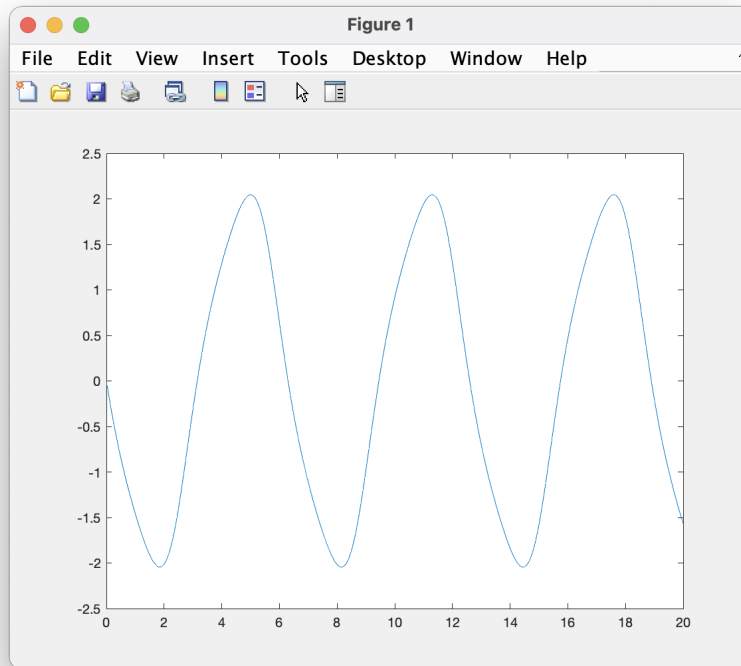
y asignar un valor específico al parámetro μ ; en este caso, $\mu = 0.2$.

Ejecutamos este código en MATLAB® y observamos las dos imágenes generadas.

La particularidad que tiene la solución del problema de valores iniciales es su carácter periódico.

Command Window

>> proyecto12



Proyecto 1.3

La integral definida que debemos aproximar está vinculada a la solución de un problema de valores iniciales. Por ello, la calcularemos mediante la orden de MATLAB® `ode45` y no por medio de órdenes asociadas a la cuadratura numérica.

Considera la función

$$w(t) = \int_0^t v^2(s) ds, \quad t \geq 0.$$

Observa que $w(0) = 0$ y que

$$w(20) = \int_0^{20} v^2(s) ds.$$

Es decir, $w(20)$ es, precisamente, la integral que nos piden calcular. Esto sugiere que consideremos el problema de valores iniciales

$$\begin{cases} u' = 1 + u^2v - 4u, \\ v' = 3u - u^2v, \\ w' = v^2, \end{cases}$$

junto con las condiciones iniciales

$$u(0) = 1.5, \quad v(0) = 3, \quad w(0) = 0.$$

A continuación, resolvemos este problema de valores iniciales con la orden de MATLAB® `ode45` en el intervalo $[0, 20]$ y así obtenemos una aproximación de la solución en $t = 20$.

Operamos en MATLAB®:

Command Window

```
>> format long
>> f=@(t,y)[1+y(1)^2*y(2)-4*y(1); 3*y(1)-y(1)^2*y(2);y(2)^2];
>> [~,Y]=ode45(f,[0 20],[1.5 3 0]);
>> I=Y(end,3)
```

I =

2.223842929591794e+02

Fíjate en que $Y(\text{end},3)$ es el valor de la integral que nos pedían calcular, puesto que la última fila de la matriz Y corresponde a la solución en el tiempo final, $t = 20$, y su tercera columna indica la tercera componente de la solución, esto es, w .

Proyecto 1.4

Consideremos la ecuación no lineal $F(\alpha) = 0$, donde $F(\alpha) = y'(1.2, \alpha) - 0.5$, siendo $y(t, \alpha)$ la solución del problema de valores iniciales

$$\begin{cases} y'' = \cos y, \\ y(0) = \alpha, \\ y'(0) = 0. \end{cases}$$

El siguiente código de MATLAB® permite encontrar un cero de la ecuación no lineal $F(\alpha) = 0$ utilizando la orden `fzero`.

Editor ► Ecuación diferencial y ecuación no lineal

proyecto14.m

```
alpha=fzero(@F,0.1);
disp(alpha)

function y=F(alpha)
[~,Y]=ode45(@f,[0 1.2],[alpha 0]);
y=Y(end,2)-0.5;
end

function out=f(~,y)
out=[y(2);cos(y(1))];
end
```

Ejecutamos este código:

Command Window

```
>> format long
>> proyecto14
1.020992309776808
```

Vamos a comprobar que, efectivamente, para el valor de α calculado se verifica que $y'(1.2) = 0.5$:

Command Window

```
>> format short e
>> f=@(t,y)[y(2);cos(y(1))];
>> [T,Y]=ode45(f,[0 1.2],[alpha 0]);
>> abs(0.5-Y(end,2))

ans =

1.1102e-16
```

Proyecto 2.1

Aplicamos la función que implementa el método de disparo no lineal. Puedes encontrarla al final de la sección 2.2. Tomamos dos valores del punto inicial s_0 para obtener las dos soluciones y, a continuación, las dibujamos:

Command Window

```
>> f=@(t,y)[y(2);-exp(y(1))];  
>> s0=1;  
>> [s1,T1,Y1]=disparonolineal(s0,f,0,1,0,0);  
>> s0=10;  
>> [s1,T2,Y2]=disparonolineal(s0,f,0,1,0,0);  
>> plot(T1,Y1,'b',T2,Y2,'r'),shg
```

