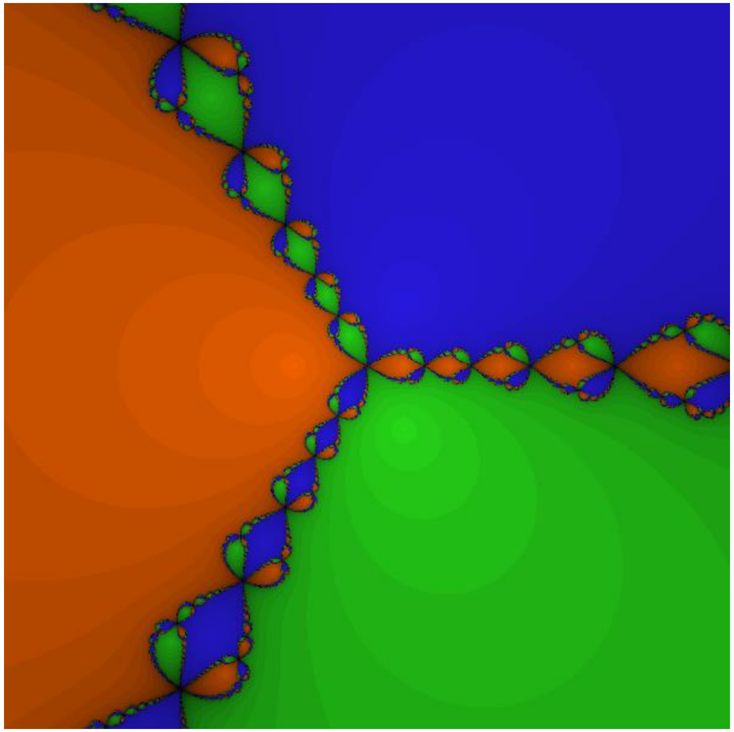
Métodos Numéricos Avanzados en Ingeniería

Alicia Cordero

Neus Garrido

Juan R. Torregrosa



Tema 4: Problemas de valor inicial. Métodos multipaso

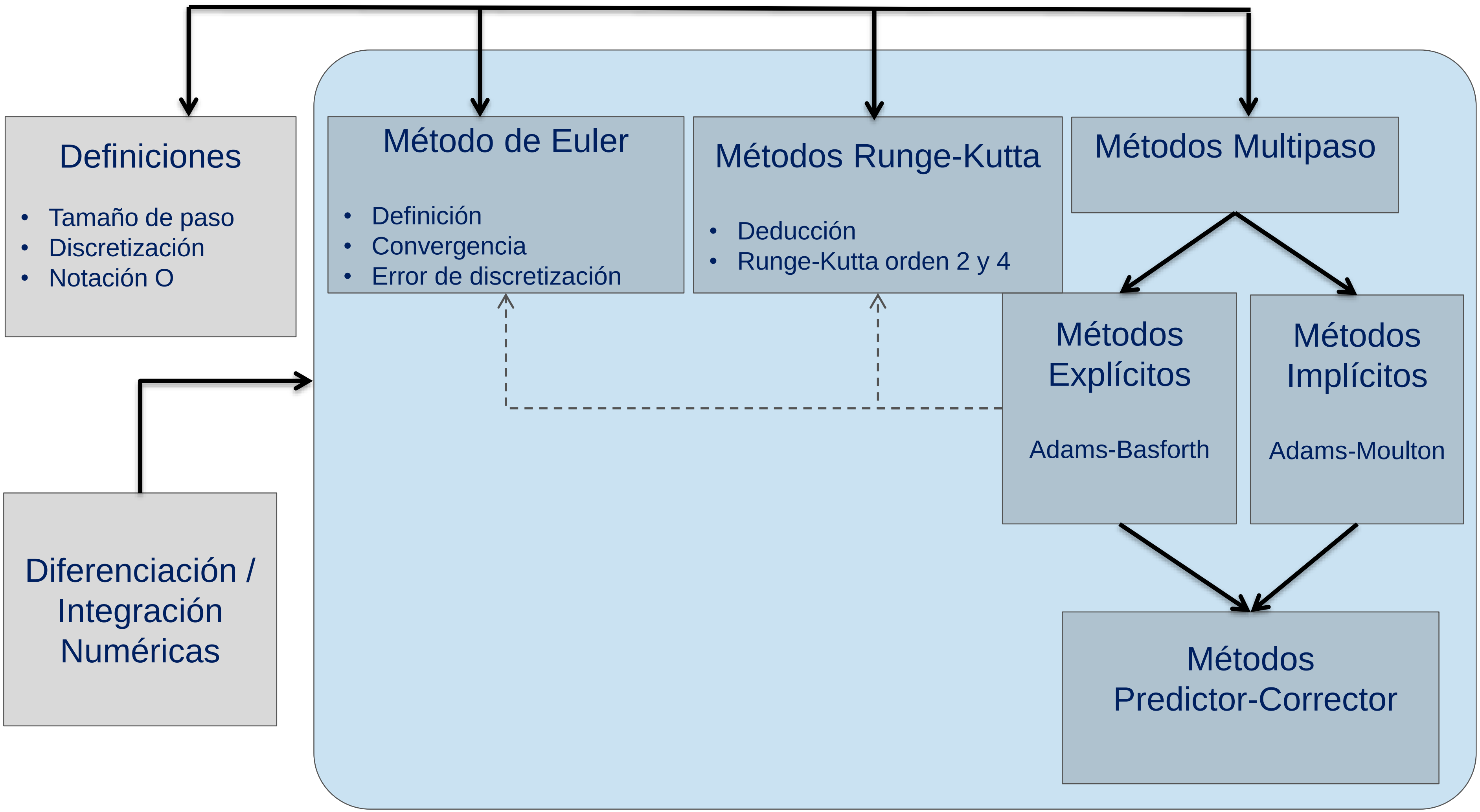
LA UNIVERSIDAD EN INTERNET

Universidad Internacional de La Rioja

Ecuaciones diferenciales de primer orden

- Métodos multipaso
  - Explícitos: Adams-Bashforth
  - Implícitos: Adams-Moulton
  - Predictor-corrector
- Ecuaciones rígidas

Métodos numéricos para problemas de valor inicial



Métodos multipaso

Planteamiento general

- En general,

$$\boxed{y(x_{k+1}) = y(x_k) + \int_{x_k}^{x_{k+1}} f(\mathrm{t}, y(t))\, dt}$$

Para aproximar la integral se sustituye el integrando por un polinomio interpolador que pasa por los puntos:

- Adams- Bashforth (explícito, más simple)

$$\{(x_{k-n}, f(x_{k-n}, y_{k-n})), (x_{k-n+1}, f(x_{k-n+1}, y_{k-n+1})), \dots, (x_k, f(x_k, y_k))\}$$

- Adams-Moulton (implícito, más estables)

$$\{(x_{k-n}, f(x_{k-n}, y_{k-n})), (x_{k-n+1}, f(x_{k-n+1}, y_{k-n+1})), \dots, (x_{k+1}, f(x_{k+1}, y_{k+1})$$

Adams-Bashforth de dos pasos

- Método de dos pasos (orden 2) AB2

Partimos de dos puntos $\{(x_{k-1}, f(x_{k-1}, y_{k-1})), (x_k, f(x_k, y_k))\}$.

Calculamos el polinomio interpolador que pasa por ellos:

$$\boxed{\begin{array}{lll} f(x_k, y_k) & \searrow & \\ f(x_{k-1}, y_{k-1}) & \rightarrow & f[x_k, x_{k-1}] \end{array}}$$

$$\begin{aligned} p(t) &= f(x_k, y_k) + \frac{f(x_k, y_k) - f(x_{k-1}, y_{k-1})}{x_k - x_{k-1}}(t - x_k) \\ &= f(x_k, y_k)\left(1 + \frac{t - x_k}{x_k - x_{k-1}}\right) + f(x_{k-1}, y_{k-1})\frac{x_k - t}{x_k - x_{k-1}} \\ &= f(x_k, y_k)\frac{t - x_{k-1}}{h} + f(x_{k-1}, y_{k-1})\frac{x_k - t}{h} \end{aligned}$$

Adams-Bashforth de dos pasos

- Método de dos pasos (orden 2) AB2

Partimos de dos puntos$\{(x_{k-1}, f(x_{k-1}, y_{k-1})), (x_k, f(x_k, y_k))\}$.

Calculamos el polinomio interpolador que pasa por ellos:

$$p(t) = f(x_k, y_k)\frac{t - x_{k-1}}{h} + f(x_{k-1}, y_{k-1})\frac{x_k - t}{h}$$

sustituyendo en la integral:

$$y_{k+1} = y_k + \int_{x_k}^{x_{k+1}} \left(f(x_k, y_k)\frac{t - x_{k-1}}{h} + f(x_{k-1}, y_{k-1})\frac{x_k - t}{h} \right) dt$$

Haciendo el cambio de variable $t = x_k + hu \Rightarrow dt = hdu$, resulta

$$x_k - t = -hu$$
$$t - x_{k-1} = t - x_k + h = h(u+1)$$

Adams-Bashforth de dos pasos

Así que:

$$\int_{x_k}^{x_{k+1}} \left(f(x_{k-1}, y_{k-1}) \frac{x_k - t}{h} + f(x_k, y_k) \frac{t - x_{k-1}}{h} \right) dt =$$

$$= \int_0^1 \left(f(x_{k-1}, y_{k-1}) \left(-\frac{hu}{h} \right) + f(x_k, y_k) \frac{h(u+1)}{h} \right) h du =$$

$$= h \left(f(x_{k-1}, y_{k-1}) \int_0^1 -u du + f(x_k, y_k) \int_0^1 (u+1) du \right) =$$

$$= h \left(-\frac{1}{2} f(x_{k-1}, y_{k-1}) + \frac{3}{2} f(x_k, y_k) \right)$$

De modo que,

$$\boxed{y_{k+1} = y_k + \frac{h}{2} \left(3f(x_k, y_k) - f(x_{k-1}, y_{k-1}) \right)}$$

Es un método de segundo orden con un error global $O(h^2)$.

Adams-Bashforth

- Tres pasos AB3:

$$\boxed{y_{k+1} = y_k + \frac{h}{12}\left(23f(x_k, y_k) - 16f(x_{k-1}, y_{k-1}) + 5f(x_{k-2}, y_{k-2})\right)}$$

Conocido como el método de Adams-Bashforth de tercer orden cuyo error global es $O(h^3)$.

- Cuatro pasos AB4:

$$\boxed{y_{k+1} = y_k + \frac{h}{24}\left(55f(x_k, y_k) - 59f(x_{k-1}, y_{k-1}) + 37f(x_{k-2}, y_{k-2}) - 9f(x_{k-3}, y_{k-3})\right)}$$

Método de Adams-Bashforth de 4 pasos, con un error global $O(h^4)$.

<u>Para empezar</u>: partiendo de la condición inicial, necesitaremos calcular varios pasos con otro método para poder "lanzar" estos esquemas.

Algoritmo de AB4

- Entrada:
  - a, b, y_1, N
- Proceso:
  - Cálculo del paso de integración, h y obtención de x
  - Inicialización del vector solución y en a
  - Para k desde 1 hasta 3

$$k_1 = f(x_k, y_k),\ k_2 = f\left(x_k + \frac{h}{2}, y_k + \frac{h}{2}k_1\right),\ k_3 = f\left(x_k + \frac{h}{2}, y_k + \frac{h}{2}k_2\right)$$

$$k_4 = f(x_{k+1}, y_k + hk_3)$$

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

  - Fin para k
  - Para k desde 4 hasta N

$$y_{k+1} = y_k + \frac{h}{24}(55f(x_k, y_k) - 59f(x_{k-1}, y_{k-1}) + 37f(x_{k-2}, y_{k-2}) - 9f(x_{k-3}, y_{k-3}))$$

  - Fin para k
- Salida: x, y

Ejemplo AB4

- $y'(t) = (1 - 2t)\, y(t)$ N = 16

  $y(0) = 1 \;\rightarrow\; y_0 = 1$

Aplicamos RK4 para calcular y_1, y_2, y_3:

y_1 = 1.1646

y_2 = 1.2641

y_3 = 1.2790

Partiendo de estos puntos, aplicamos AB4:

y_4 = 1.2047

y_5 = 1.0598

$\ldots$

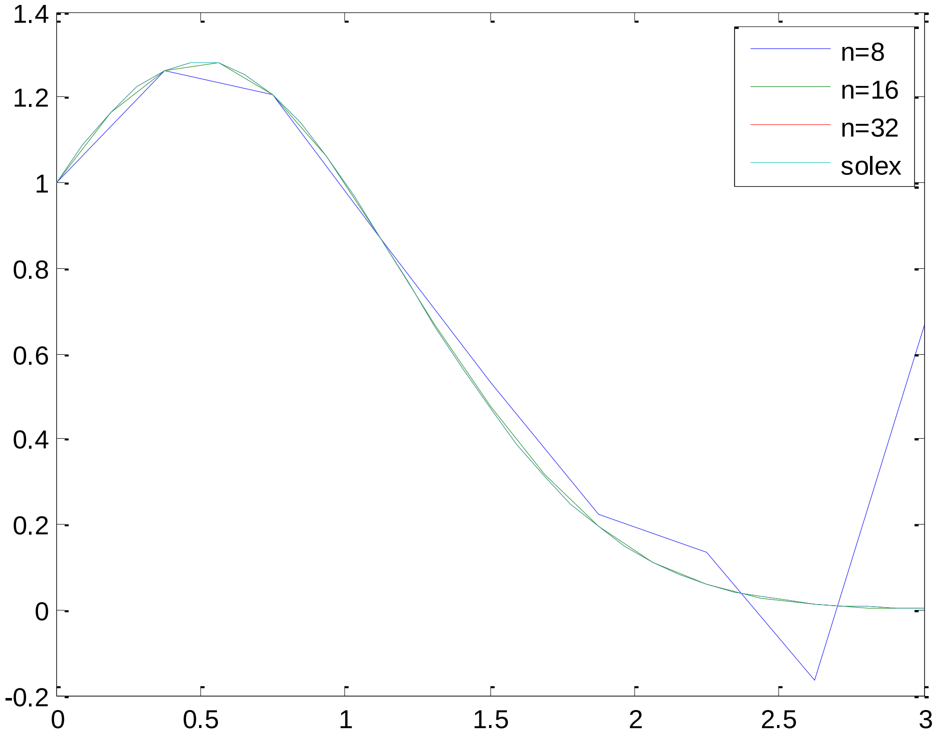
$y_{16} = 0.0055$

$y_{17} = 0.0023$

Ejemplo AB4

- $y'(t) = (1 - 2t)\, y(t)$
  $y(0) = 1$

n	Error máximo	$E_{\frac{n}{2}}/E_n$
8	0.6659	
16	0.0049	134.6022
32	2.8817e-04	17.1675
64	2.0374e-05	14.1441



Solución exacta (solex) $$y(x) = e^{\frac{1}{4} - \left(\frac{1}{2} - x\right)^2}$$

Métodos implícitos Adams-Moulton

- Método implícito de un paso (orden 2):

Utilizando los puntos $\{(x_k, f(x_k, y_k)), (x_{k+1}, f(x_{k+1}, y_{k+1}))\}$

$$p(t) = f(x_k, y_k)\frac{t - x_{k+1}}{x_k - x_{k+1}} + f(x_{k+1}, y_{k+1})\frac{t - x_k}{x_k - x_{k+1}} =$$

$$= f(x_k, y_k)\frac{x_{k+1} - t}{h} + f(x_{k+1}, y_{k+1})\frac{t - x_k}{h}$$

sustituyendo en:

$$y_{k+1} = y_k + \int_{x_k}^{x_{k+1}} p(t)dt$$

y operando, es fácil llegar a

$$\boxed{y_{k+1} = y_k + \frac{h}{2}\left(f(x_{k+1}, y_{k+1}) + f(x_k, y_k)\right)}$$

que es el método de Adams-Moulton de un paso.

Métodos implícitos Adams-Moulton

- Dos pases (orden 3)

$$y_{k+1} = y_k + \frac{h}{12}\left(-f(x_{k-1}, y_{k-1}) + 8f(x_k, y_k) + 5f(x_{k+1}, y_{k+1})\right)$$

- Tres pasos (orden 4)

$$y_{k+1} = y_k + \frac{h}{24}\left(f(x_{k-2}, y_{k-2}) - 5f(x_{k-1}, y_{k-1}) + 19f(x_k, y_k) + 9f(x_{k+1}, y_{k+1})\right)$$

- En general:
  - Un método AM de n pasos tiene orden n+1
  - Se necesita un método distinto para inicializar $y_1, y_2, \ldots$
  - Hay que resolver una ecuación no lineal para aplicar el método y calcular y_{k+1}.

```
function [x,y] = AM4( f,a,b,n,y0 )

h = (b-a)/n;
x = a:h:b;
y = zeros(n+1,1);
y(1) = y0;
for k = 1:2
    ff(k) = feval(f,x(k),y(k));
    k1 = ff(k);
    k2 = feval(f,x(k)+h/2,y(k)+h*k1/2);
    k3 = feval(f,x(k)+h/2,y(k)+h*k2/2);
    k4 = feval(f,x(k+1),y(k)+h*k3);
    y(k+1) = y(k)+h*(k1+2*k2+2*k3+k4)/6;
end

for k = 3:n    % estimacion por Newton de y(k+1)
ex = 1; iter = 0;   tol = 1e-6;   maxiter = 10;
y1 = y(k); %estimación inicial

fk = feval(f,x(k),y(k));
fkm1 = feval(f,x(k-1),y(k-1));
fkm2 = feval(f,x(k-2),y(k-2));

[ffun,dffun]=feval(f,x(k+1),y1);
```

```
fx=y1-y(k)-h/24*(19*fk-5*fkm1+fkm2+9*ffun);
efx = norm(fx);

while iter<maxiter &&  efx > tol && ex>tol

    dfx = 1-h/24*9*dffun;
    d = fx/dfx;
    t = y1 -d;

    [ffun,dffun] = feval(f,x(k+1),t);
    ft = t-y(k)-h/24*(19*fk-5*fkm1+fkm2+9*ffun);

    efx = norm(ft);     ex  = norm(t-y1);
    iter = iter+1;
    y1 = t;             fx = ft;

end

    y(k+1) = y1;
end
end
```

```
function [fun,dfun] = f(x,y)
%
fun=(1-2*x)*y;
dfun=(1-2*x);
end
```

Ejemplo AM4

- $y'(t) = (1-2t)\,y(t)$ N = 16

  $y(0) = 1 \rightarrow \; y_0 = 1$

Aplicamos RK4 para calcular y_1, y_2, y_3:

y_1 = 1.1646

y_2 = 1.2641

y_3 = 1.2790

Partiendo de estos puntos, aplicamos Newton (tolerancia 10^{-6}, maxiter 10) para encontrar cada nuevo valor con AM4:

$$\mathrm{g}(y_4) = y_4 - y_3 - \frac{h}{24}\big(f(x_1, y_1) - 5f(x_2, y_2) + 19f(x_3, y_3) + 9f(x_4, y_4)\big) = 0$$

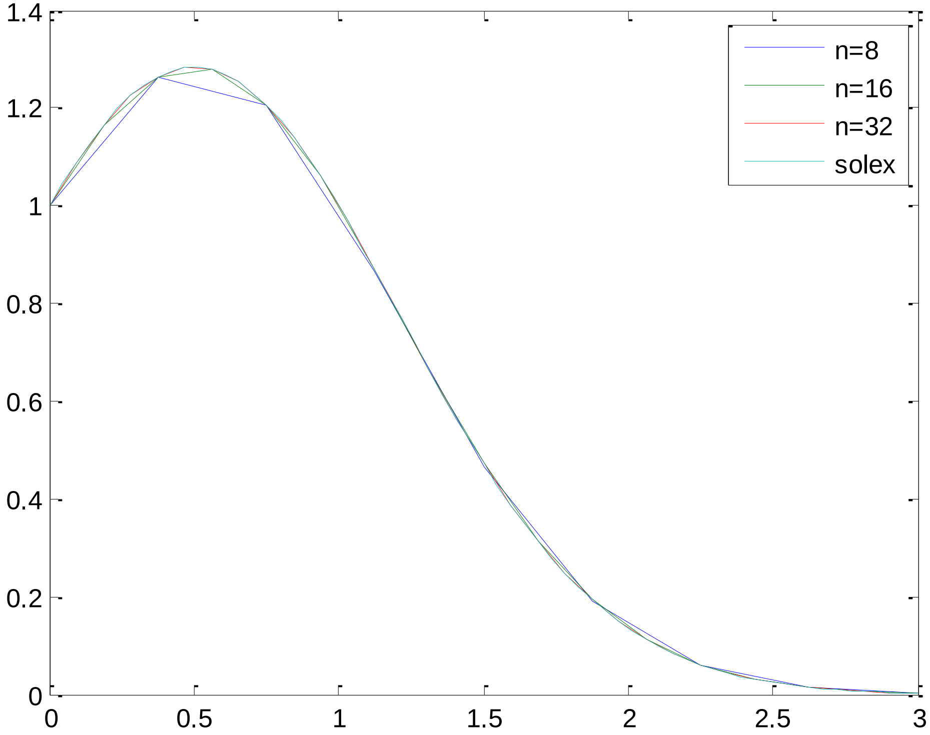
utilizando $z_0 = y_3$ como estimación inicial

$$z_i = z_{i-1} - \frac{g(z_{i-1})}{g'(z_{i-1})}, i = 1,2, \ldots \rightarrow y_4$$

Ejemplo AM4

- $y'(t) = (1-2t)\,y(t)$
  $y(0) = 1$

n	Error máximo	$E_{\frac{n}{2}}/E_n$
8	0.0067	
16	3.8377e-04	17.5203
32	2.0957e-05	18.3122
64	1.6371e-06	12.8016



Solución exacta (solex) $y(x) = e^{\frac{1}{4}-\left(\frac{1}{2}-x\right)^2}$

Métodos predictor-corrector

- En estos métodos la aproximación y_{k+1} es calculada por un método explícito cómo los estudiados con anterioridad y luego utilizar un método implícito para mejorarlo.

Por ejemplo, utilizando los métodos de Adams-Bashforth de dos pasos y Adams-Moulton de 2 pasos, podemos obtener un método predictor-corrector como sigue:

- Predictor: $y_{k+1}^{(p)} = y_k + \frac{h}{2}\left(3\, f(t_k, y_k) - \; f(t_{k-1}, y_{k-1})\right)$
- Corrector: $y_{k+1} = y_k + \frac{h}{2}\left(f\left(t_{k+1}, y_{k+1}^{(p)}\right) + f(t_k, y_k)\right)$

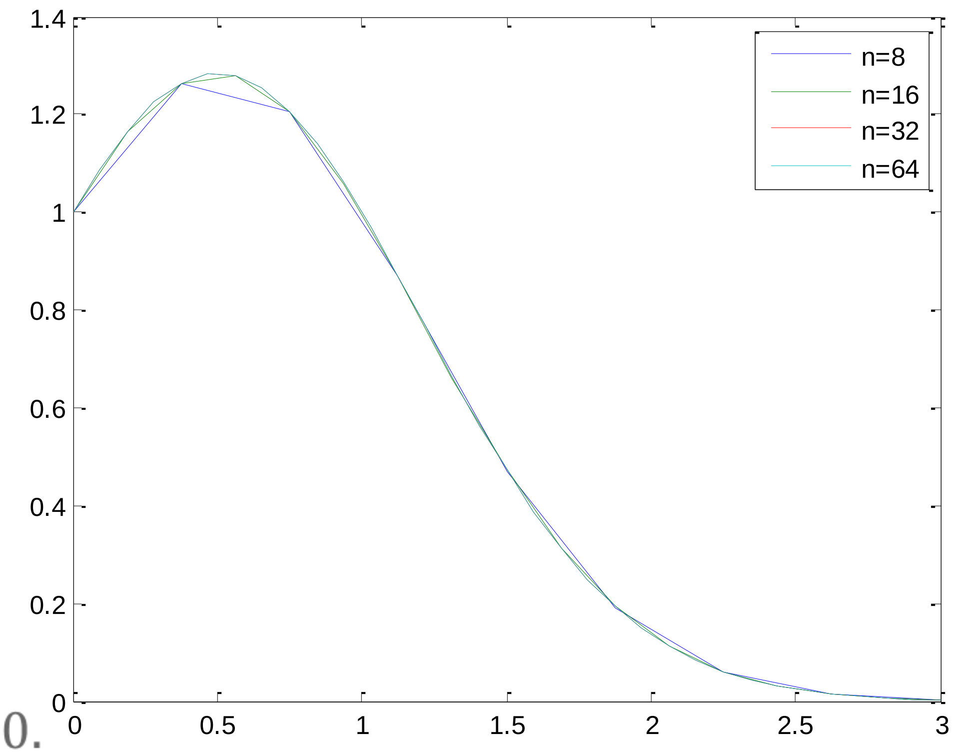
Explícito

Implícito

Ejemplo ABM4

- $y'(t) = (1-2t)\,y(t)$
  $y(0) = 1$

n	Error máximo	$E_{\frac{n}{2}}/E_n$
8	0.0830	
16	0.0012	67.7615
32	4.5191e-05	27.1175
64	1.9147e-06	23.6026

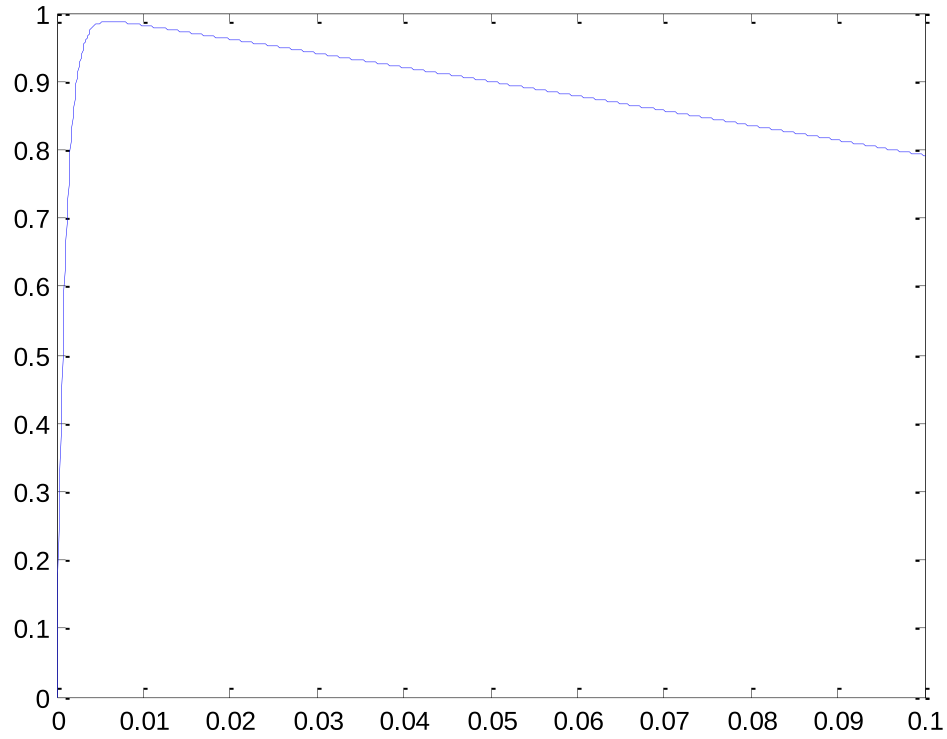


Newton: $\mathrm{tol} = 10^{-6}, maxiter = 10.$

Solución exacta (solex) $\quad y(x) = e^{\frac{1}{4}-\left(\frac{1}{2}-x\right)^2}$

Motivación: Ecuaciones rígidas (stiff)

- $y'(x) = -1000y(x) + 3000 - 2000e^{x}$
  $y(0) = 0,$
  $x \in [0,0.1]$



Sol. exacta: $y(x) = 3 - 1.9980e^{x} - 1.002e^{-1000x}$

Ecuaciones rígidas

- $y'(t) = -1000y(t) + 3000 - 2000e^t$
  $y(0) = 0, t \in [0,0.1]$

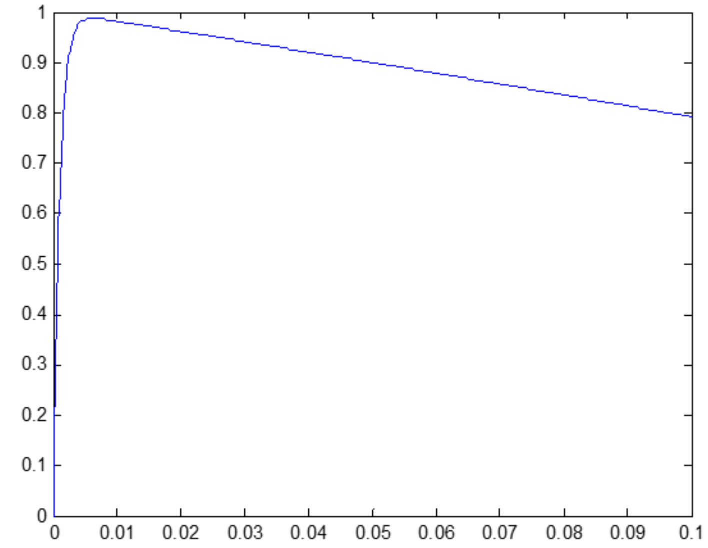
Método	n	Error máx
Euler	10	3.49e+09
	100	0.3686
Euler impl.	10	0.0911
	100	0.1324
Heun	10	1.35e+16
	100	0.1324
RK4	10	4.36e+24
	100	0.0071

Método	n	Error máx
AB4	10	8.02e+16
	100	1.40e+37
AM4	10	5.98e+06
	100	0.0071
ABM4	10	3.06e+20
	100	0.0240
ode45	10	1.89e-04
	20	8.95e-04

Sol. exacta: $y(t) = 3 - 1.9980e^t - 1.002e^{-1000t}$

Estabilidad del método iterativo

- $y'(t) = -1000y(t) + 3000 - 2000e^t$
  $y(0) = 0, t \in [0,0.1]$



- La parte transitoria de la solución está dominada por la exponencial rápida e^{-1000t}, pero a partir de $t > 0.0062114$ esta parte termina y la ecuación se rige por la exponencial lenta.

- Analizando la parte homogénea de la EDO, se puede determinar el tamaño de paso necesario para que la solución sea estable.

Estabilidad Euler explícito

- Consideremos la ecuación homogénea, con condición inicial:

$$y'(t) = -\lambda y(t), \qquad y(0) = y_0, \quad t \in [0,0.1]$$

cuya solución exacta es $y = y_0 e^{-\lambda t}$.

Si aplicamos el método de Euler,

$$y_{k+1} = y_k - h\lambda y_k = y_k(1 - h\lambda),$$

y mediante un proceso de inducción,

$$y_{k+1} = y_0(1 - h\lambda)^{\mathrm{k+1}} \, .$$

Deducimos que para que la solución numérica esté acotada, debe verificarse

$$|1 - \lambda h| < 1$$

- En nuestro problema: $\lambda = 1000$ y por tanto, $h < \frac{1}{1000}$

```
>> [x,y] = euler('f',0,0.1,10000,0);
>> solex = 3-1.9980*exp(x)-1.002*exp(-1000*x);
>> error = max(abs(y-solex))
                    error =     0.0018
```

Estabilidad Euler implícito

- Consideremos la ecuación homogénea, con condición inicial:

$$y'(t) = -\lambda y(t), \qquad y(0) = y_0, \quad t \in [0,0.1]$$

cuya solución exacta es $y = y_0 e^{-\lambda t}$.

Si aplicamos el método de Euler hacia atrás,

$$y_{k+1} = y_k - h\lambda y_{k+1},$$

$$y_{k+1}(1 + h\lambda) = \mathrm{y_k}$$

$$y_{k+1} = y_0 \left(\frac{1}{1+h\lambda}\right)^{\mathrm{k+1}} .$$

Para que la solución numérica esté acotada, debe verificarse

$$\left|\frac{1}{1+h\lambda}\right| < 1 \quad \Longrightarrow \quad \text{Incondicionalmente estable}$$

Resolución de problemas rígidos con Matlab

Matlab ODE solvers	
ode23	non-stiff, low order
ode113	non-stiff, variable order
ode15s	stiff, variable order, includes DAE
ode23s	stiff, low order
ode23t	trapezoid rule
ode23tb	stiff, low order
ode45	non-stiff, medium order (Runge-Kutta)

- Todos los algoritmos usan paso adaptativo
- Puede aproximar la solución en los nodos solicitados
- Mejor opción para problemas rígidos: ode15s

Para profundizar…

- J.H. Mathews, K. D. Fink, Métodos numéricos con Matlab, Ed. Prectice Hall, Madrid, 2000.

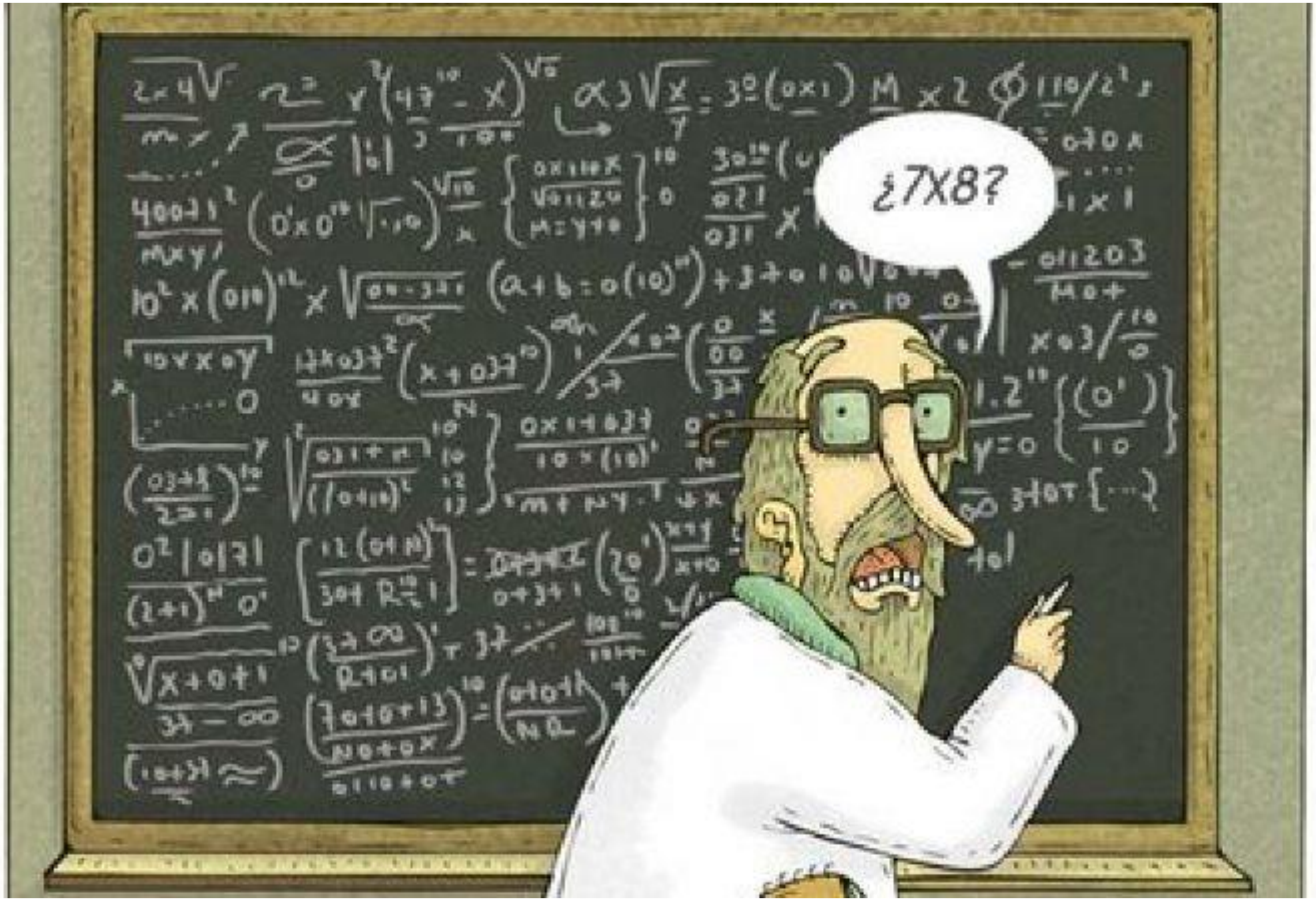
- R.L. Burden, J.D. Faires, Análisis numérico, Ed. Thomson Learning, México DF, 2002.

- D. F. Griffiths, D. J. Higham. *Numerical Methods for Ordinary Differential Equations: Initial Value Problems*, 1st Edition. Springer, 2010.

- http://www.math.pitt.edu/~sussmanm/2071Spring09/lab03/index.html#StiffSystems

- A. Cordero, J.L. Hueso, E. Martínez, J.R. Torregrosa, Problemas resueltos de métodos numéricos, Ed. Thomson, 2006.

¿Dudas?



unir

LA UNIVERSIDAD EN INTERNET

www.unir.net