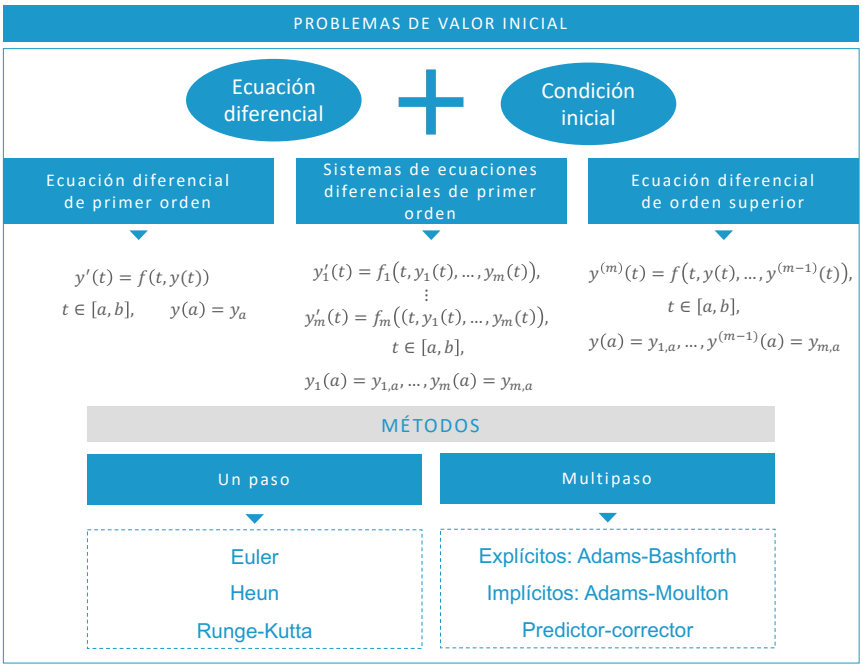
Métodos Numéricos Aplicados I

Problemas de valor inicial I

Índice

Esquema


PROBLEMAS DE VALOR INICIAL
Ecuación diferencial
Condición inicial
Ecuación diferencial de primer orden
Sistemas de ecuaciones diferenciales de primer orden
Ecuación diferencial de orden superior
$y'(t) = f(t, y(t))$
$t \in [a, b], \quad y(a) = y_a$
$y_1'(t) = f_1(t, y_1(t), \dots, y_m(t)),$
$y_m'(t) = f_m((t, y_1(t), \dots, y_m(t)),$
$t \in [a, b],$
$y_1(a) = y_{1,a}, \dots, y_m(a) = y_{m,a}$
$y^{(m)}(t) = f(t, y(t), \dots, y^{(m-1)}(t)),$
$t \in [a, b],$
$y(a) = y_{1,a}, \dots, y^{(m-1)}(a) = y_{m,a}$
MÉTODOS
Un paso
Multipaso
Euler
Heun
Runge-Kutta
Explícitos: Adams-Bashforth
Implícitos: Adams-Moulton
Predictor-corrector

Ideas clave

6.1 Introducción y objetivos

Las ecuaciones diferenciales permiten modelar matemáticamente multitud de fenómenos físicos, mecánicos, biológicos o económicos, entre otros. La trayectoria de una partícula, la evolución de la temperatura, el comportamiento de un circuito eléctrico o los procesos migratorios, son ejemplos de sistemas dinámicos que se describen utilizando estas ecuaciones.

En este tema y el siguiente vamos a estudiar un caso particular de ecuaciones y sistemas de ecuaciones diferenciales, denominados problemas de valor inicial. En este tipo de problemas, vamos a conocer el modelo y una característica particular, que es cómo se encuentra la magnitud bajo estudio en el instante inicial. A partir de esta información, describiremos una serie de métodos numéricos que, lejos de obtener la solución analítica al problema de valor inicial, van a obtener una aproximación precisa bajo una serie de condiciones.

Veamos a continuación como ejemplo de problema de valor inicial un modelo muy utilizado en el contexto biológico. En el estudio del crecimiento poblacional, uno de los modelos más extendidos es el modelo de Malthus, el cual establece que la velocidad de crecimiento de una población es proporcional al número de individuos presentes en cada instante de tiempo. Siendo k el coeficiente de proporcionalidad e $y(t)$ el número de individuos de la población en el instante t, este crecimiento se modeliza por medio de la ecuación diferencial

$$y'(t) = ky(t).$$

En particular, el modelo obtenido con $k < 0$ describe habitualmente procesos de

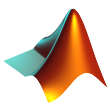
desintegración radiactiva. Asimismo, el modelo logístico o de Verhulst introduce una modificación en la ley de Malthus para tener en cuenta que los recursos que tiene la población pueden ser limitados. Teniendo en cuenta estas limitaciones, se añade a la ecuación un término no lineal que a su vez sirve para frenar este crecimiento, de modo que el modelo de Verhulst se describe con la ecuación

$$y'(t) = ky(t) - p(y(t))^2.$$

Con los métodos que vamos a ver a lo largo de estos dos temas y conociendo el número de individuos de la población en el instante inicial $y(0) = y_0$ seremos capaces de obtener una solución numérica para determinar el número de inviduos a lo largo del tiempo.

Los métodos numéricos los podemos clasificar en métodos de un paso y métodos multipaso. A lo largo del tema trabajaremos sobre el primer tipo de métodos, mientras que en el siguiente tema abordaremos los métodos multipaso. Por tanto, los objetivos de este tema son los siguientes:

- Definir los problemas de valor inicial y los distintos esquemas de ecuaciones diferenciales que los definen.
- Comprender la relación entre solución analítica y discreta de un PVI.
- Conocer procesos distintos para diseñar métodos numéricos y el error cometido en el cálculo de la solución de un PVI.
- Estudiar los métodos numéricos de Euler, Heun y Runge-Kutta así como su orden de convergencia obtenido de forma numérica.



Algunas funciones Matlab utilizadas en este tema

- `ode23`: resuelve EDO's y sistemas de EDO's

6.2 Problemas de valor inicial

Los problemas de valor inicial se pueden presentar a partir de diferentes esquemas de ecuaciones diferenciales. En primer lugar, mostramos aquellos que se basan en una única ecuación diferencial de primer orden. A continuación, presentaremos aquellos esquemas formados por diversas ecuaciones diferenciales de primer orden en los que se expresa la variación de más de una magnitud. Por último, si en una ecuación intervienen derivadas de órdenes superiores, tendremos otro tipo de problema de valor inicial. Veremos que estos problemas se traducen a sistemas de ecuaciones de primer orden.

PVI definidos por ecuaciones diferenciales de primer orden

Partiendo de una ecuación diferencial de primer orden que expresa la variación de la variable dependiente y con respecto a la variable independiente t:

$$y'(t) = f(t, y(t)), \qquad t \in [a, b],$$

el objetivo de un PVI es encontrar la solución $y(t)$ de la ecuación conociendo la condición inicial:

$$y(a) = y_a.$$

El siguiente resultado determina las condiciones que debe satisafacer la función f que describe el PVI para que éste tenga solución.

Teorema 1

Supongamos que $D = \{(t, y) : t \in [a, b], y \in \mathbb{R}\}$ y que $f(t, y)$ es continua en D. Si para todo par de puntos $(t_1, y_1), (t_2, y_2) \in D$ existe una constante $L > 0$ tal que f cumple la condición de Lipschitz:

$$|f(t_1, y_1) - f(t_2 - y_2)| \leq L|y_1 - y_2|,$$

entonces el PVI dado por

$$y'(t) = f(t, y(t)), \qquad t \in [a, b], \qquad y(a) = y_a,$$

tiene una solución única $y(t)$, $t \in [a, b]$.

Existen tanto técnicas analíticas o teóricas, como técnicas numéricas para obtener la solución a este tipo de probemas. Las técnicas analíticas permiten determinar la existencia de la solución del problema e incluso las condiciones para que sea única. En cambio, las técnicas numéricas proporcionan aproximaciones a la solución cuando no es posible calcularla de forma analítica o el proceso es complejo.

Hasta hace unas décadas, la solución continua $y(t)$ de un PVI se obtenía exclusivamente por medio de técnicas analíticas, siendo el proceso analítico de resolución diferente en función del tipo de ecuación. Sin embargo, la irrupción de las computadoras en el cálculo matemático ha permitido el uso de las técnicas numéricas, las cuales obtienen como aproximación a $y(t)$ una solución discreta $y_k \approx y(t_k)$ del PVI, donde t_k, $k = 0, 1, 2, \ldots, N$, son los nodos de la discretización.

Para realizar la discretización del problema, dividimos el intervalo $[a, b]$ en N subintervalos $[t_k, t_{k+1}]$ cuyos extremos son los nodos equiespaciados

$$t_k = a + kh, \qquad k = 0, 1, \ldots, N,$$

siendo h el paso de la partición. La discretización de la variable independiente se ilustra en la Figura 1. Obtendremos la solución numérica en los nodos t_k, de forma que $y_k \approx y(t_k)$.

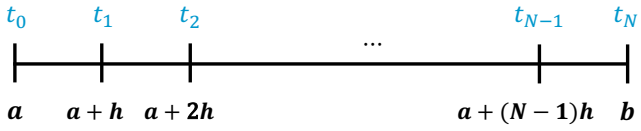


Figura 1: Proceso de discretización

Tras la discretización de la variable independiente, mostramos en el Ejemplo 1 cómo resolver de forma genérica en Matlab un PVI de estas características. Lo haremos con el comando `ode23` de Matlab, el cual es un método numérico de orden bajo para resolver este tipo de problemas.

Ejemplo 1.

Sea el PVI definido por el siguiente caso particular del modelo de Verhulst

$$y'(t) = (3 - 0.1y(t))y(t), \qquad t \in [0, 2],$$

con una población inicial $y(0) = 10$. Resolveremos el PVI utilizando el comando de Matlab `ode23` con 30 subintervalos.

En primer lugar, definimos una función de Matlab que describe la ecuación diferencial:

```
function dy = VerhulstPVI(t,y)
    k = 3;
    p = 0.1;
    dy = (k-p*y).*y;
end
```

En segundo lugar, discretizamos la variable independiente en $[0, 2]$ tomando 30 subintervalos:

```
>> a=0; b=2; N=30;
>> h=(b-a)/N;
>> td=a:h:b;
```

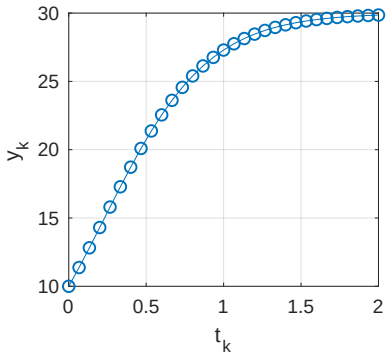
Por último, definimos la condición inicial y ejecutamos la función `ode23`:

```
>> ya=10;
```

```
>> [t,y]=ode23('VerhulstPVI',td,ya);
```

Representamos la solución obtenida:

```
>> plot(t,y,'o-')
```



PVI definidos por sistemas de ecuaciones diferenciales de primer orden

Un sistema de ecuaciones diferenciales de primer orden expresa las relaciones que verifican diversas funciones y sus derivadas de orden uno respecto a una variable independiente. Un sistema con m funciones incógnitas $y_1, y_2, \ldots, y_m$ y variable independiente t se expresa en forma general como

$$\left.\begin{array}{rcl} y_1'(t) & = & f_1(t, y_1(t), y_2(t), \ldots, y_m(t)), \\ y_2'(t) & = & f_2(t, y_1(t), y_2(t), \ldots, y_m(t)), \\ & \vdots & \\ y_m'(t) & = & f_m(t, y_1(t), y_2(t), \ldots, y_m(t)), \end{array}\right\}, \qquad t \in [a, b].$$

Tendremos un problema de valor inicial si conocemos el valor de las m funciones incógnita en el instante inicial:

$$y_1(a) = y_{1,a}, \quad y_2(a) = y_{2,a}, \quad \ldots, \quad y_m(a) = y_{m,a}.$$

Podemos representar este sistema en términos vectoriales utilizando una notación más compacta. De este modo, si denotamos las funciones incógnita como $Y : \mathbb{R} \longrightarrow \mathbb{R}^m$ tal que

$$Y(t) = [y_1(t), y_2(t), \ldots, y_m(t)]^T$$

y las ecuaciones como funciones coordenadas de una función vectorial de varias variables $F : \mathbb{R}^{m+1} \longrightarrow \mathbb{R}^m$:

$$F(t, Y(t)) = \begin{bmatrix} f_1(t, y_1(t), y_2(t), \ldots, y_m(t)) \\ f_2(t, y_1(t), y_2(t), \ldots, y_m(t)) \\ \vdots \\ f_m(t, y_1(t), y_2(t), \ldots, y_m(t)) \end{bmatrix}$$

podemos escribir el sistema como

$$Y'(t) = F(t, Y(t)),$$

siendo la condición inicial del PVI

$$Y_a = Y(a) = [y_{1,a}, y_{2,a}, \ldots, y_{m,a}]^T.$$

Para este tipo de problemas también obtendremos la solución en cada nodo siguiendo el mismo proceso de discretización de la variable independiente que en el caso de una ecuación diferencial de primer orden. De la misma forma, definiremos previamente la función de Matlab que describe el sistema de ecuaciones diferenciales. Resolveremos un problema de estas características con `ode23` tras describir los PVI obtenidos a partir de ecuaciones diferenciales de orden mayor que uno.

PVI definidos por ecuaciones diferenciales de orden superior

Se obtiene un problema de valor inicial distinto a los anteriores cuando intervienen en la ecuación derivadas de órdenes superiores. Las ecuaciones diferenciales de orden

m son ecuaciones diferenciales en las cuales intervienen la función incógnita y sus derivadas hasta orden m. Representamos de forma general estas ecuaciones como

$$y^{(m)}(t) = f\left(t, y(t), y'(t), \ldots, y^{(m-1)}(t)\right).$$

Para obtener un problema de valor inicial a partir de una ecuación diferencial de orden m, es necesario imponer m condiciones sobre la solución en el instante inicial. Por tanto, tendremos un PVI si cononemos los valores de la función incógnita y de sus derivadas hasta orden $m-1$ en el instante inicial, es decir,

$$y(a) = y_{1,a}, \quad y'(a) = y_{2,a}, \quad \ldots, \quad y^{(m-1)}(a) = y_{m,a}.$$

Para resolver el PVI obtenido, se realiza el cambio de variables

$$y_1(t) = y(t), \quad y_2(t) = y'(t), \quad \ldots, \quad y_m(t) = y^{(m-1)}(t),$$

de modo que la ecuación diferencial de orden m se convierte en el sistema de ecuaciones diferenciales

$$\begin{bmatrix} y_1(t) \\ y_2(t) \\ y_3(t) \\ \vdots \\ y_m(t) \end{bmatrix} = \begin{bmatrix} y(t) \\ y'(t) \\ y''(t) \\ \vdots \\ y^{(m-1)}(t) \end{bmatrix} \Rightarrow \begin{bmatrix} y_1'(t) \\ y_2'(t) \\ y_3'(t) \\ \vdots \\ y_m'(t) \end{bmatrix} = \begin{bmatrix} y'(t) \\ y''(t) \\ y'''(t) \\ \vdots \\ y^{(m)}(t) \end{bmatrix} = \begin{bmatrix} y_2(t) \\ y_3(t) \\ y_4(t) \\ \vdots \\ f\left(t, y_1(t), \ldots, y_m(t)\right) \end{bmatrix}$$

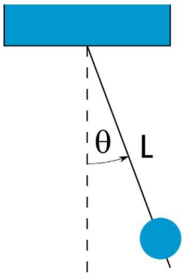
con condiciones iniciales

$$y_1(a) = y_{1,a}, \quad y_2(a) = y_{2,a}, \quad \ldots, \quad y_m(a) = y_{m,a},$$

y cuyo proceso de resolución se ha descrito anteriormente. Por tanto, obtendremos como solución del PVI una matriz con m columnas, correspondientes a las incógnitas $y_1(t), y_2(t), \ldots, y_m(t)$. La solución de la ecuación diferencial de orden m será únicamente la primera columna de esta matriz $y_1(t) = y(t)$.

Ejemplo 2.

Consideremos un péndulo de longitud $L = 1$



cuya ecuación del movimiento viene descrita por

$$\theta''(t) - \frac{g}{L} \sin \theta(t) = 0,$$

donde $g = 9.8\, m/s^2$ es la aceleración de la gravedad. Soltamos el péndulo en el instante inicial $t = 0$, de forma que en ese instante el ángulo es de $\theta(0) = \pi/6$ radianes y la velocidad angular es $\theta'(0) = 0\, m/s$.

Obtendremos la solución discreta tomando como paso $h = 0.1$. Para resolver el PVI obtenido por medio de una ecuación diferencial de orden dos, transformaremos en primer lugar el problema en el siguiente sistema de dos ecuaciones diferenciales de primer orden:

$$\begin{bmatrix} y_1(t) \\ y_2(t) \end{bmatrix} = \begin{bmatrix} \theta(t) \\ \theta'(t) \end{bmatrix} \Leftrightarrow \begin{bmatrix} y_1'(t) \\ y_2'(t) \end{bmatrix} = \begin{bmatrix} \theta'(t) \\ \theta''(t) \end{bmatrix} = \begin{bmatrix} y_2(t) \\ \frac{g}{L} \sin y_1(t) \end{bmatrix},$$

sujeto a las condiciones iniciales

$$y_1(0) = \frac{\pi}{6}, \qquad y_2(0) = 0.$$

Implementamos la función de Matlab que define el sistema:

```
function dY = PenduloPVI(t,theta)
```

```
    g=9.8; L=1;
    Y1=theta(1); Y2=theta(2);
    dY=[Y2; (g/L)*sin(Y1)];
end
```

Definimos la variable independiente discretizada, tomando el paso indicado:

```
>> a=0; b=2; h=0.1;
>> td=a:h:b;
```

Por último, definimos el vector de condiciones iniciales y ejecutamos el comando ode23:

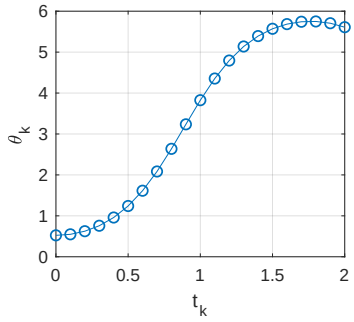
```
>> Ya=[pi/6; 0];
>> [t,Y]=ode23('PenduloPVI',td,Ya);
```

Mostramos a continuación los valores obtenidos para la solución del PVI en los instantes $t \in \{0, 0.5, 1, 1.5, 2\}$:

t	$\theta(t)$
0	0.5236
0.5	1.2403
1	3.8255
1.5	5.5706
2	5.6092

Tabla 1: Solución del PVI

así como la gráfica de la solución discreta en la cual podemos observar la evolución del ángulo del péndulo:



6.3 Diseño de métodos numéricos y convergencia

La resolución de los PVI propuestos en los ejemplos anteriores se ha realizado utilizando funciones ya implementadas en Matlab. Concretamente, con el comando `ode23`. A continuación proporcionaremos diversas técnicas que nos van a permitir diseñar nuevos métodos de resolución numérica. También decribiremos los distintos errores que se cometen en esta resolución numérica.

Para diseñar métodos numéricos veremos fundamentalmente tres técnicas. En primer lugar, utilizando cuadraturas, es decir, fórmulas de aproximación de integrales. Utilizaremos técnicas de integración numérica al aplicar el Teorema Fundamental del Cálculo, el cual permite transformar el problema de valor inicial

$$y'(t) = f(t, y(t)), \qquad t \in [a, b], \qquad y(a) = y_a$$

en la siguiente ecuación integral donde intervienen la condición inicial del problema y una integral definida:

$$y(t) = y_a + \int_a^t f(\tau, y(\tau))d\tau.$$

Otra técnica con la que podemos deducir los métodos es la aproximación de las derivadas, es decir, mediante diferenciación numérica. Por ejemplo, podemos aproximar la

derivada de la variable utilizando la definición de derivada como límite de un cociente incremental

$$y'(t) = \lim_{h\to 0} \frac{y(t+h)-y(t)}{h} \quad \Rightarrow \quad y'(t) \approx \frac{y(t+h)-y(t)}{h}.$$

Por último, también podemos obtener los diferentes métodos numéricos a partir de los desarrollos de Taylor de la función hasta un determinado orden en un entorno del punto:

$$y(t+h) = \sum_{j=0}^{m} \frac{f^{(j)}(t)}{j!}h^j + \frac{f^{(m+1)}(\xi)}{(m+1)!}h^{m+1}, \qquad \xi \in]t, t+h[.$$

En las técnicas que acabamos de mencionar, dependiendo de cómo construyamos el método, es decir, dependiendo del orden que utilicemos en la aproximación de la derivada, estaremos cometiendo un error de mayor o menor magnitud.

Cuando aproximamos la función por el cociente incremental o utilizando el polinomio de Taylor truncado, estamos cometiendo en cada nodo t_k, $k = 0, 1, \ldots, N$, un **error de truncamiento local** o de discretización local, $L_k(h)$. Este es el error que cometemos en un solo paso cuando reemplazamos un proceso infinito por uno finito. Si se tiene en cuenta el error acumulado en todo el proceso de discretización, tendríamos el **error de truncamiento global**, $L(h)$, definido como

$$L(h) = \frac{1}{h} \max_{1\leq k\leq N} |L_k(h)|.$$

Por tanto, el error de truncamiento global siempre es una unidad inferior al obtenido si consideramos el error en cada nodo.

Por otro lado, tendríamos el **error de redondeo** local (en cada nodo) o global (su acumulación), inherente al software utilizado para la resolución, provocado por la limitada precisión de los ordenadores. En definitiva, en el proceso de resolución numérica se acumula un **error total** local o global, compuesto por la suma del error de truncamiento y el de redondeo local o global, respectivamente. Asimismo, si conocemos la

solución exacta $y(t)$ en cada nodo t_k, el **error exacto local** es la diferencia entre ésta solución y la solución discreta y_k:

$$e_k = y(t_k) - y_k, \qquad k = 0, 1, \ldots, N.$$

Definición 1: Convergencia y consistencia

Sea $y(t)$, $t \in [a, b]$ la solución exacta de un problema de valor inicial e $y_k \approx y(t_k)$ la solución discreta obtenida de forma numérica, donde $t_k = a + kh$, $k = 0, 1, \ldots, N$, son los nodos de la discretización. Decimos que un **método numérico converge** a la solución del PVI si

$$\lim_{h \to 0} |e_k| = 0.$$

Asimismo, decimos que el **método numérico es consistente** con un problema de valor inicial si verifica

$$\lim_{h \to 0} \max_{1 \leq k \leq N} |L_k(h)| = 0.$$

6.4 Métodos numéricos de un paso para resolver PVIs

Hasta el momento hemos resuelto diferentes PVI utilizando el comando de Matlab `ode23`. Sin embargo, utilizando las técnicas descritas en el apartado anterior, entre otras, podemos diseñar métodos mucho más eficientes que, desde un punto de vista numérico y de cálculo computacional, mejoren considerablemente las prestaciones del comando `ode23`.

Método de Euler

Consideremos la función $y(t)$ solución de la ecuación diferencial

$$y'(t) = f(t, y(t)), \qquad t \in [a, b].$$

Dado el paso h, el desarrollo en serie de Taylor de primer orden de la función $y(t)$ está dado por

$$y(t + h) = y(t) + hy'(t) + \mathcal{O}(h^2),$$

donde $\mathcal{O}(h^2)$ son los términos de orden mayor o igual a 2. Sustituyendo $y'(t) = f(t, y(t))$ en el desarrollo obtenemos

$$y(t + h) = y(t) + hf(t, y(t)) + \mathcal{O}(h^2).$$

Así, conocida la condición inicial $y(t_0) = y(a) = y_a$, podemos aproximar la solución en el segundo nodo de la discretización como

$$y(t_1) \approx y_1 = y_0 + hf(t_0, y_0) = y_a + hf(t_0, y_a),$$

y de forma sucesiva en los siguientes puntos $y(t_{k+1}) \approx y_{k+1}$ como

$$y_{k+1} = y_k + hf(t_k, y_k), \qquad k = 0, 1, \ldots, N - 1. \tag{1}$$

El esquema numérico (1) se denomina **método de Euler**.

Nótese que utilizando la definición de la derivada como límite del cociente incremental y sustituyendo en la ecuación diferencial $y' = f(t, y)$:

$$\frac{y(t + h) - y(t)}{h} \approx f(t, y) \quad \Rightarrow \quad y(t + h) \approx y(t) + hf(t, y)$$

se obtendría también el método de Euler.

Asimismo, también podemos integrar directamente en la ecuación diferencial obte-

niendo

$$y(t) = y_a + \int_a^t f(\tau, y(\tau))d\tau.$$

En particular, tras la discretización de la variable independiente en N subintervalos $[t_k, t_{k+1}]$, $k = 0, 1, \ldots, N-1$, y aproximando la integral con la fórmula de los rectángulos:

$$\begin{aligned} y(t_{k+1}) &= y(t_k) + \int_{t_k}^{t_{k+1}} f(\tau, y(\tau))d\tau \\ &\approx y(t_k) + (t_{k+1} - t_k)f(t_k, y(t_k)) = y(t_k) + hf(t_k, y(t_k)) \end{aligned}$$

obtendríamos de nuevo el método de Euler.

En la deducción de la expresión de la fórmula de Euler, se descartaron en el polinomio de Taylor los términos de orden mayor o igual a dos. El error local de truncamiento está dado por

$$e_{k+1} = y(t_{k+1}) - (y(t_k) + hy'(t_k)) = \frac{h^2}{2}y''(\xi_k), \qquad \xi_k \in]t_k, t_{k+1}[.$$

Teniendo en cuenta que, por ser $y''(t)$ continua, se satisface

$$\sum_{k=0}^{N-1} y''(\xi_k) = Ny''(\xi), \qquad \xi \in [a, b],$$

y que $h = \frac{b-a}{N}$, entonces el error global de truncamiento tras N pasos es

$$\sum_{k=0}^{N-1} \frac{h^2}{2}y''(\xi_k) = \frac{h^2}{2}Ny''(\xi) = \frac{1}{2}(b-a)y''(\xi)h = \mathcal{O}(h), \qquad \xi \in [a, b].$$

Por tanto, el método de Euler tiene orden 1. Recordemos que el orden del error global siempre es una unidad inferior al orden del error local de truncamiento del método. Resumimos esta deducción en el Teorema 2.

Teorema 2

Sea f tal que $y'(t) = f(t, y(t))$, $t \in [a, b]$, con condición inicial $y(a) = y_a$. Si $y(t) \in \mathcal{C}^2[a, b]$ y $\{(t_k, y_k)\}_{k\geq 0}$ es la sucesión de aproximaciones dadas por el método de

Euler, entonces

$$|e_k| \leq |y(t_k) - y_k| = \mathcal{O}(h^2)$$

y

$$E(h) = \frac{1}{h} \max_{1 \leq k \leq N} |e_k| = \mathcal{O}(h).$$

En el Teorema 2 se muestra el orden del error del método de Euler, deducido de forma teórica a partir de desarrollos de Taylor de la función. Sin embargo, conocida la solución analítica, también podemos estimar numéricamente el orden de un método para resolver un PVI como

$$\log_2 \left(\lim_{N \to \infty} \frac{E_{N/2}}{E_N} \right),$$

donde E_N es el error máximo cometido entre la solución exacta $y(t_k)$ y la solución numérica y_k, utilizando N subintervalos, es decir,

$$E_N = \max_{1 \leq k \leq N} |y(t_k) - y_k|.$$

Si, por el contrario, no conocemos el valor de la solución analítica, también podemos realizar una estimación numérica del orden comparando las soluciones discretas obtenidas duplicando el número de subintervalos a partir de

$$\log_2 \left(\lim_{N \to \infty} \frac{\epsilon_{N/2}}{\epsilon_N} \right),$$

donde

$$\epsilon_N = \left\| y_k^{(N)} - y_{1+2k}^{(2N)} \right\|, \qquad k = 0, 1, \ldots, N,$$

siendo $y^{(N)}$ la solución discretizada del PVI utilizando N subintervalos.

La implementación en Matlab del método de Euler para resolver los PVIs basados en una ecuación diferencial de primer orden se muestra a continuación.

Euler.m

```
function [t,y] = Euler(f,a,b,N,ya)
```

```
% Código para resolver un PVI con el metodo de Euler
h=(b-a)/N;
t=a:h:b;
t=t(:);
y=zeros(N+1,1);
y(1)=ya;
for k=1:N
    y(k+1) = y(k)+h*feval(f,t(k),y(k)) ;
end
end
```

Ejemplo 3.

El PVI definido con el modelo poblacional de Verhulst

$$y'(t) = (k - py(t))y(t), \qquad t \in [a, b], \qquad y(a) = y_a$$

sabemos que tiene solución exacta

$$y(t) = \frac{ky_a}{py_a + (k - py_a)e^{-kt}}.$$

Consideremos en PVI con las mismas constantes que en el Ejemplo 1:

$$y'(t) = (3 - 0.1y(t))y(t), \quad t \in [0, 2], \qquad y(0) = 10.$$

Proporcionemos una solución numérica del PVI utilizando el método de Euler para $N = \{2, 4, 8, 16, 32, 64\}$ subintervalos, así como una estimación del orden numérico.

Comenzamos ejecutando la función `Euler.m` para obtener la solución discreta en cada subintervalo, y a continuación calculamos las solución analítica para cada subintervalo discretizado. Mostramos las instrucciones para $N = 2$ y $N = 4$:

```
>> [t2,y2] = Euler('VerhulstPVI',0,2,2,10);
```

```
>> sol2=3*10./(0.1*10+(3-0.1*10).*exp(-3*t2));

>> [t4,y4] = Euler('VerhulstPVI',0,2,4,10);
>> sol4=3*10./(0.1*10+(3-0.1*10).*exp(-3*t4));
```

Calculamos el error máximo obtenido en cada uno de los subintervalos:

```
>> E2=max(abs(sol2-y2));
>> E4=max(abs(sol4-y4));
```

Y la sucesión de los cocientes entre los errores máximos:

```
>> E=[E2 E4 E8 E16 E32 E64];
>> orden=log2(E(1:end-1)./E(2:end));
```

Podemos observar en la Tabla 2 que el orden numérico del método tiende a 1 y que el error máximo disminuye cuando aumenta el número de subintervalos.

N	E_N	$\log_2\left(E_{N/2}/E_N\right)$
2	2.7167	–
4	2.7167	0
8	1.0659	1.3497
16	0.4878	1.1277
32	0.2361	1.0467
64	0.1164	1.0202

Tabla 2: Estimación del orden del método de Euler

Previamente al diseño de métodos numéricos de mayor orden, consideraremos el método que se obtiene si, siguiendo los mismos pasos que en el diseño del método de

Euler, aproximamos la derivada de la ecuación $y'(t) = f(t, y(t))$ como

$$\frac{dy}{dt}(t_{k+1}) \approx \frac{y(t_{k+1}) - y(t_k)}{h}.$$

De aquí, se obtiene el método de Euler implícito:

$$y_{k+1} = y_k + hf(t_{k+1}, y_{k+1}). \tag{2}$$

El método de Euler (1) es un método explícito porque podemos obtener directamente y_{k+1} a partir y_k. En cambio, denominamos a (2) método de Euler implícito porque para obtener y_{k+1} a partir de y_k es necesario resolver la siguiente ecuación no lineal:

$$g(y_{k+1}) = y_{k+1} - y_k - hf(t_{k+1}, y_{k+1}) = 0.$$

A continuación, describimos los pasos que sigue la implementeación del algoritmo del método de Euler implícito:

- Entrada: f, a, b, N, y_a
- Proceso:
  - Obtención de la variable independiente discretizada t
  - Inicialización del vector solución y en a
  - Para k desde 0 hasta $N - 1$:
    - Uso de un método de punto fijo (por ejemplo, método de Newton) para encontrar cada nuevo y_{k+1} resolviendo la ecuación no lineal:

$$g(y_{k+1}) = y_{k+1} - y_k - hf(t_{k+1}, y_{k+1}) = 0.$$

  - Fin para k
- Salida: t, y

En el siguiente vídeo mostramos cómo programar el método de Euler implícito en

Matlab utilizando como método de punto fijo para resolver la ecuación no lineal que interviene en el algoritmo el método de Newton.

Accede al vídeo: Método de Euler implícito con Matlab

Aunque profundizaremos en el tema siguiente en la descripción de métodos implícitos, anticipamos en el Ejemplo 4 un PVI para el cual el método de Euler explícito es inestable, mientras que el método de Euler implícito funciona correctamente.

Ejemplo 4.

Consideremos el PVI definido por el modelo de Malthus

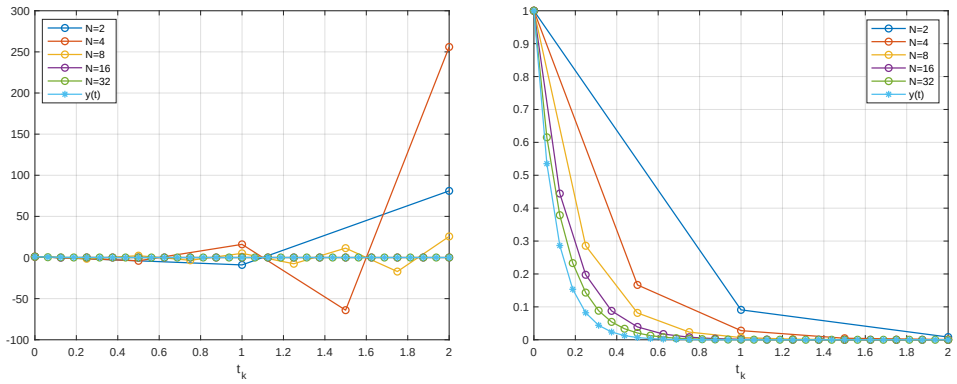
$$y'(t) = ky(t), \qquad t \in [a, b], \qquad y(a) = y_a,$$

cuya solución analítica es

$$y(t) = y_a e^{kt}.$$

Obtendremos la solución discreta del PVI utilizando el método de Euler explícito y el método de Euler implícito en el intervalo $t \in [0, 3]$ con $k = -10$, estimación inicial $y_a = 1$ y tomando $N = \{2, 4, 8, 16, 32\}$ subintervalos.

En la Figura 2 podemos observar la solución obtenida para cada método y cada número de subintervalos, así como la solución exacta del PVI. Observamos que en el método de Euler implícito, aunque se utilicen pocos puntos, se obtiene una curva que se aproxima a la solución.



(a) Euler explícito (b) Euler implícito

Figura 2: Solución de los métodos para $N = \{2, 4, 8, 16, 32\}$

Asimismo, en la Tabla 3 mostramos el error máximo cometido por ambos métodos, comparando con la solución analítica.

N	E_N explícito	E_N implícito
2	81	0.0909
4	256	0.1599
8	25.6289	0.2036
16	0.5365	0.1579
32	0.1603	0.0922

Tabla 3: Máximo error de los métodos de Euler explícito e implícito

Podemos comprobar que para este problema el método de Euler explícito es inestable y requiere de muchos puntos para disminuir el error. Por el contrario, el método de Euler implícito requiere de pocos puntos para aproximarse a la solución de este PVI.

Método de Heun

En la deducción del método de Euler, hemos partido del desarrollo de Taylor de $y(t)$ de primer orden. Consideremos ahora el desarrollo de Taylor de la función hasta orden dos

$$y(t+h) = y(t) + hy'(t) + \frac{h^2}{2}y''(t) + \mathcal{O}(h^3),$$

donde $\mathcal{O}(h^3)$ denota los términos de orden mayor o igual a 3. Teniendo en cuenta que $y'(t) = f(t, y(t))$, y que $y''(t) = \frac{\partial f(t,y(t))}{\partial t} + \frac{\partial f(t,y(t))}{\partial y}y'(t)$, el desarrollo de Taylor queda como

$$\begin{aligned} y(t+h) &= y(t) + hf(t,y) + \frac{h^2}{2}\left(\frac{\partial f(t,y)}{\partial t} + \frac{\partial f(t,y)}{\partial y}f(t,y)\right) + \mathcal{O}(h^3) \\ &= y(t) + \frac{h}{2}f(t,y) + \frac{h}{2}\left(f(t,y) + h\frac{\partial f(t,y)}{\partial t} + hf(t,y)\frac{\partial f(t,y)}{\partial y}\right) + \mathcal{O}(h^3). \end{aligned} \tag{3}$$

Por otro lado, consideremos el desarrollo de Taylor de dos variables

$$f(t+h, y+k) = f(t,y) + h\frac{\partial f(t,y)}{\partial t} + k\frac{\partial f(t,y)}{\partial y} + \cdots$$

Reemplazando k por $hf(t,y)$:

$$f(t+h, y+hf(t,y)) = f(t,y) + h\frac{\partial f(t,y)}{\partial t} + hf(t,y)\frac{\partial f(t,y)}{\partial y} + \mathcal{O}(h^2). \tag{4}$$

Sustituyendo (4) en (3), se obtiene

$$y(t+h) = y(t) + \frac{h}{2}f(t,y) + \frac{h}{2}f\left(t+h, y+hf(t,y)\right) + \mathcal{O}(h^3).$$

En particular, si $t = t_k$ y teniendo en cuenta que $y(t_{k+1}) = y(t_k + h)$, con la aproximación $y_k \approx y(t_k)$ podemos escribir

$$y_{k+1} = y_k + \frac{1}{2}hf(t_k, y_k) + \frac{1}{2}hf\left(t_{k+1}, y_k + hf(t_k, y_k)\right) = y_k + \frac{1}{2}k_1 + \frac{1}{2}k_2, \tag{5}$$

donde $k_1 = hf(t_k, y_k)$ y $k_2 = hf\left(t_{k+1}, y_k + k_1\right)$. La expresión (5) se denomina **método de Heun**.

De forma análoga al método de Euler, podemos deducir la expresión del método utilizando integración numérica. En el método de Heun, siendo h el paso de la discretización, aproximaremos la integral utilizando la regla de los trapecios:

$$y(t_{k+1}) = y_k + \int_{t_k}^{t_{k+1}} f(\tau, y(\tau))d\tau \approx y_k + \frac{h}{2}\left(f(t_k, y(t_k)) + f(t_{k+1}, y(t_{k+1}))\right).$$

Como no conocemos el valor de $y(t_{k+1})$, podemos predecir primero un valor con Euler

$$\bar{y}_{k+1} = y_k + hf(t_k, y_k)$$

y ajustarlo después por trapecios

$$y(t_{k+1}) \approx y_{k+1} = y_k + \frac{h}{2}\left(f(t_k, y_k) + f(t_{k+1}, \bar{y}_{k+1})\right)$$

de modo que obtenemos la misma expresión que en (5).

El Teorema 3 muestra que el método de Heun tiene orden 2 con un error global de $\mathcal{O}(h^2)$.

Teorema 3

Sea f tal que $y'(t) = f(t, y(t))$, $t \in [a, b]$, con condición inicial $y(a) = y_a$. Si $y(t) \in \mathcal{C}^3[a, b]$ y $\{(t_k, y_k)\}_{k\geq 0}$ es la sucesión de aproximaciones dadas por el método de Heun, entonces

$$\begin{aligned} |e_k| &\leq |y(t_k) - y_k| \\ &= \left|y(t_k) - y_{k-1} - \frac{1}{2}hf(t_{k-1}, y_{k-1}) - \frac{1}{2}hf(t_k, y_{k-1} + hf(t_{k-1}, y_{k-1}))\right| \\ &= \mathcal{O}(h^3) \end{aligned}$$

y

$$E(h) = \frac{1}{h} \max_{1\leq k\leq N} |e_k| = \mathcal{O}(h^2).$$

El siguiente algoritmo muestra los pasos para implementar con Matlab el algoritmo del método de Heun:

- Entrada: f, a, b, N, y_a
- Proceso:
  - Obtención de la variable independiente discretizada t
  - Inicialización del vector solución y en a
  - Para k desde 0 hasta $N-1$:
    - Cálculo de k_1 y k_2
    - $y_{k+1} = y_k + \frac{k_1}{2} + \frac{k_2}{2}$
  - Fin para k
- Salida: t, y

Ejemplo 5.

Obtén la solución numérica del PVI

$$y'(t) = (3 - 0.1y(t))y(t), \qquad t \in [0, 2], \qquad y(0) = 10,$$

utilizando el método de Heun y $N = \{2, 4, 8, 16, 32, 64\}$ subintervalos. Obtén una estimación del orden numérico sin utilizar la solución analítica.

Apliquemos el método de Heun variando el valor de N. Por ejemplo, para $N = 2$ y $N = 4$ utilizamos la instrucción:

```
>> [t2,y2] = Heun('VerhulstPVI',0,2,2,10);
>> [t4,y4] = Heun('VerhulstPVI',0,2,4,10);
```

Para aproximar numéricamente el orden del método sin conocer la solución analítica, calculamos los valores de ϵ_N como $\epsilon_N = \left\| y_k^{(N)} - y_{1-2k}^{(2N)} \right\|$ o bien:

```
>> eps4=max(abs(y2-y4(1:2:end)));
>> eps8=max(abs(y4-y8(1:2:end)));
```

Obtenemos la sucesión de los cocientes entre los valores de ϵ_N para realizar la estimación del orden:

```
>> epsilon=[eps4 eps8 eps16 eps32 eps64];
>> orden=log2(epsilon(1:end-1)./epsilon(2:end));
```

En los resultados de la Tabla 4 podemos observar que la columna del orden va tendiendo a 2 conforme vamos duplicando los subintervalos.

N	ϵ_N	$\log_2\left(\epsilon_{N/2}/\epsilon_N\right)$
2	–	–
4	18.2934	–
8	1.9319	3.2432
16	0.3287	2.5549
32	0.0686	2.2603
64	0.0158	2.1153

Tabla 4: Estimación del orden del método de Heun

Método de Runge-Kutta

Deduciremos con detalle el método de Runge-Kutta utilizando integración numérica. Para ello, consideremos la expresión general de un PVI descrito por una ecuación diferencial de primer orden

$$y'(t) = f(t, y(t)), \qquad t \in [a, b], \qquad y(a) = t_a.$$

Integrando directamente la ecuación diferencial obtenemos

$$y(t) = y_a + \int_a^t f(\tau, y(\tau))d\tau.$$

Realizando la discretización del intervalo $[a, b]$ en los puntos $t_k = a+kh$, $k = 0, 1, \ldots, N$, e integrando en cada uno de los subintervalos $[t_k, t_{k+1}]$:

$$y(t_{k+1}) = y(t_k) + \int_{t_k}^{t_{k+1}} f(\tau, y(\tau))d\tau, \qquad k = 0, 1, \ldots, N-1. \tag{6}$$

Aproximando la integral definida utilizando la fórmula de Simpson

$$\int_{t_k}^{t_{k+1}} f(\tau, y(\tau))d\tau \approx \frac{t_{k+1} - t_k}{6}\left(f(t_k, y_k) + 4f(t_{k+\frac{1}{2}}, y_{k+\frac{1}{2}}) + f(t_{k+1}, y_{k+1})\right)$$

y reemplazando en la ecuación (6), se obtiene:

$$y_{k+1} = y_k + \frac{h}{6}\left(f(t_k, y_k) + 4f(t_{k+\frac{1}{2}}, y_{k+\frac{1}{2}}) + f(t_{k+1}, y_{k+1})\right),$$

siendo h el paso de la discretización. Sin embargo, tanto $f(t_{k+\frac{1}{2}}, y_{k+\frac{1}{2}})$ como $f(t_{k+1}, y_{k+1})$ son valores que no conocemos, motivo por el cual realizamos las aproximaciones

$$f(t_{k+\frac{1}{2}}, y_{k+\frac{1}{2}}) \approx f\left(t_k + \frac{h}{2}, y\left(t_k + \frac{h}{2}\right)\right) \approx \frac{1}{2}(k_2 + k_3),$$

$$f(t_{k+1}, y_{k+1}) \approx k_4,$$

donde

$$\begin{aligned}
k_1 &= f(t_k, y_k),\\
k_2 &= f\left(t_k + \frac{h}{2}, y_k + \frac{h}{2}k_1\right),\\
k_3 &= f\left(t_k + \frac{h}{2}, y_k + \frac{h}{2}k_2\right),\\
k_4 &= f(t_{k+1}, y_k + hk_3).
\end{aligned}$$

A partir de las aproximaciones anteriores, la expresión general del **método de Runge-Kutta** es

$$y_{k+1} = y_k + \frac{h}{6}(k_1 + 2k_2 + 2k_3 + k_4), \qquad k = 0, 1, \ldots, N-1.$$

Teorema 4

Sea f tal que $y'(t) = f(t, y(t)), t \in [a, b]$, con condición inicial $y(a) = y_a$. Si $y(t) \in \mathcal{C}^5[a, b]$ y $\{(t_k, y_k)\}_{k\geq 0}$ es la sucesión de aproximaciones dadas por el método de Runge-Kutta, entonces

$$|e_k| \leq |y(t_k) - y_k| = \mathcal{O}(h^5)$$

y

$$E(h) = \frac{1}{h} \max_{1\leq k\leq N} |e_k| = \mathcal{O}(h^4).$$

Por tanto, el error global del método de Runge-Kutta es $\mathcal{O}(h^4)$.

Los pasos para implementar con Matlab el algoritmo del método de Runge-Kutta son los siguientes:

- Entrada: f, a, b, N, y_a
- Proceso:
  - Obtención de la variable independiente discretizada t
  - Inicialización del vector solución y en a
  - Para k desde 0 hasta $N - 1$:
    - Cálculo de k_1, k_2, k_3 y k_4
    - $y_{k+1} = y_k + \frac{h}{6}(k_1 + 2k_2 + 2k_3 + k_4)$
  - Fin para k
- Salida: t, y

Ejemplo 6.

Obtén la solución numérica del PVI

$$y'(t) = (3 - 0.1y(t))y(t), \qquad t \in [0, 2], \qquad y(0) = 10,$$

utilizando el método de Runge-Kutta y $N = \{2, 4, 8, 16, 32, 64\}$ subintervalos. Obtén una estimación del orden numérico sabiendo que la solución analítica es:

$$y(t) = \frac{30}{1 + 2e^{-3t}}.$$

Seguiremos los mismos pasos que en los Ejemplos 3 y 5. Si llamamos `RK.m` a la función de Matlab que implementa el método de Runge-Kutta, tendríamos que realizar el mismo proceso variando el número de subintervalos. Por ejemplo, para $N = 2$ ejecutaríamos:

```
>> a=0; b=2; N=2; ya=10;
>> [t2,y2] = RK('VerhulstPVI',a,b,N,ya);
>> sol2=30./(1+2*exp(-3*t2));
>> E2=max(abs(sol2-y2));
```

Podemos observar en la última columna de la Tabla 5 la estimación del orden obtenida duplicando el número de subintervalos.

N	E_N	$\log_2 \left(E_{N/2}/E_N\right)$
2	4.7316	–
4	0.1442	5.0361
8	0.0065	4.4646
16	0.0003	4.2346
32	0	4.1223
64	0	4.0624

Tabla 5: Estimación del orden del método de Runge-Kutta

Métodos numéricos para resolver sistemas de ecuaciones diferenciales

La extensión de los métodos numéricos de Euler, Heun y Runge-Kutta a la resolución de PVI basados en sistemas de ecuaciones diferenciales es directa. Por ejemplo, utilizando

la notación compacta anteriormente presentada $Y' = F(t, Y(t))$ podemos escribir el método de Heun para el caso vectorial como

$$Y_{k+1} = Y_k + hF(t_k, Y_k), \qquad k = 0, 1, \ldots, N-1.$$

En cuanto a su implementación, debemos tener en cuenta que los vectores son ahora matrices, y que la evaluación de las funciones debe realizarse en las dimensiones adecuadas. Partiendo de la función `Euler.m`, presentamos a continuación la implementación de este método en Matlab para resolver PVI basados en sistemas de ecuaciones diferenciales.

Euler_Sistemas.m

```
function [t,Y] = Euler_Sistemas(f,a,b,N,Ya)
% Código para resolver un PVI basado en un sistema
% de EDOs con el metodo de Euler
h=(b-a)/N;
t=a:h:b;
t=t(:);
Y=zeros(N+1,length(Ya));
Y(1,:)=Ya;
for k=1:N
    Y(k+1,:) = Y(k,:)+h*feval(f,t(k),Y(k,:))';
end
end
```

La extensión de los métodos de Heun y Runge-Kutta al caso vectorial también es directa. Asimismo, adaptaríamos las funciones de Matlab de forma análoga al método de Euler.

Ejemplo 7.

Uno de los modelos más sencillos para estudiar la propagación de enfermedades infecciosas es el modelo SIR, dado por el sistema de ecuaciones diferenciales:

$$\begin{cases} S'(t) &= -\beta S(t)I(t), \\ I'(t) &= \beta S(t)I(t) - \nu I(t), \qquad \beta, \nu > 0, \\ R'(t) &= \nu I(t), \end{cases}$$

donde $S(t)$, $I(t)$ y $R(t)$ son la población susceptible, infectada y recuperada, respectivamente. Vamos a tomar $\beta = 0.01$, $\nu = 0.5$ y las condiciones iniciales:

$$S(0) = 100, \qquad I(0) = 32, \qquad R(0) = 5.$$

Estudiaremos la evolución diaria de la enfermedad durante 10 días utilizando los métodos de Euler, Heun y Runge-Kutta para sistemas. Es decir, tomaremos $N = 10$ subintervalos en $[0, 10]$.

En primer lugar, implementamos la función de Matlab que define el modelo SIR:

```
function dY = ModeloSIR(t,Y)
    S=Y(1); I=Y(2); R=Y(3);
    beta=0.01; nu=0.5;
    dY=[-beta*S.*I;...
        beta*S.*I-nu*I;...
        nu*I];
end
```

Tras adaptar el código de los métodos numéricos para resolver sistemas, definimos los parámetros del problema y ejecutamos los métodos de Euler, Heun y Runge-Kutta:

```
>> a=0; b=10; N=10;
>> Ya=[100; 32; 5];
>> [t,YE] = Euler_Sistemas('ModeloSIR',a,b,N,Ya);
>> [t,YH] = Heun_Sistemas('ModeloSIR',a,b,N,Ya);
>> [t,YR] = RK_Sistemas('ModeloSIR',a,b,N,Ya);
```

En las Tablas 6-8 mostramos algunos de los resultados obtenidos con los métodos de Euler, Heun y Runge-Kutta, respectivamente. Asimismo, en las Figuras 3-5 se observa la evolución del número de individuos susceptibles, infectados y recuperados de la enfermedad para cada método.

t	$S(t)$	$I(t)$	$R(t)$
0	100	32	5
2	35.36	56.64	45
4	7.91	31.58	97.49
6	4.42	10.13	122.43
8	3.75	2.97	130.26
10	3.58	0.85	132.55

Tabla 6: Método de Euler

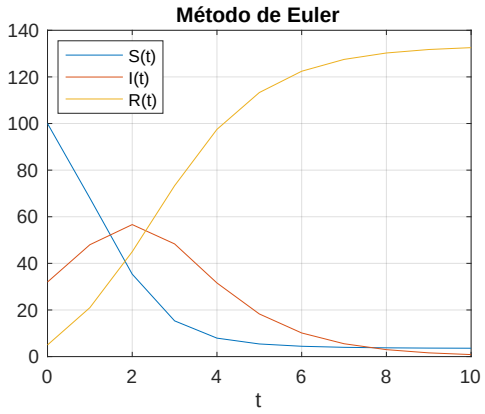


Figura 3

t	$S(t)$	$I(t)$	$R(t)$
0	100	32	5
2	42.85	45.02	49.11
4	20.31	28.96	87.71
6	13.36	14.86	108.77
8	10.88	7.09	119.02
10	9.88	3.29	123.81

Tabla 7: Método de Heun

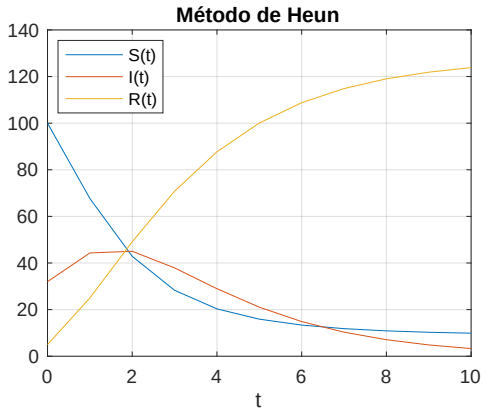


Figura 4

t	$S(t)$	$I(t)$	$R(t)$
0	100	32	5
2	42.4	46.7	47.89
4	19.27	30.38	87.33
6	12.37	15.14	109.47
8	10.02	6.94	120.03
10	9.11	3.09	124.79

Tabla 8: Método de Runge-Kutta

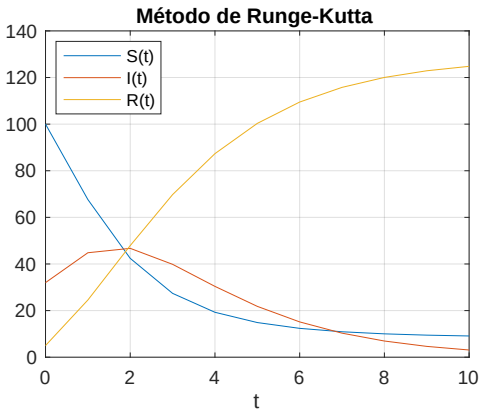


Figura 5

En esta sección hemos descrito algunos métodos numéricos de órdenes distintos. No obstante, podemos diseñar más métodos numéricos para aproximar la solución de un PVI utilizando otras técnicas y con distintas fórmulas de integración numérica.

Accede al vídeo: Diseño e implementación de un método numérico con la fórmula de cuadratura del punto medio