

# Ecuaciones Diferenciales Ordinarias

Métodos Numéricos y Simulación.

Segundo de Grado en Física.

# Métodos explícitos de un solo paso

En la clase anterior estudiamos distintas versiones de métodos de Runge-Kutta para resolver ecuaciones diferenciales del tipo:

$$\frac{dy}{dx} = f(x, y; k)$$

Todas las variantes del método de Runge-Kutta se clasifican como **métodos explícitos de un solo paso**.

- **Explícitos**, porque sólo requieren evaluaciones de la función  $f(x, y)$  en puntos cuyos valores de  $x$  e  $y$  son conocidos
- **De un solo paso**, porque para evaluar  $y_{j+1} = y(x_{j+1})$  sólo necesitamos conocer  $x_j$  e  $y_j$ . Todos los restantes valores que necesitemos para encontrar  $x_{j+1}, y_{j+1}$  los determinamos a partir de  $x_j$  e  $y_j$ .

# Métodos multipaso

Supongamos que estamos resolviendo la ecuación diferencial ordinaria:

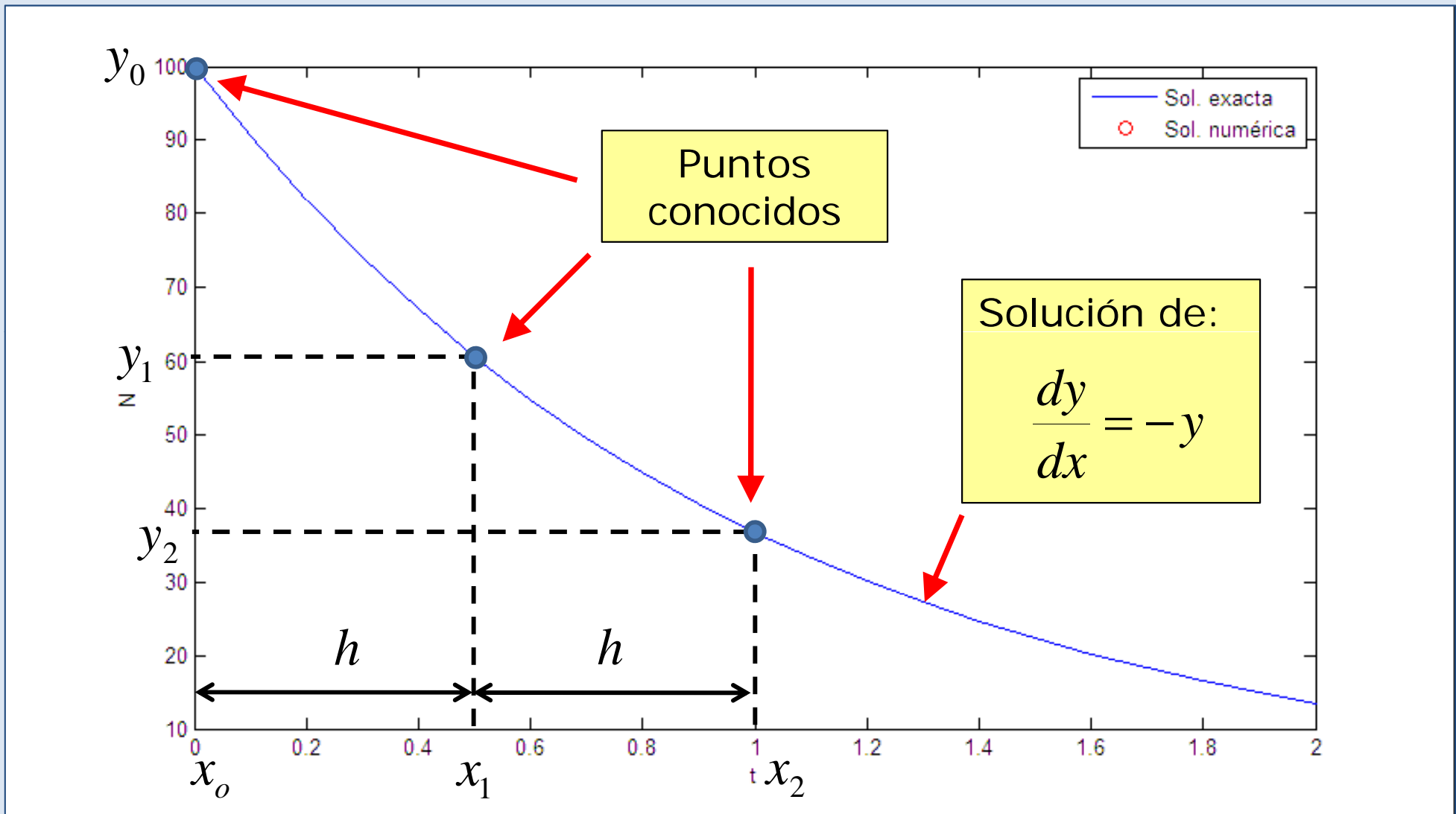
$$\frac{dy}{dx} = f(x, y; k)$$

Y que conocemos los puntos  $(x_0, y_0)$ ,  $(x_1, y_1)$ , ...,  $(x_j, y_j)$ :

- Si usamos un método de Runge-Kutta de orden  $N$ , para pasar del punto  $(x_j, y_j)$  al  $(x_{j+1}, y_{j+1})$  **al menos tengo que hacer  $N$  evaluaciones de la función  $f(x, y)$ .**
- Una estrategia alternativa puede ser usar los puntos  $(x_{j-N}, y_{j-N})$ ,  $(x_{j-N+1}, y_{j-N+1})$ , ...,  $(x_j, y_j)$
- Cuando se sigue esta estrategia, se dice que se usa un **método multipaso**.

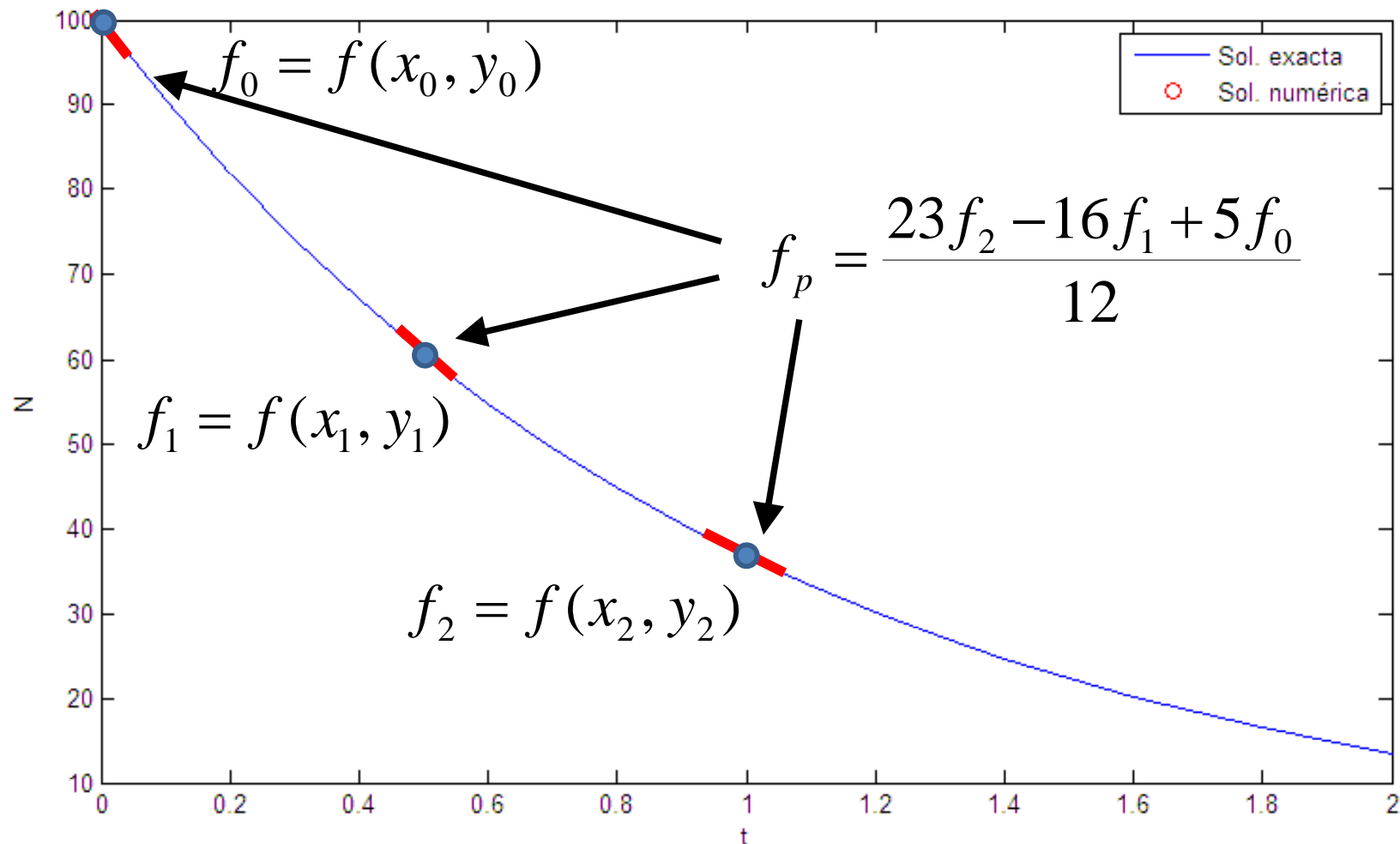
# Ejemplo: método de Adams-Basforth de 3<sup>er</sup> orden (I)

Supongo conocidos los puntos:  $(x_0, y_0)$ ,  $(x_1, y_1)$ ,  $(x_2, y_2)$  del problema de la desintegración radiactiva:



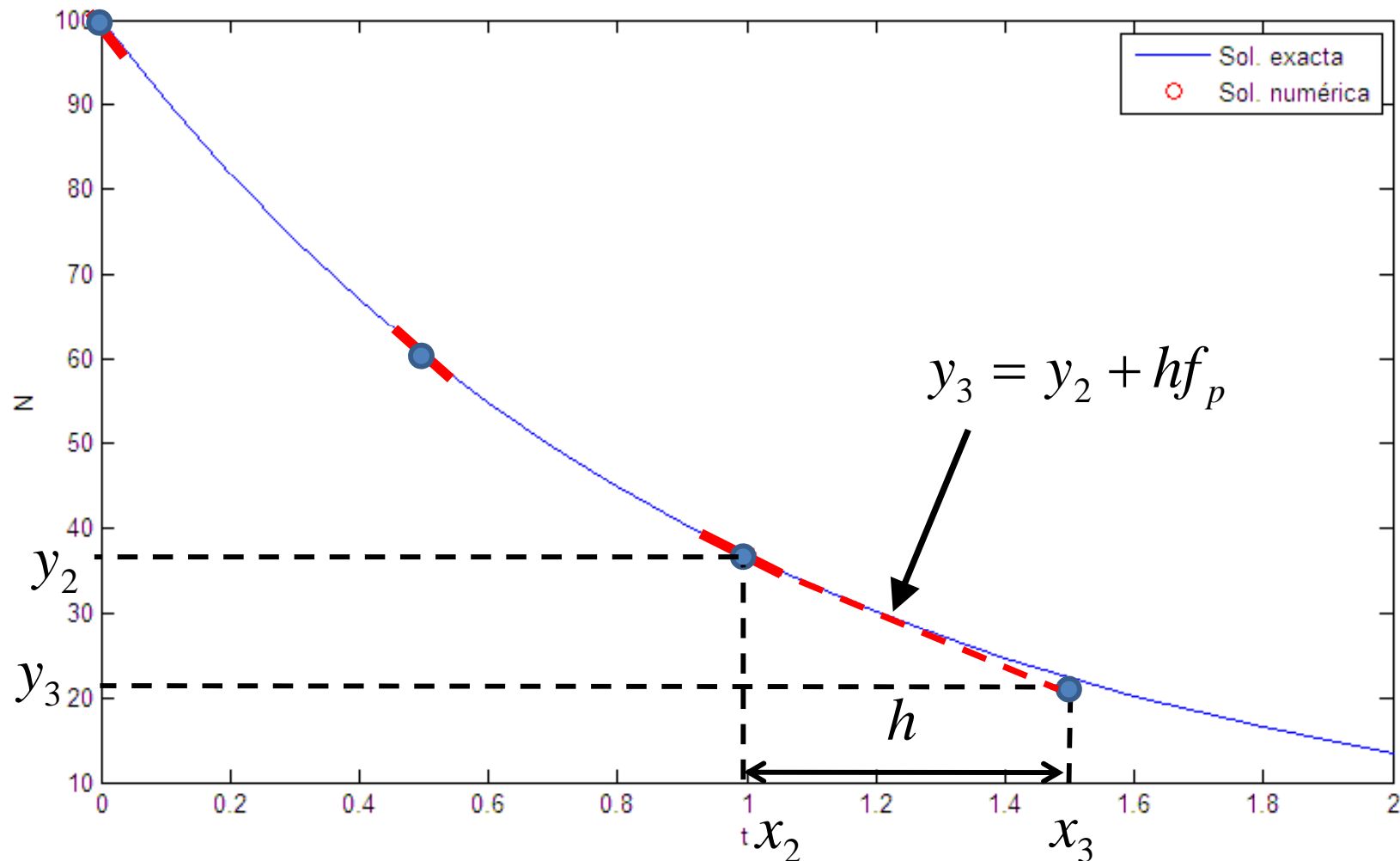
# Ejemplo: método de Adams-Basforth de 3<sup>er</sup> orden (II)

Estimamos la derivada de la solución entre  $x_2$  y el  $x_3$  a partir de las derivadas en los puntos que conocemos.



# Ejemplo: método de Adams-Basforth de 3<sup>er</sup> orden (III)

Avanzamos la solución un paso usando nuestra estimación de la derivada entre  $x_2$  y  $x_3$ .



# Método de Adams-Balforth de 3er orden: implementación en Matlab.

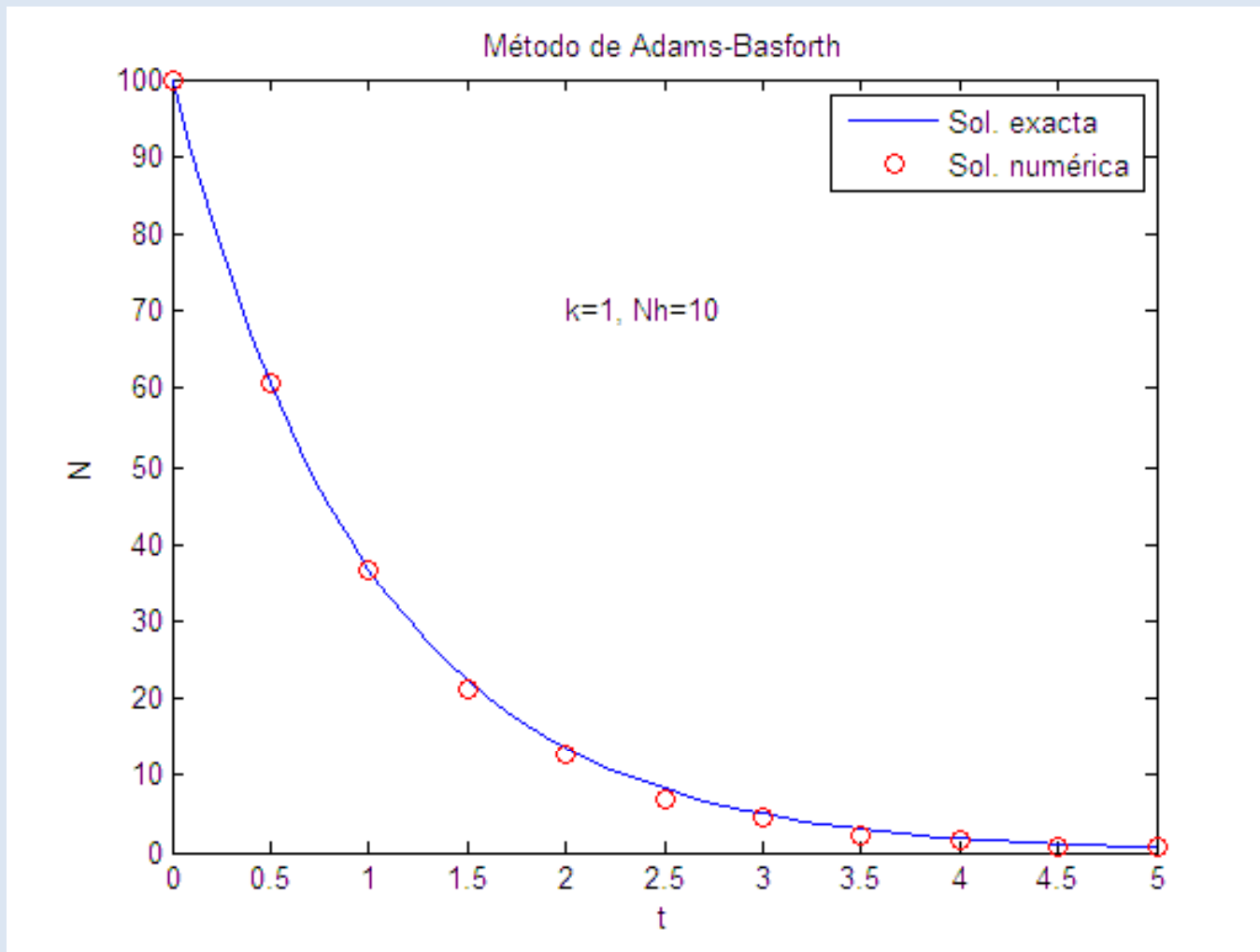
```
1 function [x y]=fAdamsBas3 (fun,x0,xf,y0,Nh)
2 - h = (xf-x0)/Nh;
3 - x=zeros(1,Nh+1); % Vector para los valores de x
4 - y=zeros(1,Nh+1); % Vector para los valores de y
5 - f=zeros(1,Nh+1); % Vector para los valores de la derivada dy/dx
6 - x(1)=x0;
7 - y(1)=y0;
8 - f(1)=fun(x0,y0);
9 % Necesitamos tres puntos. Tengo que usar un método de un sólo paso.
10 % Uso el método de Runge-Kutta de cuarto orden.
11 - Naux = 2;
12 - xfaux = x0+2*h;
13 - [xaux,yaux]=fRK4 (fun,x0,xfaux,y0,Naux);
14 - x(2)=xaux(2);
15 - y(2)=yaux(2);
16 - f(2)= fun(x(2),y(2));
17 - x(3)=xaux(3);
18 - y(3)=yaux(3);
19 - f(3)=fun(x(3),y(3));
20 % Ya tenemos tres puntos. Podemos usar el método de Adams-Basforth
21 - for j=4:1:Nh+1
22 - y(j) = y(j-1) + h/12*(23*f(j-1)-16*f(j-2)+5*f(j-3));
23 - x(j) = x(j-1) + h;
24 - f(j) = fun(x(j),y(j));
25 - end
26 - end
```

Aparte de x e y almacenamos los valores de la derivada  $f(x,y)$ .

Necesitamos usar un método de un solo paso hasta conocer tres puntos de la solución.

Aquí usamos valores conocidos de la derivada  $f(x,y)$ .

# Adams-Basforth: ejemplo



**El método de Adams-Basforth de tercer orden es un método multipaso explícito.**



# Ventajas de los métodos multipaso.

- Es fácil aumentar el orden del método: sólo hay que considerar más puntos previos. Con esta estrategia se puede:
  - Crear métodos de orden muy alto. Se usan cuando se requiere mucha precisión en la solución.
  - Crear métodos de orden variable, considerando más o menos puntos previos según nos convenga.
- Si están programados de forma que se almacenen los valores previos de  $f(x,y)$ , requieren menos evaluaciones de  $f(x,y)$  que los métodos de un solo paso.
  - Esto los hace más eficientes (ahorro de tiempo) que los métodos de un solo paso cuando la solución no tiene variaciones bruscas y  $f(x,y)$  es complicada de evaluar.

# Métodos implícitos

Se llaman métodos implícitos aquellos en los que hay que hacer evaluaciones de la función  $f(x,y)$  en puntos en los que no conocemos el valor de  $y$ .

Por ejemplo, en el **método de Euler explícito**, hacemos:

$$(1) \quad x_{j+1} = x_j + h \quad (2) \quad f_1 = f(x_j, y_j) \quad (3) \quad y_{j+1} = y_j + f_1 h$$

En cambio, en el **método de Euler implícito**, hacemos:

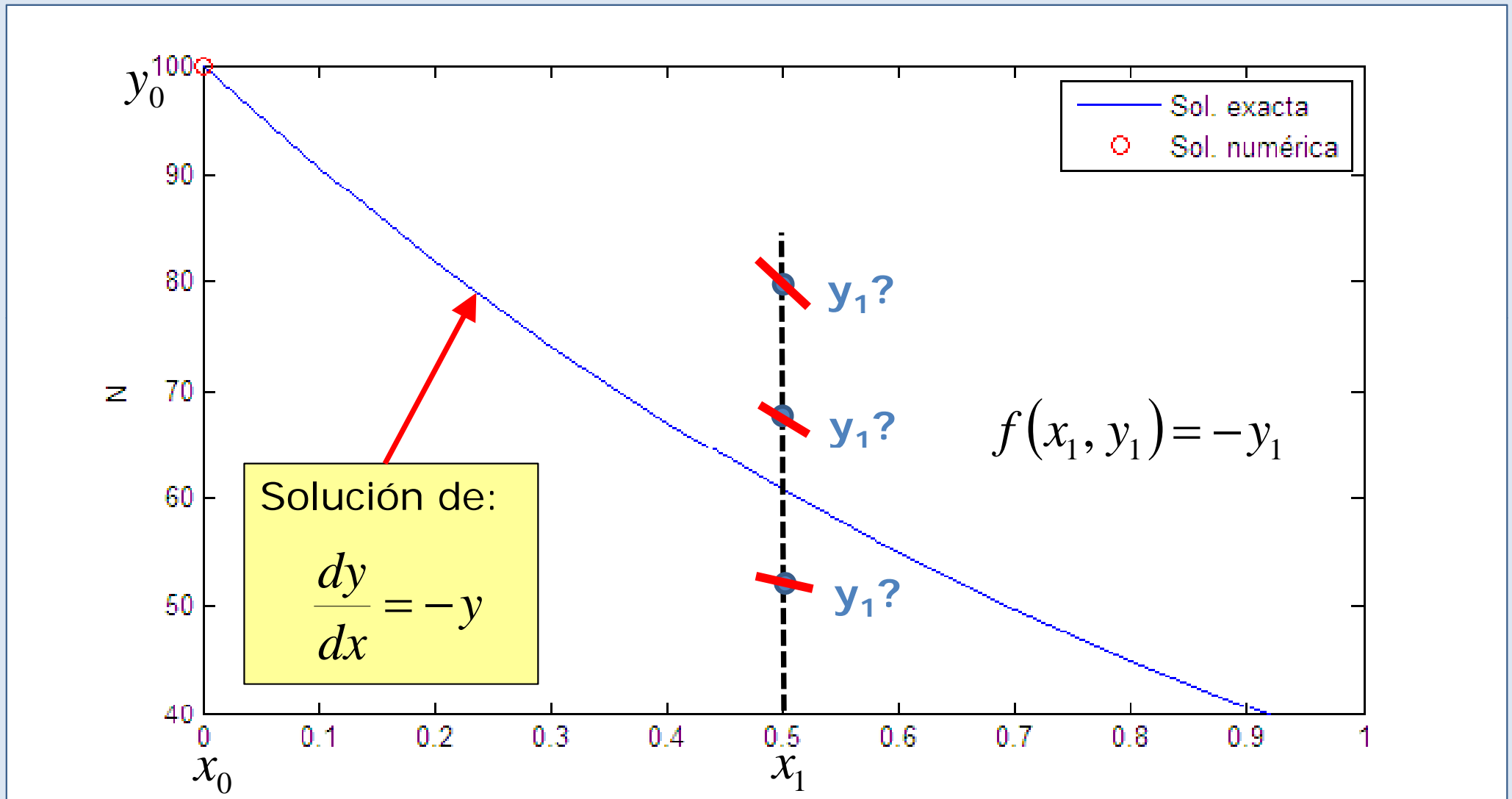
$$(1) \quad x_{j+1} = x_j + h \quad (2) \quad f_1 = f(x_{j+1}, y_{j+1}) \quad (3) \quad y_{j+1} = y_j + f_1 h$$

Como en el paso (2) no conocemos  $y_{j+1}$ , el paso (3) implica **resolver una ecuación algebraica no lineal**.

# Método de Euler implícito (I)

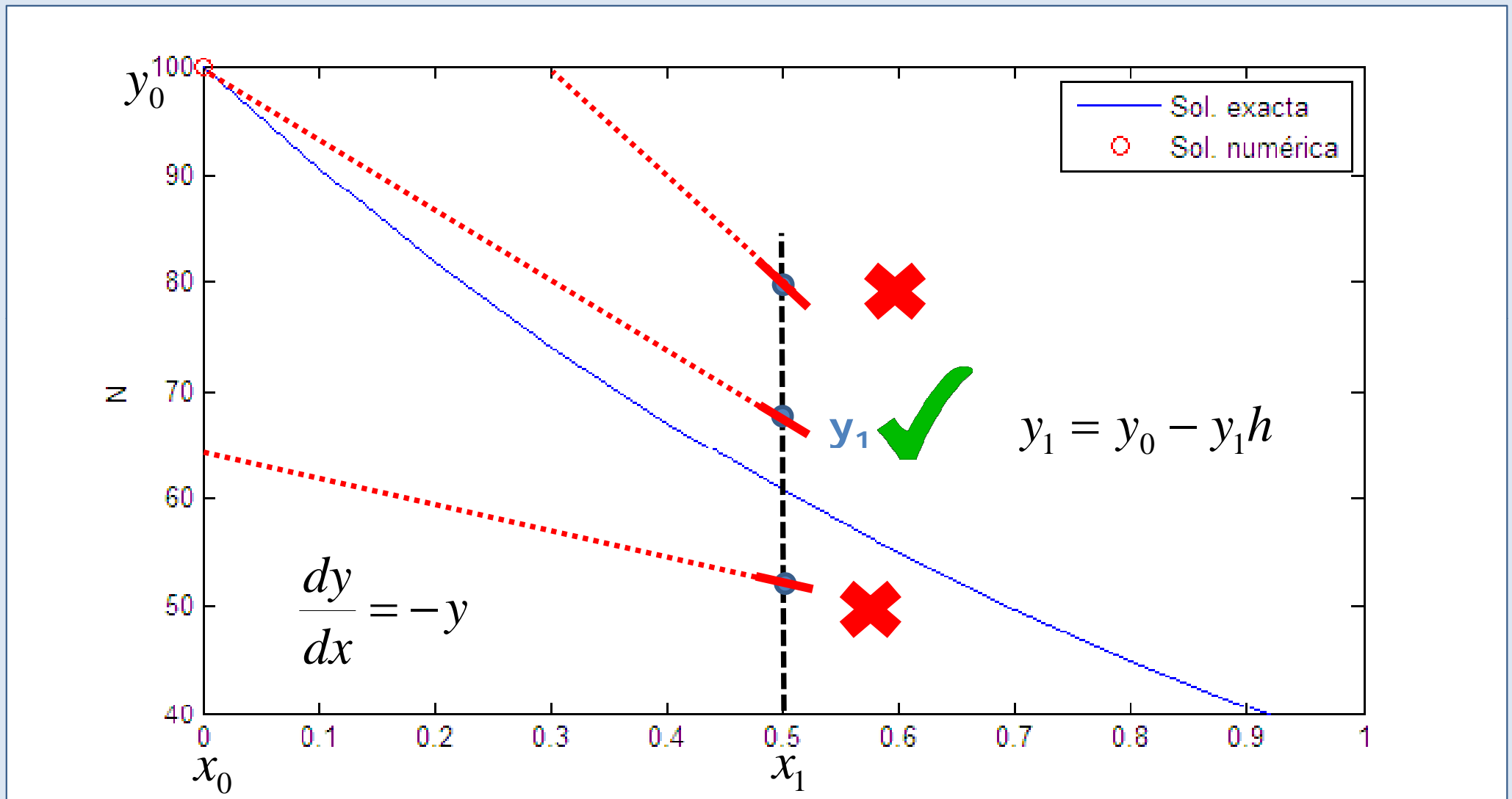
Buscamos un punto que cumpla:

$$y_1 = y_0 + f(x_1, y_1)h$$

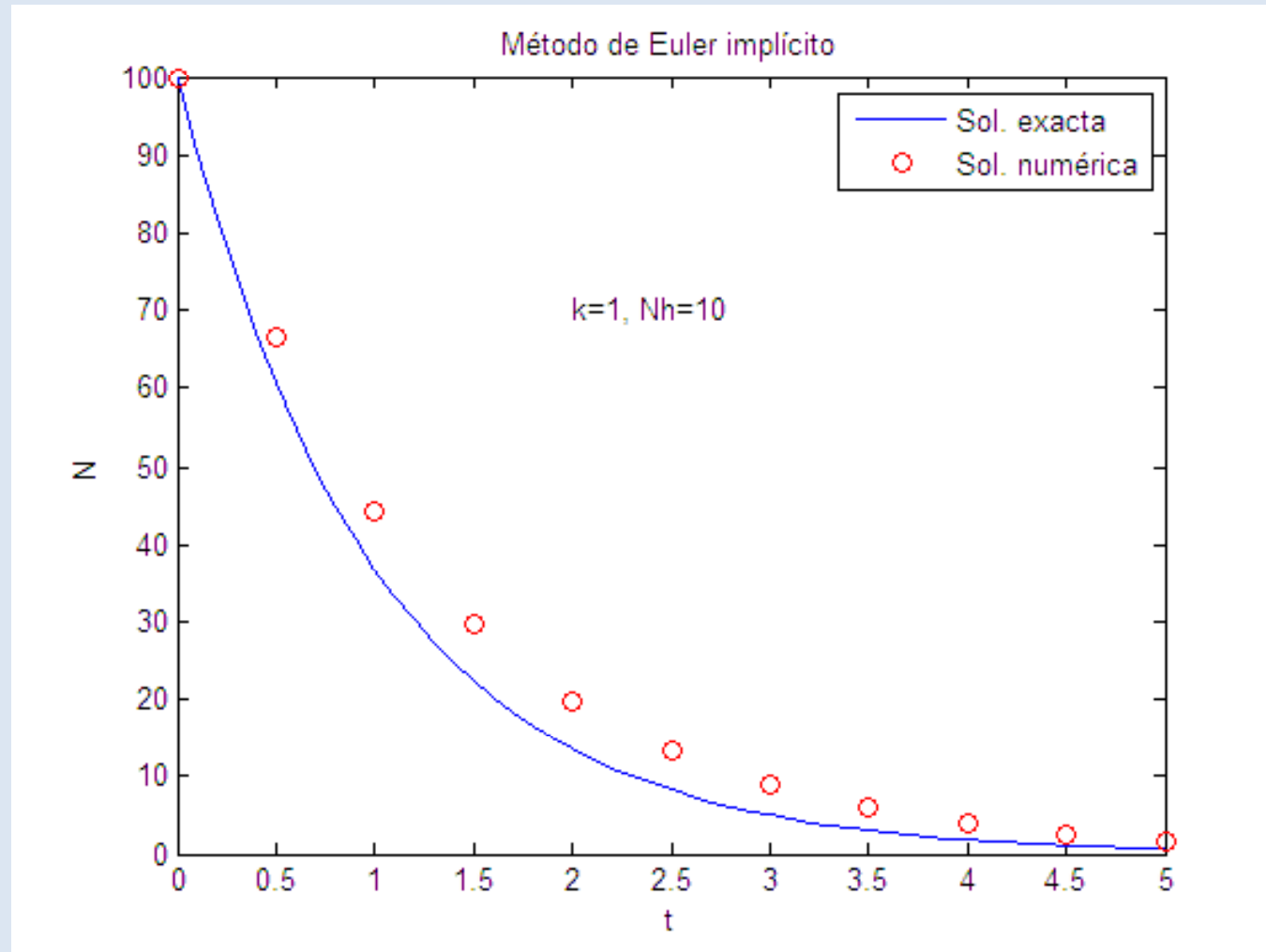


# Método de Euler implícito (II)

El punto  $(x_1, y_1)$  que aceptamos es aquel para el  $f(x_1, y_1)$  es la pendiente de la recta que lo conecta con el punto  $(x_0, y_0)$



# Método de Euler implícito: resultados



**NOTA:** el método de Euler implícito es un método de primer orden, como su equivalente explícito.

# Método de Euler implícito: implementación en Matlab.

```
1 function [x,y]=bfeuler(fun,x0,xf,y0,Nh)
2 - h=(xf-x0)/Nh;
3 - x=zeros(1,Nh+1);
4 - y=zeros(1,Nh+1);
5 - x(1)=x0;
6 - y(1)=y0;
7 - for j=2:1:Nh+1
8 -     x(j) = h+x(j-1);
9 -     g = @(s) s - y(j-1) - h*fun(x(j),s);
10 -     y(j) = fsolve(g,y(j-1));
11 - end
12 - end
```

Ecuación de la que se obtiene  $y(j)$  a partir de  $y(j-1)$

Uso la función **fsolve** de la **Optimization Toolbox** de Matlab para resolver la ecuación algebraica no lineal  $g(s)=0$ .

Solución de  $g(s)=0$ .

Valor cercano a la solución de  $g(s)=0$

# Métodos implícitos de orden superior.

Existen métodos implícitos de orden superior a 1. Por ejemplo, el método del trapecio o método de Crank-Nicolson:

$$y_{j+1} = y_j + \frac{h}{2} [f(x_j, y_j) + f(x_{j+1}, y_{j+1})]$$

relación implícita

El método del trapecio es un algoritmo implícito de un solo paso de orden 2.

Conforme aumenta el orden del método aumenta el número de evaluaciones de la función  $f(x, y)$ .

Por este motivo, **no se suelen usar métodos implícitos de orden muy alto.**

# Ventajas de los métodos implícitos

Los métodos implícitos son superiores a los métodos explícitos para aquellos problemas que se dicen **rígidos**.

Cuando entre los términos de la solución de una EDO, al menos uno depende del tiempo muy rápidamente y el resto lentamente, se dice que el problema es **rígido**.

- En los métodos explícitos, la estabilidad de la solución numérica depende del término que cambia con el tiempo con mayor rapidez.
  - En los métodos explícitos, el paso  $h$  lo controla el término que varía más rápidamente con el tiempo.
  - Si sólo nos interesa el término que varía lentamente con el tiempo, esto nos puede obligar a hacer muchos pasos.
- **Los métodos implícitos no tienen esta limitación.**



# Ejemplo de problema rígido con solución analítica.

Consideremos la ecuación diferencial ordinaria:

$$\frac{dy}{dx} = -ay + bx \quad y(0) = y_0$$

**Solución analítica:**

$$y(x) = \left( y_0 + \frac{b}{a^2} \right) e^{-ax} + \frac{b}{a} x - \frac{b}{a^2}$$

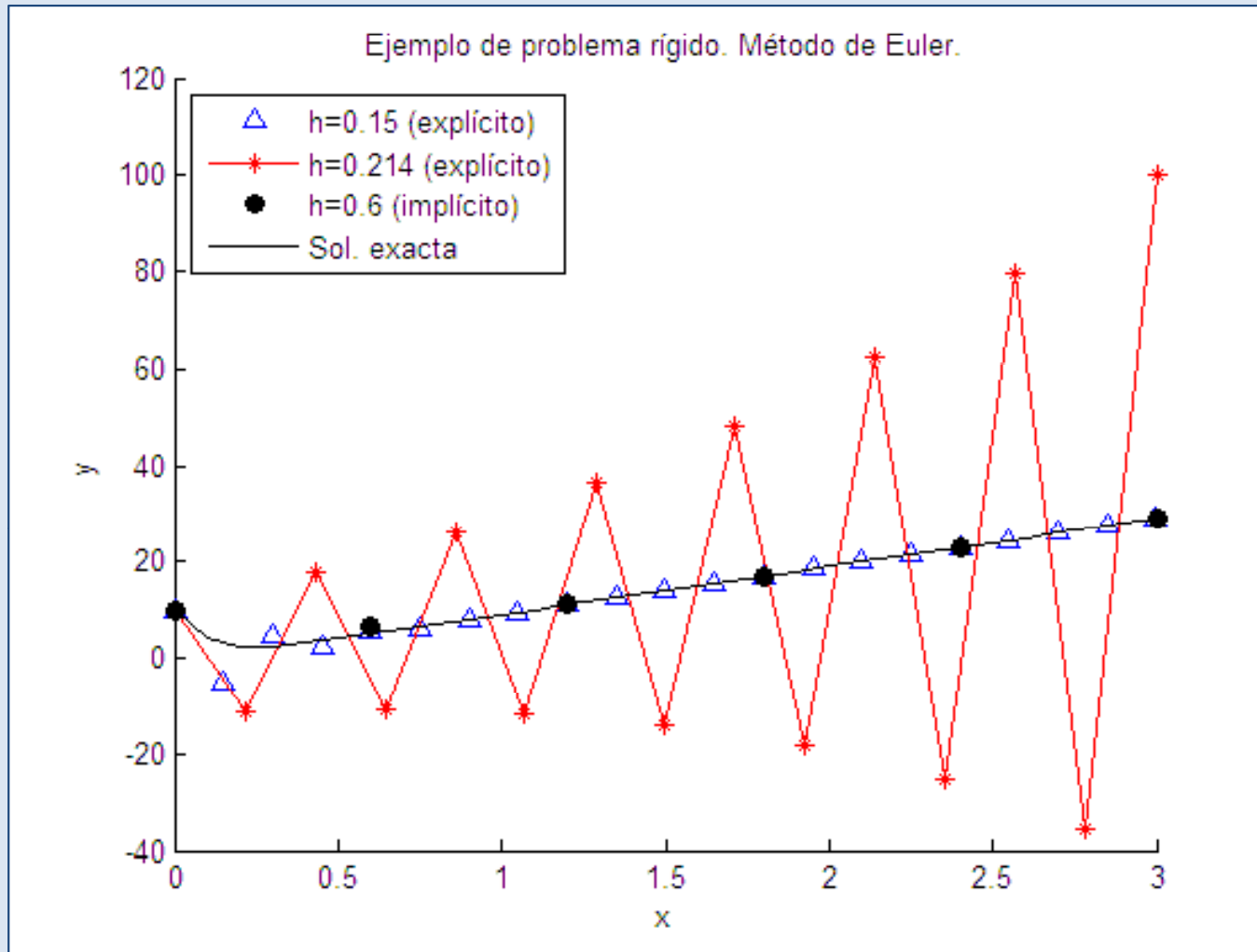
**Término transitorio.** Este término varía rápidamente con el tiempo si  $a > 1$ .

**Término estacionario.** Este término varía lentamente con el tiempo.

Si intentamos resolver este problema por el método de Euler explícito, el paso  $h$  debe ser menor que  $2/a$  para que el método sea estable.

Si  $a$  es grande y sólo nos interesa conocer la parte estacionaria de la solución, esto nos impone un paso prohibitivamente pequeño.

# Comparativa: método de Euler implícito y explícito.



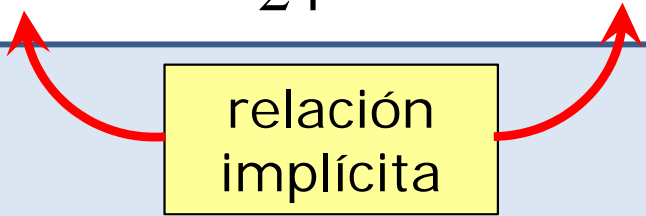
**El método de Euler implícito permite un tamaño de paso más grande que el método de Euler explícito.**

# Métodos implícitos multipaso

Al igual que para los métodos explícitos, también existen métodos implícitos multipaso.

Ejemplo: Método de Adams-Moulton de cuarto orden.

$$y_{j+1} = y_j + \frac{h}{24} [9f(x_{j+1}, y_{j+1}) + 19f(x_j, y_j) - 5f(x_{j-1}, y_{j-1}) + f(x_{j-2}, y_{j-2})]$$



relación  
implícita

Su fundamento es el mismo que el de los métodos multipaso explícitos pero involucran también la derivada en el punto final del paso.

# Métodos predictor-corrector

Estos métodos combinan un método explícito y un método implícito. Por ejemplo, método de Adams-Basforth-Moulton.

(1) Etapa predictora: Método de Adams-Basforth (explícito).

$$y_{j+1} = y_j + \frac{h}{12} [23f(x_j, y_j) - 16f(x_{j-1}, y_{j-1}) + 5f(x_{j-2}, y_{j-2})]$$

(2) Etapa correctora: Método de Adams-Moulton (implícito).

$$y_{j+1} = y_j + \frac{h}{24} [9f(x_{j+1}, y_{j+1}) + 19f(x_j, y_j) - 5f(x_{j-1}, y_{j-1}) + f(x_{j-2}, y_{j-2})]$$

De esta forma no tenemos que resolver una ecuación algebraica no lineal y conservamos el orden de la etapa correctora, pero el método se hace explícito y perdemos las ventajas de los algoritmos implícitos.

# Métodos predictor-corrector en Matlab

***[x,y]=ode113(odefun,xspan,y0,options)***

ode[orden del método][orden en el que se evalúa el error]

La función ode113 de Matlab usa un algoritmo predictor-corrector combinado con un control de paso variable y un mecanismo para que el orden del método sea variable.

El orden del método varía entre 1 y 13 dependiendo de la precisión requerida.

Su uso se recomienda en aquellos problemas que requieran mucha precisión en la solución y en los que la derivada de la función incógnita varíe suavemente.

# Algoritmos para problemas rígidos

- Métodos implícitos multipaso.

$$[x,y]=ode15s(odefun,xspan,y0,options)$$

Es un método multipaso implícito de paso variable y orden variable entre 1 y 5.

- Métodos implícitos de un solo paso.

$$[x,y]=ode23s(odefun,xspan,y0,options)$$
$$[x,y]=ode23t(odefun,xspan,y0,options)$$
$$[x,y]=ode23tb(odefun,xspan,y0,options)$$

# Bibliografía del Tema.

Numerical Recipes,  
<http://www.nr.com/oldverswitcher.html>

A. Quarterioni, F. Salieri, Cálculo científico con Matlab y Octave (capítulo 7) , Springer-Verlag (2006), ISBN 10 88-470-0503-5

Cleve Moler. Numerical computing with Matlab.  
<http://www.mathworks.com/moler/chapters.html>

Abramowitz and Stegun. Handbook of mathematical functions.  
[http://www.nrbook.com/abramowitz\\_and\\_stegun/](http://www.nrbook.com/abramowitz_and_stegun/)

W.E. Boyce, R.C DiPrima. Elementary differential equations and boundary value problems. Wiley and sons (2010, 9<sup>a</sup> edición), ISBN 978-0-470-39873-9.