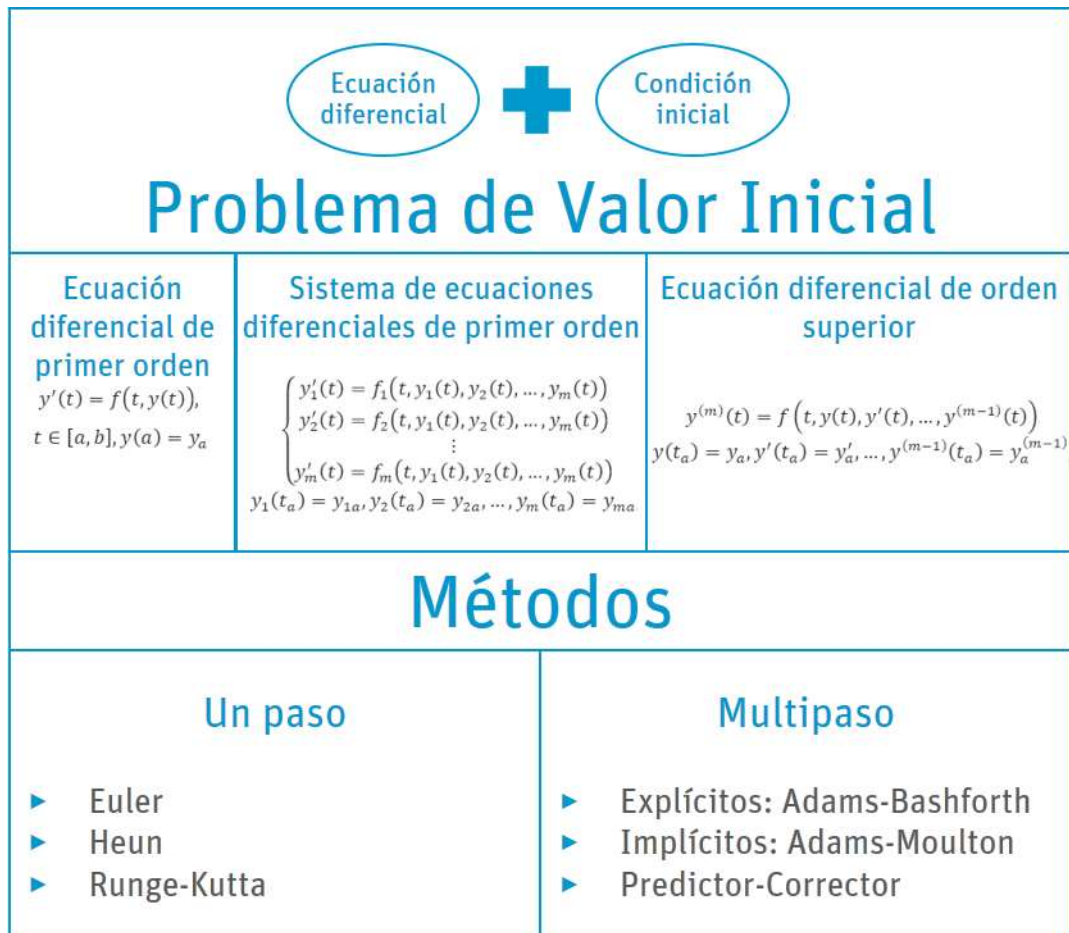


Métodos Numéricos Avanzados en Ingeniería

Problemas de valor inicial. Métodos de un paso

Índice

Esquema	3
Ideas clave	4
3.1. ¿Cómo estudiar este tema?	4
3.2. Problemas de valor inicial	6
3.3. Métodos numéricos para resolver PVI. Métodos de un paso	14
Lo + recomendado	33
+ Información	35
Test	40



3.1. ¿Cómo estudiar este tema?

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

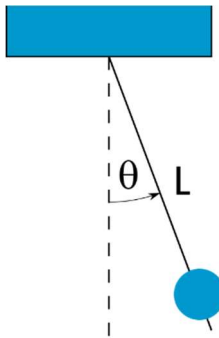
Las ecuaciones diferenciales nos permiten modelar multitud de fenómenos físicos, mecánicos, biológicos o económicos, entre otros, con la variación de determinadas magnitudes en función de otras. Esta modelización no tiene otro objetivo que predecir qué va a ocurrir en un determinado momento o bajo una serie de parámetros.

De forma general, vamos a trabajar con la variación de determinadas magnitudes en función del tiempo. Si tomamos el ejemplo de la cinemática de la partícula, sabemos que la velocidad es la variación de la trayectoria con respecto del tiempo, o que la aceleración es la variación de la velocidad con respecto del tiempo. Si tomamos el modelo de población de Malthus, podemos predecir cómo va a modificarse la población a lo largo del tiempo.

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 un ejemplo de problema de valor inicial. Consideremos un péndulo de longitud L cuya ecuación del movimiento viene descrita por:

$$\frac{d^2\theta}{dt^2} + \frac{g}{L}\sin\theta = 0$$



donde g es la aceleración de la gravedad. Soltamos el péndulo en el instante $t = 0$, de forma que en ese instante el ángulo inicial es $\theta(0) = \theta_a$, y la velocidad angular es $\theta'(0) = 0$. Con los métodos que vamos a ver a lo largo de estos dos temas, seremos capaces de obtener una solución numérica a las posiciones que ocupa el péndulo a lo largo del tiempo.

Figura 1. Problema de valor inicial: péndulo.

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.

Los apartados de los que consta este tema son:

- ▶ Problemas de valor inicial (PVI):
 - PVI definidos por ecuaciones diferenciales de primer orden.
 - PVI definidos por sistemas de ecuaciones diferenciales de primer orden.
 - PVI definidos por ecuaciones diferenciales de orden n .
- ▶ Métodos numéricos de un paso:
 - Euler.
 - Heun.
 - Runge-Kutta.

3.2. Problemas de valor inicial

Los problemas de valor inicial se pueden presentar a partir de diferentes esquemas de ecuaciones diferenciales. Comenzaremos por presentar los más sencillos, aquellos basados en una ecuación diferencial de primer orden, e iremos aumentando la complejidad.

PVI definidos por ecuaciones diferenciales de primer orden

Un PVI tiene como objetivo encontrar la solución $y(t)$ de la ecuación diferencial de primer orden:

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

conociendo la condición inicial:

$$y(a) = y_a$$

Teorema 1. Supongamos que $D = \{(t, y) : t \in [a, b], y \in \mathbb{R}\}$, y que $f(x, y)$ es continua en D . Si dados $(t_1, y_1), (t_2, y_2) \in D$, 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:

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

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

A partir del Teorema 1, las técnicas analíticas tratan de obtener la solución continua $y(t)$. En el caso que nos ocupa, que son las técnicas numéricas, vamos a obtener la solución discreta $y_i = y(t_i)$, $i = 0, 1, 2, \dots, n$, donde t_i , $i = 0, 1, 2, \dots, n$ son los nodos de la discretización.

Para ello, el primer paso que debemos dar consiste en discretizar la variable independiente t , dividiendo el intervalo completo en N subintervalos, como se ilustra en la Figura .



Figura 2. Proceso de discretización.

Dividimos el intervalo $[a, b]$ en N subintervalos $[t_i, t_{i+1}]$, cuyos extremos son los nodos equiespaciados $t_i = a + ih$, $i = 0, 1, 2, \dots, N$, siendo $h = \frac{b-a}{N}$ el paso de la partición. La solución numérica la obtendremos en los nodos t_i , de forma que $y_i \approx y(t_i)$.

Una vez tenemos discretizada la variable independiente, veamos cómo resolver de forma genérica en Matlab este tipo de PVI. Para ello, utilizaremos el comando `ode23` de Matlab, que es un método numérico de bajo orden. En primer lugar, debemos definir a través de una *function* la función $y'(t) = f(t, y(t))$. A continuación, definiremos los valores de t discretizados. Por último, indicaremos la condición inicial $y(a) = y_a$. El Ejemplo 1 muestra un caso.

Ejemplo 1. Resuelve el PVI:

$$y'(t) = (1 - 2t)y(t), t \in [0, 3], y(0) = 1$$

utilizando 30 subintervalos. Representa la solución $y(t)$.

En primer lugar, definimos una *function* que implemente la función $f(t, y(t)) = (1 - 2t)y(t)$.

```
function dy=ejemplo1(t,y)
dy=(1-2*t)*y;
end
```

A continuación, definimos la variable independiente discretizada, tomando 30 subintervalos:

```
>> a=0; b=3; N=30;
>> h=(b-a)/N;
>> td=a:h:b;
```

Definimos la condición inicial:

```
>> ya=1;
```

Ejecutamos la función ode23:

```
>> [t,y]=ode23('ejemplo1',td,ya);
```

Representamos la solución:

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

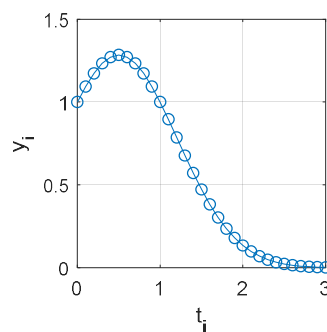


Figura 3. Representación solución PVI.

PVI definidos por sistemas de ecuaciones diferenciales de primer orden

Un sistema de ecuaciones diferenciales de primer orden expresa las relaciones que verifican varias funciones desconocidas y sus primeras derivadas respecto a la variable independiente. De este modo, el objetivo es resolver:

$$\begin{cases} y_1'(t) = f_1(t, y_1(t), y_2(t), \dots, y_m(t)) \\ y_2'(t) = f_2(t, y_1(t), y_2(t), \dots, y_m(t)) \\ \vdots \\ y_m'(t) = f_m(t, y_1(t), y_2(t), \dots, y_m(t)) \end{cases}$$

Si conocemos los valores de las m funciones incógnita en el instante inicial t_a , es decir:

$$y_1(t_a) = y_{1a}, y_2(t_a) = y_{2a}, \dots, y_m(t_a) = y_{ma}$$

entonces estamos ante un PVI.

La estructura de este sistema permite utilizar una notación compacta en términos vectoriales. De este modo, si escribimos las funciones incógnita como $Y: \mathbb{R} \rightarrow \mathbb{R}^m$, tal que:

$$Y(t) = [y_1(t) \quad y_2(t) \quad \cdots \quad y_m(t)]^T$$

y las ecuaciones como componentes de una función vectorial de varias variables $F: \mathbb{R}^{m+1} \rightarrow \mathbb{R}^m$, tal que:

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

podemos escribir el sistema como:

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

siendo la condición inicial del PVI:

$$Y_a = Y(t_a) = [y_1(t_a) \quad y_2(t_a) \quad \cdots \quad y_m(t_a)]^T = [y_{1a} \quad y_{2a} \quad \cdots \quad y_{ma}]^T$$

La discretización de la variable independiente sigue la misma estructura que en el caso de las ecuaciones diferenciales de primer orden. Para resolver los PVI planteados como ecuaciones diferenciales de primer orden, la modificación que tenemos que realizar se basa en la estructura de la función que introducimos como *function* y en la adaptación del método numérico al caso vectorial. En cuanto a la *function*, tomará como parámetros de entrada las funciones $f_i(t, y_1(t), \dots, y_m(t))$, $i = 1, 2, \dots, m$.

Como parámetro de salida tendremos un vector con los valores de las f_i evaluadas en los puntos a los que llame el método numérico correspondiente.

Para cada uno de los métodos numéricos que desarrollemos, veremos su implementación vectorial. Sobre el comando `ode23` no hay que introducir modificaciones. No obstante, hay que destacar que como parámetros de entrada habrá que introducir, además de la función que implementa f_i y la variable independiente discretizada, a las condiciones iniciales y_{ia} , $i = 1, 2, \dots, m$. El resultado del comando `ode23` nos aportará las soluciones y_i , $i = 1, 2, \dots, m$, una en cada columna. Veamos un caso de resolución en el Ejemplo 2.

Ejemplo 2. Obtén la solución numérica del PVI:

$$\begin{cases} y_1'(t) = y_2(t), \\ y_2'(t) = -y_1(t) - 2e^t + 1 \\ y_3'(t) = -y_1(t) - e^t + 1 \end{cases}$$

con $t \in [0, 2]$, $y_1(0) = 1$, $y_2(0) = 0$, $y_3(0) = 1$, tomando como paso $h = 0.1$. Representa las soluciones $y_1(t)$, $y_2(t)$ e $y_3(t)$

Definimos una *function* que implemente el sistema:

```
function DY=ejemplo2(t,y)
y1=y(1); y2=y(2); y3=y(3);
DY=[y2; -y1-2*exp(t)+1; -y1-exp(t)+1];
End
```

A continuación, definimos la variable independiente discretizada, tomando el paso indicado:

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

Definimos el vector de condiciones iniciales:

```
>> ya=[1;0;1];
```

Ejecutamos la función ode23:

```
>> [t,y]=ode23('ejemplo2',td,ya);
```

Representamos la solución:

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

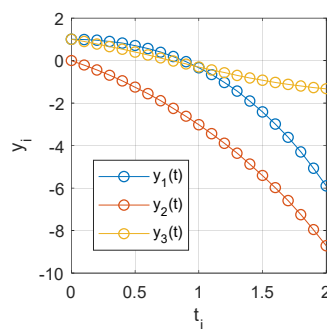


Figura 4. Representación solución PVI 2.

PVI definidos por ecuaciones diferenciales de orden superior

Las ecuaciones diferenciales de orden m son aquellas ecuaciones diferenciales en las que intervienen la función incógnita y sus derivadas hasta el orden m . De forma general, podemos expresar una ecuación diferencial de orden m como:

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

De nuevo, si conocemos los valores de las condiciones iniciales de la función incógnita y las derivadas hasta el orden $m - 1$, es decir:

$$y(t_a) = y_a, y'(t_a) = y'_a, \dots, y^{(m-1)}(t_a) = y_a^{(m-1)}$$

estamos ante un PVI.

El procedimiento para la resolución de estas ecuaciones diferenciales es similar al de los sistemas de ecuaciones diferenciales de primer orden, a través del cambio de variables:

$$y_1(t) = y(t), y_2(t) = y'(t), y_3(t) = y''(t), \dots, y_m(t) = y^{(m-1)}(t)$$

De este modo, la ecuación diferencial de orden m se ha convertido en el sistema de ecuaciones diferenciales:

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

El Ejemplo 3 muestra un caso de este tipo de PVI.

Ejemplo 3. Obtén la solución del PVI dado por el péndulo de longitud $L = 1$ m y la aceleración de la gravedad $g = 9.8$ m/s², modelado por la ecuación diferencial:

$$\frac{d^2\theta(t)}{dt^2} - \frac{g}{L} \sin \theta(t) = 0$$

sabiendo que $\theta(0) = \pi/6$ rad, $\theta'(0) = 0$ m/s y tomando como paso $h = 0.1$. Indica el valor del ángulo θ para $t = \{0,1,2\}$ y representa la evolución de $\theta(t)$.

Reescribamos la ecuación diferencial, dejando a un lado la derivada de mayor orden.

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

Se trata de una ecuación diferencial de orden 2. Hagamos los cambios de variables.

$$\begin{bmatrix} \theta_1(t) \\ \theta_2(t) \end{bmatrix} = \begin{bmatrix} \theta(t) \\ \theta'(t) \end{bmatrix} \rightarrow \begin{bmatrix} \theta'_1(t) \\ \theta'_2(t) \end{bmatrix} = \begin{bmatrix} \theta'(t) \\ \theta''(t) \end{bmatrix} = \begin{bmatrix} \theta_2(t) \\ \frac{g}{L} \sin \theta_1(t) \end{bmatrix}$$

Definimos una *function* que implemente el sistema.

```
function DY=ejemplo3(t,theta)
L=1; g=9.8;
theta1=theta(1); theta2=theta(2);
DY=[theta2; g/L*sin(theta1)];
End
```

A continuación, definimos la variable independiente discretizada, tomando el paso indicado:

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

Definimos el vector de condiciones iniciales:

```
>> theta_a=[pi/6;0];
```

Ejecutamos la función ode23:

```
>> [t,theta]=ode23('ejemplo3',td,theta_a);
Obtenemos los valores del  $\theta(t)$  para  $t = \{0,1,2\}$ .
>> table(t(1:10:21), theta(1:10:21,1))
```

t	$\theta(t)$
0	0.5236
1	3.8255
2	5.6092

Tabla 1. Valores de $\theta(t)$ para $t = \{0,1,2\}$.

Representamos la solución:

```
>> plot(t,theta(:,1),'o-')
```

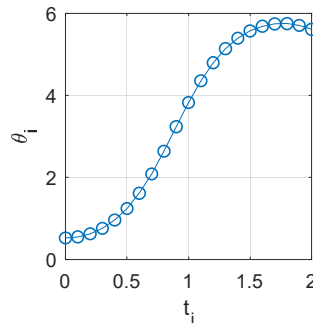


Figura 5. Representación solución PVI 3.

Cabe destacar que, en el caso de las ecuaciones diferenciales de orden m , la solución aportada por los métodos numéricos es una matriz de m columnas. La columna i de la solución numérica representa la función solución $y^{(i-1)}(t)$, para $i = 1, 2, \dots, m - 1$. Es por ello que para representar la función solución $y(t)$ tomamos solo la primera columna, como ocurre en el Ejemplo 3.

3.3. Métodos numéricos para resolver PVI.

Métodos de un paso

Hasta ahora hemos resuelto diferentes tipos de PVI a partir del comando `ode23` de Matlab. Sin embargo, hay métodos mucho más eficientes y sofisticados que, desde un punto de vista numérico y de cálculo computacional, mejoran considerablemente las prestaciones del comando `ode23`.

Método de Euler

Comenzaremos por un método muy elemental e intuitivo. Haciendo el desarrollo en serie de Taylor de la función $y(t)$:

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

donde $\mathcal{O}(h^2)$ son los términos de orden mayor o igual a 2. Podemos despejar la función derivada como:

$$y'(t) = \frac{y(t+h) - y(t)}{h}$$

y, reemplazando en la expresión de la ecuación diferencial de primer orden, obtenemos:

$$\begin{aligned} y'(t) = f(t, y(t)) &\leftrightarrow \frac{y(t+h) - y(t)}{h} = f(t, y(t)) \leftrightarrow \\ &\leftrightarrow y(t+h) = y(t) + hf(t, y(t)) \end{aligned}$$

siendo h muy próximo a cero. Si conocemos la condición inicial $y(t_a) = y_a$, podemos obtener los diferentes puntos como:

$$y_{k+1} = y_k + hf(t_k, y_k), k = 0, 1, \dots, N-1$$

Al realizar el desarrollo de Taylor se descartaron en la expresión del error los términos de orden mayor o igual a 2. El error local de truncamiento es:

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

Al ser $y''(t)$ continua:

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

Después de N pasos, el error global acumulado es:

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

Por tanto, el orden del método de Euler es 1. En el Teorema 2 recogemos este resultado.

Teorema 2. Sea f tal que $y'(t) = f(t, y(t))$ con condición inicial $y(t_a) = y_a$. Si

$$\left| \frac{\partial f}{\partial y}(t, y) \right| \leq M_1, \quad \text{e} \quad |y''(t)| \leq M_2, \quad \forall t \in [a, b]$$

entonces las aproximaciones de Euler y_0, y_1, \dots, y_N convergen a la solución exacta $y(t)$ cuando $h \rightarrow 0$, y el error de discretización del método de Euler

$$|e_k| = |y(t_k) - y_k| \leq \frac{hM_2}{2M_1} (e^{M_1(t_k - t_a)} - 1)$$

satisface $E(h) = \mathcal{O}(h)$.

Una manera de estimar numéricamente el orden de un método conociendo la solución analítica es a partir de:

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

donde E_N es el error máximo entre la solución analítica y la solución numérica utilizando N subintervalos.

En caso de que no conozcamos la solución analítica, también es posible obtener una estimación numérica del orden del método a partir de:

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

donde $\epsilon_N = \|y_k^{(N)} - y_{1+2k}^{(2N)}\|, k = 0, 1, \dots, N$, e $y^{(N)}$ representa la solución discretizada del PVI utilizando N subintervalos.

El código en Matlab que implementa el método de Euler se muestra a continuación:

```
1 function [t,y]=eulerPVI(fun,a,b,N,ya)
2 % Código para resolver un PVI por el método de Euler
3 % Entrada: fun -> función f(t,y)
4 %           a, b -> intervalo de la variable t
5 %           N -> número de subintervalos
6 %           ya -> condición inicial en t=a
7 % Salida:  t -> variable t discretizada
8 %           y -> vector columna con la solución
9 h=(b-a)/N;
10 t=a:h:b;
11 y(1)=ya;
12 for k=1:N
13     y(k+1)=y(k)+h*feval(fun,t(k),y(k));
14 end
15 t=t(:); y=y(:);
```

Veamos un caso de aplicación del método de Euler en el Ejemplo 4.

Ejemplo 4. Obtén la solución del PVI

$$y'(t) = (1 - 2t)y(t), y(0) = 1, t \in [0,3]$$

Utiliza el método de Euler y $N = \{2,4,8,16,32\}$ subintervalos. Obtén el error máximo sabiendo que la solución analítica es

$$y(t) = e^{\frac{1}{4} - \left(\frac{1}{2}t\right)^2} \text{ y una estimación del orden numérico.}$$

La *function* de este PVI la implementamos en el Ejemplo 1. Apliquemos el método de Euler para los diferentes subintervalos (indicamos solo las dos primeras instrucciones):

```
>> [t2,y2]=eulerPVI('ejemplo1',0,3,2,1);
>> [t4,y4]=eulerPVI('ejemplo1',0,3,4,1);
```

Obtenemos la solución analítica para los diferentes intervalos (indicamos solo las dos primeras instrucciones).

```
>> yA2=exp(1/4-(1/2-t2).^2);
>> yA4=exp(1/4-(1/2-t4).^2);
```

Calculamos el error máximo cometido en cada número de subintervalos (indicamos solo las dos primeras instrucciones).

```
>> eM2=max(abs(y2-yA2));
>> eM4=max(abs(y4-yA4));
```

Obtenemos la sucesión de los cocientes entre los errores máximos.

```
>> eM=[eM2 eM4 eM8 eM16 eM32];
>> orden=log2(eM(1:end-1)./eM(2:end));
```

Obtenemos los siguientes resultados:

N	Error máximo	$\log_2 \left(\frac{E_N}{E_{N/2}} \right)$
2	5.0025	
4	0.8862	2.4970
8	0.3531	1.3275
16	0.1688	1.0648
32	0.0819	1.0430

Tabla 2. Resultados PVI 4.

Vemos como la columna del orden va tendiendo a 1 conforme vamos duplicando el número de subintervalos.

Hemos visto hasta el momento la implementación del método de Euler para resolver PVI basados en ecuaciones diferenciales de primer orden. A continuación, veremos

cómo aplicar el método de Euler sobre PVI basados en sistemas de ecuaciones diferenciales.

A nivel conceptual, podemos extender el método del caso escalar al caso vectorial sin mayores problemas. Partiendo de:

$$y_{k+1} = y_k + h f(t_k, y_k), k = 0, 1, \dots, N - 1$$

podemos reescribir para el caso vectorial:

$$Y_{k+1} = Y_k + h F(t_k, Y_k), k = 0, 1, \dots, N - 1$$

A nivel de implementación tenemos que tener en cuenta que los vectores se convierten en matrices, y que la evaluación de las funciones debe estar en las dimensiones adecuadas. Es por ello que, partiendo del primer código, presentamos en este la implementación del método de Euler para sistemas.

```
1 function [t,Y]=eulerPVISistema(fun,a,b,N,Ya)
2 % Código en Matlab para resolver un sistema de PVIs por el
  método de Euler
3 % Entrada: fun -> función F(t,Y)
4 %          a, b -> intervalo de la variable t
5 %          N -> número de subintervalos
6 %          Ya -> condiciones iniciales en t=a
7 % Salida:  t -> variable t discretizada
8 %          Y -> matriz con la solución
9 Ya=Ya(:); Ya=Ya';
10 h=(b-a)/N;
11 t=a:h:b; t=t(:);
12 Y=zeros(N,size(Ya,2));
13 Y(1,:)=Ya;
14 for k=1:N
```

```

15     Y(k+1,:)=Y(k,:)+h*feval(fun,t(k),Y(k,:))';
16 end

```

El Ejemplo 5 muestra un caso de aplicación del método de Euler para sistemas.

Ejemplo 5. Utiliza el método de Euler para obtener la solución numérica de:

$$\begin{cases} u_1' = 2u_1 + 2u_2 \\ u_2' = 3u_1 + u_2 \end{cases}$$

en el intervalo $t \in [0,2]$, sabiendo que $u_1(0) = -1, u_2(0) = 4$ y tomando como paso $h = 0.5$. Indica el valor de las soluciones u_1 y u_2 para $t = \{0,1,2\}$ y representa la evolución de $u_1(t)$ y $u_2(t)$.

Definimos una *function* que implemente el sistema.

```

1 function DY=ejemplo5(t,U)
2 u1=U(1); u2=U(2);
3 DY=[2*u1+2*u2; 3*u1+u2];
4 End

```

A continuación, definimos los parámetros de entrada relacionados con la variable independiente que requiere el método de Euler:

```
>> a=0; b=2; h=0.5; N=(b-a)/h;
```

Definimos el vector de condiciones iniciales:

```
>> Ua=[-1;4];
```

Ejecutamos la función `eulerPVISistema`:

```

>> [t,U]=eulerPVISistema('ejemplo5',a,b,N,Ua);
Obtenemos los valores del  $u_1(t)$  y  $u_2(t)$  para  $t = \{0,1,2\}$ .
>> table(t(1:2:5), U(1:2:5,:))

```

t	$u_1(t)$	$u_2(t)$
0	-1	4

1	8.5	9.75
2	80.875	81.188

Tabla 3. Valores de $u_1(t)$ y $u_2(t)$ para $t = \{0,1,2\}$

Representamos la solución:

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

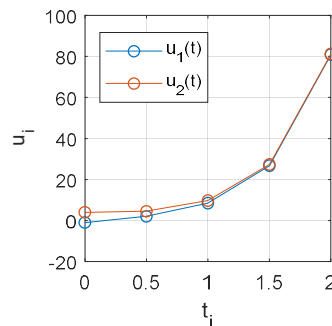


Figura 6. Representación solución PVI 5.

Método de Heun

Si en el método de Euler desarrollábamos por Taylor la función $y(t)$ hasta los términos de orden 2, en el método de Heun vamos a tomar un término más.

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

donde $\mathcal{O}(h^3)$ son 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:

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

Consideremos el desarrollo de Taylor para dos variables:

$$f(t+h, y+k) = f(t, y) + h \frac{\partial f(t, y)}{\partial t} + k \frac{\partial f(t, y)}{\partial y} + \mathcal{O}(h^2)$$

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

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

Sustituyendo en la expresión de $y(t+h)$, se obtiene:

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

Si $t_k + h = t_{k+1}$ e $y(t_k + h) = y_{k+1}$, entonces:

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

donde $k_1 = hf(t_k, y_k)$ y $k_2 = hf(t_{k+1}, y_k + k_1)$.

Teorema 3. Sea f tal que $y'(t) = f(t, y(t))$ con condición inicial $y(t_a) = y_a$. 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:

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

Y:

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

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

1. Obtención de la variable independiente discretizada
2. Inicialización del vector solución
3. Para $k=1:N$
 - i. Valor de k_1
 - ii. Valor de k_2
 - iii. Valor de $y(k+1)$
4. Fin para k
5. Preparar datos de salida

Veamos un caso de aplicación del método de Heun en el Ejemplo 6.

Ejemplo 6. Obtén la solución del PVI:

$$y'(t) = (1 - 2t)y(t), y(0) = 1, t \in [0,3]$$

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

La *function* de este PVI la implementamos en el Ejemplo 1. Apliquemos el método de Heun para los diferentes subintervalos (indicamos solo las dos primeras instrucciones):

```
>> [t2,y2]=heunPVI('ejemplo1',0,3,2,1);  
>> [t4,y4]=heunPVI('ejemplo1',0,3,4,1);
```

Obtenemos los valores de ϵ para aproximar numéricamente el orden del método sin conocer la solución analítica (indicamos solo las dos primeras instrucciones):

```
>> epsilon4=norm(y2-y4(1:2:end));  
>> epsilon8=norm(y4-y8(1:2:end));
```

Obtenemos la sucesión de los cocientes entre los valores de epsilon:

```
>> epsilon=[epsilon4    epsilon8    epsilon16
epsilon32];
>> orden=log2(epsilon(1:end-
1)./epsilon(2:end)));
```

Obtenemos los siguientes resultados.

N	ϵ	$\log_2 \left(\frac{\epsilon_N}{\epsilon_{N/2}} \right)$
2		
4	15.0776	
8	0.8813	4.0965
16	0.0982	3.1658
32	0.0183	2.4216

Tabla 4. Resultados PVI 6.

Vemos como la columna del orden va tendiendo a 2 conforme vamos duplicando el número de subintervalos.

La extensión del método de Heun para sistemas presenta modificaciones similares al método de Euler. En este caso:

$$Y_{k+1} = Y_k + \frac{1}{2}K_1 + \frac{1}{2}K_2$$

siendo:

$$K_1 = h F(t_k, Y_k), K_2 = h F(t_{k+1}, Y_k + K_1)$$

El Ejemplo 7 muestra un caso de resolución de un PVI para sistemas con el método de Heun.

Ejemplo 7. Utiliza el método de Heun para obtener la solución numérica de:

$$y''(t) - 2y'(t) + y(t) = te^t - t$$

en el intervalo $t \in [0,1]$, sabiendo que $y(0) = y'(0) = 0$ y tomando como paso $h = 0.1$. Indica el valor de la solución $y(t)$ para $t = \{0, \frac{1}{2}, 1\}$ y representa la evolución de $y(t)$.

Transformamos la ecuación diferencial de segundo orden en un sistema de ecuaciones diferenciales de primer orden.

$$\begin{aligned} \begin{bmatrix} y_1(t) \\ y_2(t) \end{bmatrix} &= \begin{bmatrix} y(t) \\ y'(t) \end{bmatrix} \rightarrow \begin{bmatrix} y_1'(t) \\ y_2'(t) \end{bmatrix} = \begin{bmatrix} y'(t) \\ y''(t) \end{bmatrix} \\ &= \begin{bmatrix} y_2(t) \\ 2y_2(t) - y_1(t) + te^t - t \end{bmatrix} \end{aligned}$$

Definimos la *function* que implementa el PVI:

```
1 function DY=ejemplo7(t,Y)
2 y1=Y(1); y2=Y(2);
3 DY=[y2; 2*y2-y1+t*exp(t)-t];
4 End
```

A continuación, definimos los parámetros de entrada relacionados con la variable independiente que requiere el método de Euler:

```
>> a=0; b=1; h=0.1; N=(b-a)/h;
```

Definimos el vector de condiciones iniciales:

```
>> Ya=[0;0];
```

Ejecutamos la función `heunPVISistema`:

```
>> [t,Y]=heunPVISistema('ejemplo7',a,b,N,Ya);
```

Obtenemos los valores del $y(t)$ para $t = \{0, \frac{1}{2}, 1\}$.

```
>> table(t(1:5:11), Y(1:5:11,1))
```

t	$y(t)$
0	0
0.5	0.006927
1	0.16624

Tabla 5. Valores del $y(t)$ para $t = \{0, \frac{1}{2}, 1\}$.

Representamos la solución:

```
>> plot(t,Y(:,1),'o-')
```

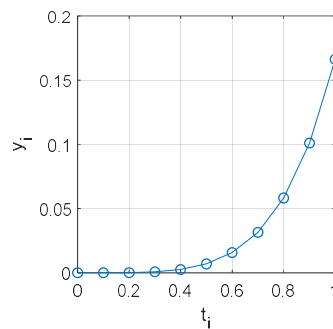


Figura 6. Representación solución PVI 7.

Método de Runge-Kutta

Los dos métodos descritos anteriormente se han basado en el desarrollo en serie de Taylor de la función $y(t)$. No obstante, también podrían haber sido deducidos a partir de determinadas fórmulas de integración numérica como vamos a hacer a continuación con el método de Runge-Kutta.

Dado el PVI:

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

podemos obtener la solución a partir de la integración como:

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

Si integramos en cada uno de los subintervalos:

$$\begin{aligned} y(t_1) &= y(a) + \int_a^{t_1} f(\tau, y(\tau)) d\tau \\ y(t_2) &= y(t_1) + \int_{t_1}^{t_2} f(\tau, y(\tau)) d\tau \\ &\vdots \\ y(b) &= y(t_N) + \int_{t_N}^b f(\tau, y(\tau)) d\tau \end{aligned}$$

De forma general:

$$y(t_{k+1}) = y(t_k) + \int_{t_k}^{t_{k+1}} f(\tau, y(\tau)) d\tau$$

Aproximando la integral por la fórmula de Simpson:

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

Obtenemos:

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

El problema es que no conocemos los términos $f\left(t_{k+\frac{1}{2}}, y_{k+\frac{1}{2}}\right)$ y $f(t_{k+1}, y_{k+1})$. Para ello, realizamos las siguientes aproximaciones:

$$f\left(t_{k+\frac{1}{2}}, y_{k+\frac{1}{2}}\right) = f\left(t_k + \frac{h}{2}, y\left(t_k + \frac{h}{2}\right)\right) \approx \frac{1}{2}(f_2 + f_3)$$

donde:

$$\begin{aligned} f_1 &= f(t_k, y_k), \\ f_2 &= f\left(t_k + \frac{h}{2}, y_k + \frac{h}{2}f_1\right) \\ f_3 &= f\left(t_k + \frac{h}{2}, y_k + \frac{h}{2}f_2\right) \end{aligned}$$

de modo que:

$$y_{k+1} \approx y_k + \frac{h}{6}(f_1 + 2f_2 + 2f_3 + f(t_{k+1}, y_k + hf_3))$$

La expresión general de método de Runge-Kutta es:

$$y_{k+1} \approx y_k + \frac{h}{6}(k_1 + 2k_2 + 2k_3 + k_4),$$

Donde:

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

Teorema 4. Sea f tal que $y'(t) = f(t, y(t))$ con condición inicial $y(t_a) = y_a$. 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:

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

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

1. Obtención de la variable independiente discretizada
2. Inicialización del vector solución
3. Para $k=1:N$
 - i. Valor de k_1
 - ii. Valor de k_2
 - iii. Valor de k_3
 - iv. Valor de k_4
 - v. Valor de $y(k+1)$
4. Fin para k
5. Preparar datos de salida

Veamos en el Ejemplo 8 un caso de aplicación del método de Runge-Kutta.

Ejemplo 8. Obtén la solución del PVI:

$$y'(t) = (1 - 2t)y(t), y(0) = 1, t \in [0,3]$$

Utiliza el método de Runge-Kutta y $N = \{2,4,8,16,32\}$ subintervalos. Obtén el error máximo sabiendo que la solución analítica es $y(t) = e^{\frac{1}{4} - (\frac{1}{2}t)^2}$ y una estimación del orden numérico.

La *function* de este PVI la implementamos en el Ejemplo 1. Apliquemos el método de Runge-Kutta para los diferentes subintervalos (indicamos solo las dos primeras instrucciones):

```
>> [t2,y2]=RK4PVI('ejemplo1',0,3,2,1);  
>> [t4,y4]=RK4PVI('ejemplo1',0,3,4,1);
```

Obtenemos la solución analítica para los diferentes intervalos (indicamos solo las dos primeras instrucciones).

```
>> yA2=exp(1/4-(1/2-t2).^2);  
>> yA4=exp(1/4-(1/2-t4).^2);
```

Calculamos el error máximo cometido en cada número de subintervalos (indicamos solo las dos primeras instrucciones):

```
>> eM2=max(abs(y2-yA2));
>> eM4=max(abs(y4-yA4));
```

Obtenemos la sucesión de los cocientes entre los errores máximos:

```
>> eM=[eM2 eM4 eM8 eM16 eM32];
>> orden=log2(eM(1:end-1)./eM(2:end));
```

Obtenemos los siguientes resultados:

N	Error máximo	$\log_2 \left(\frac{E_N}{E_{N/2}} \right)$
2	3.9979	
4	0.2292	4.1245
8	3.401e-3	6.0748
16	1.428e-4	4.5736
32	7.296e-6	4.2909

Tabla 6. Sucesión cocientes entre errores máximos.

Vemos como la columna del orden va tendiendo a 4 conforme vamos duplicando el número de subintervalos.

Para extender el método de Runge-Kutta a sistemas, debemos tener en cuenta de nuevo las consideraciones que presentamos en los casos de Euler y Heun. En el Ejemplo 9 mostramos un caso de aplicación.

Ejemplo 9. Utiliza el método de Runge-Kutta para obtener la solución numérica de:

$$y''(t) + 2y'(t) + y(t) = e^t$$

en el intervalo $t \in [0,2]$, sabiendo que $y(0) = 1$ e $y'(0) = -1$ y tomando como paso $h = 0.1$. Indica el valor de la solución $y(t)$ para $t = \{0,1,2\}$ y representa la evolución de $y(t)$.

Transformamos la ecuación diferencial de segundo orden en un sistema de ecuaciones diferenciales de primer orden:

$$\begin{aligned} \begin{bmatrix} y_1(t) \\ y_2(t) \end{bmatrix} &= \begin{bmatrix} y(t) \\ y'(t) \end{bmatrix} \rightarrow \begin{bmatrix} y_1'(t) \\ y_2'(t) \end{bmatrix} = \begin{bmatrix} y'(t) \\ y''(t) \end{bmatrix} \\ &= \begin{bmatrix} y_2(t) \\ -2y_2(t) - y_1(t) + e^t \end{bmatrix} \end{aligned}$$

Definimos la *function* que implementa el PVI:

```
1 function DY=ejemplo9(t,Y)
2 y1=Y(1); y2=Y(2);
3 DY=[y2; -2*y2-y1+exp(t)];
4 end
```

A continuación, definimos los parámetros de entrada relacionados con la variable independiente que requiere el método de Euler:

```
>> a=0; b=2; h=0.1; N=(b-a)/h;
```

Definimos el vector de condiciones iniciales:

```
>> Ya=[1; -1];
```

Ejecutamos la función RK4PVISistema:

```
>> [t,Y]=RK4PVISistema('ejemplo9',a,b,N,Ya);
```

Obtenemos los valores del $y(t)$ para $t = \{0,1,2\}$.

```
>> table(t(1:10:21), Y(1:10:21,1))
```

t	$y(t)$
0	1
1	0.77154
2	1.8134

Tabla 7. Valores del $y(t)$ para $t = \{0,1,2\}$.

Representamos la solución:

```
>> plot(t,Y(:,1),'o-')
```

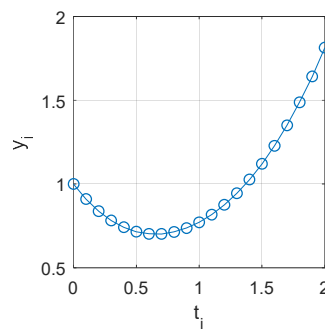


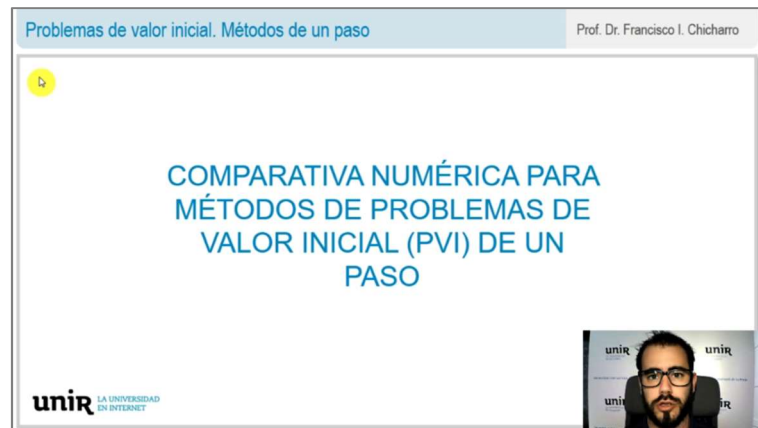
Figura 7. Solución PVI ejemplo 9.

Lo + recomendado

Lecciones magistrales

Comparativa numérica para métodos de problemas de valor inicial (PVI) de un paso

En esta lección magistral vamos a realizar una comparativa numérica de los tres métodos de resolución de PVI que hemos visto en este tema: Euler, Heun y Runge-Kutta.



Accede a la lección magistral a través del aula virtual

No dejes de leer

Métodos numéricos para ingenieros

Chapra, S. (2007). *Métodos numéricos para ingenieros* (5a. Ed.). Madrid: McGraw-Hill.

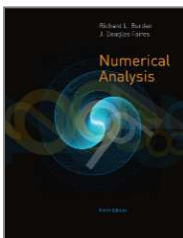


En el capítulo 25 de este libro se resuelven ecuaciones diferenciales ordinarias a partir de métodos numéricos. Además, aparecen los métodos que hemos implementado a lo largo de este Tema.

Accede al libro a través de la Biblioteca Virtual de UNIR

Numerical analysis

Burden, R. L. y Faires, J. D. (2011). *Numerical analysis* (9ª ed). Boston: Brooks/Cole CENGAGE learning.



El libro *Numerical Analysis* es el libro de referencia de esta asignatura. En la primera parte del capítulo 5 se presentan los problemas de valor inicial, con multitud de ejemplos y algoritmos para la implementación de los diferentes métodos numéricos de un paso.

A fondo

Derivación de tres métodos de tipo Runge-Kutta

Özis, T. y Turaci, M. Ö. (2017). Derivation of three-derivative Runge-Kutta methods. *Numerical Algorithms* 74, 247—265.



En este artículo de investigación se puede observar cómo se generan nuevos métodos de tipo Runge-Kutta. Asimismo, se presentan tablas numéricas que recogen los resultados de aplicar los nuevos métodos implementados sobre una serie de problemas.

Accede al libro a través de la Biblioteca Virtual de UNIR

Webgrafía

Maths Tools

En los métodos de tipo Runge-Kutta, los diferentes valores de k_i están multiplicados por un coeficiente. Estos coeficientes se pueden obtener a partir de las tablas de Butcher para generar métodos de Runge-Kutta de cualquier orden. Podemos encontrar esto en la página web Maths Tools.

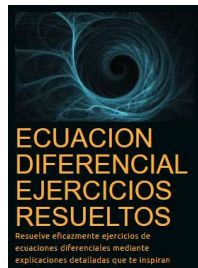


Accede a la página web a través del aula virtual o desde la siguiente dirección web:

[https://www.mathstools.com/section/main/Metodos de Runge Kutta?lang=es#.XC9F11xKjIU](https://www.mathstools.com/section/main/Metodos_de_Runge_Kutta?lang=es#.XC9F11xKjIU)

Ejercicios resueltos de ecuaciones diferenciales

En esta página puedes encontrar ejercicios resueltos de ecuaciones diferenciales. Además, en el enlace puedes acceder a la explicación de los métodos numéricos de un paso impartidos en este tema.



Accede a la página web a través del aula virtual o desde la siguiente dirección web:

<https://ecuaciondiferencialejerciciosresueltos.com/tag/metodo-numerico>

Soluciones analíticas de problemas de valor inicial

Wolfram es la empresa propietaria del *software* Mathematica. Además, también tiene disponible el portal WolframAlpha, en el cual se puede introducir cualquier expresión matemática y el mismo portal obtiene la solución. El uso de este portal es un buen método para conocer la solución analítica de un problema de valor inicial y compararla con la solución numérica, de forma que se pueda verificar si ambas soluciones se aproximan.



Accede a la página web a través del aula virtual o desde la siguiente dirección web:

www.wolframalpha.com

Bibliografía

Cordero, A., Hueso, J. L., Martínez, E. y Torregrosa, J. R. (2006). *Problemas resueltos de métodos numéricos*. Madrid: Thomson.

Cordero, A., Hueso, J. L., Martínez, E. y Torregrosa, J. R. (2005). *Métodos numéricos con MATLAB*. Valencia: Universidad Politécnica de Valencia.

Mathews, J. H., Fink, K. D. (2000). *Métodos numéricos con MATLAB*. Madrid: Prentice-Hall.

Zill, D. G. (2016). *Advanced Engineering Mathematics*. Sudbury: Jones & Bartlett Learning.

1. Un problema de valor inicial se puede definir a partir de una ecuación diferencial y:
 - A. Una condición en el instante inicial sobre la función incógnita.
 - B. Una condición en un instante intermedio sobre la función incógnita.
 - C. Una condición en el instante final sobre la función incógnita.

2. En un sistema de problemas de valor inicial, dado por un sistema de ecuaciones diferenciales, es necesario conocer:
 - A. El valor de una de las funciones incógnita en el instante inicial.
 - B. El valor de todas las funciones incógnita menos una en el instante inicial.
 - C. El valor de todas las funciones incógnita en el instante inicial.

3. Una PVI definido por una ecuación diferencial de orden mayor que uno se puede resolver a través de los métodos para resolución de PVI a través de:
 - A. Un cambio de variables.
 - B. Métodos analíticos, exclusivamente.
 - C. No se puede resolver.

4. ¿Cuál de los siguientes métodos tiene menor orden?
 - A. Euler.
 - B. Heun.
 - C. Runge-Kutta.

5. ¿Cuál de los siguientes métodos tiene mayor orden?
- A. Euler.
 - B. Heun.
 - C. Runge-Kutta.
6. Sea el PVI dado por $y''(t) = f(t, y(t), y'(t)), t \in [a, b], y(a) = y_a, y'(a) = y'_a$. Al ejecutar `[t,Y]=eulerPVI sistemas(f,a,b,N,Ya)`, ¿cómo puedo representar la función $y'(t)$?
- A. `plot(t,Y)`.
 - B. `plot(t,Y(:,1))`
 - C. `plot(t,Y(:,2))`
7. Conocida la solución analítica de un PVI, ¿cómo puedo obtener una estimación numérica del orden del método utilizado?
- A. A partir del error mínimo entre la solución analítica y la solución numérica.
 - B. A partir del error máximo entre la solución analítica y la solución numérica.
 - C. A partir del error promedio entre la solución analítica y la solución numérica.
8. Sin conocer la solución analítica de un PVI, ¿cómo puedo obtener una estimación numérica del orden del método utilizado?
- A. Utilizando la norma de la diferencia entre las soluciones obtenidas con N y $2N$ subintervalos.
 - B. Utilizando el valor absoluto de la diferencia entre las soluciones obtenidas con N y $2N$ subintervalos.
 - C. No es posible obtener una estimación numérica del método.
9. Sea el PVI $y'(t) = y(t) - t^2 + 1, t \in [0,2], y(0) = 0.5, h = 1$. ¿Qué método da como solución $y(t) = [0.5, 2, 4]$?
- A. Euler.
 - B. Heun.
 - C. Runge-Kutta.

10. Sea el PVI $t^2 y''(t) - 2ty'(t) + 2y(t) = t^3 \ln t, t \in [1,2], y(1) = 1, y'(1) = 0, h = 0.5$. ¿Qué método da como solución $y'(t) = [0, -0.8206, -1.0933]$?
- A. Euler.
 - B. Heun.
 - C. Runge-Kutta.