

MATEMÁTICA SUPERIOR APLICADA

Solución Numérica de Ecuaciones Diferenciales Ordinarias en Ingeniería Química

Universidad Tecnológica Nacional – Facultad Regional Rosario

Dr. Alejandro S. M. Santa Cruz



Algunas Definiciones (I)

- **Ecuación diferencial:** Ecuación que relaciona dos o más variables en términos de sus derivadas o diferenciales.
- **Variables independientes:** Si existen una o más derivadas con respecto a esa variable.
- **Variables dependientes:** Cuando existen derivadas con respecto a esa variable.
- **Ecuaciones diferenciales ordinarias (EDO's):** Si hay una sola variable independiente (entonces las derivadas son totales).
- **Ecuaciones diferenciales en derivadas parciales (EDP's):** Si en la ecuación aparecen dos o más variables independientes (entonces las derivadas serán parciales).



Algunas Definiciones (II)

- **Orden de una ecuación diferencial:** Determinado por la derivada de mayor orden que aparece en la ecuación diferencial.
- **Grado de una ecuación diferencial:** Determinado por el grado algebraico de la derivada de mayor orden que aparece en la ecuación.
- **Ecuación diferencial lineal:** Si no aparecen potencias de la variable dependiente y sus derivadas ni productos de la variable dependiente por sus derivadas o productos entre derivadas.
- **Solución de una ecuación diferencial:** Cualquier relación funcional de las variables que no incluya derivadas o integrales de funciones desconocidas, en el sentido que la verifique idénticamente por sustitución directa.
- **Ecuación diferencial homogénea:** Una ecuación diferencial (o condición) se dice homogénea si es satisfecha por una función particular $x(t)$, también lo es por $c x(t)$.



Ejemplos

$$\frac{dx}{dt} = \cos \omega t \quad (a)$$

$$\frac{d^2 x}{dt^2} + k^2 x = 0 \quad (b)$$

$$(t^2 + x^2) dt + 2 t x dx = 0 \quad (c)$$

$$\left(\frac{d^2 x}{dt^2} \right)^3 - t x \frac{dx}{dt} + x = 0 \quad (d)$$

$$\frac{d^2 x}{dt^2} - \left(\frac{dx}{dt} \right)^3 - 8 x = 0 \quad (e)$$

$$\frac{\partial^2 V}{\partial x^2} + \frac{\partial^2 V}{\partial y^2} = 0 \quad (f)$$



Solución de una Ecuación Diferencial (I)

Consideremos la forma implícita de la EDO de orden n :

$$F(t, x, x', x'', \dots, x^{(n)}) = 0$$

donde:

$$x'(t) = \frac{dx}{dt}$$

$$x''(t) = \frac{d^2x}{dt^2}$$

⋮

$$x^{(n)}(t) = \frac{d^n x}{dt^n}$$



Solución de una Ecuación Diferencial (II)

- Así expresada, la EDO es:
 - 1) Ordinaria (sólo aparecen derivadas totales).
 - 2) De orden n (n es el mayor orden de derivación).
 - 3) En principio, puede ser lineal o no lineal.
- Si existe una función $x(t)$ que satisface la EDO de orden n , entonces $x(t)$ se llama solución de la ecuación diferencial.



Solución de una Ecuación Diferencial (III)

Para encontrar la primitiva $x(t)$ es necesario efectuar n integraciones, lo cual implica que deben aparecer n constantes arbitrarias. Entonces puede aceptarse que:

$$G(x, t, c_1, c_2, c_3, \dots, c_n) = 0$$

o

$$x = g(t, c_1, c_2, c_3, \dots, c_n)$$

es la solución de la ecuación diferencial.



Solución de una Ecuación Diferencial (IV)

Dependiendo como se elijan estos valores o constantes para particularizar una solución, distinguimos dos tipos de problemas:

- 1) Problema de valores iniciales.
- 2) Problema de valores de contorno.



Problema de Valores Iniciales

Gobernado por una EDO de orden n : $F(t, x, x', x'', \dots, x^{(n)}) = 0$
(forma implícita) y un conjunto de n condiciones independientes
válidas para el punto inicial $t = t_0$:

$$x(t_0) = x_0$$

$$x'(t_0) = x'_0$$

$$x''(t_0) = x''_0$$

\vdots

$$x^{(n-1)}(t_0) = x_0^{(n-1)}$$



Problema de Valores de Contorno

- Deben establecerse condiciones en todos y cada uno de los puntos que constituyen la frontera del dominio de soluciones del problema.
- En el espacio o dominio unidimensional hay dos puntos frontera, en $t = t_0$ y $t = t_f$ si el dominio es el intervalo cerrado $[t_0, t_f]$.
- Por consiguiente el orden mínimo de una EDO para un problema de valores de frontera es **2**.



Aproximación a la Solución de EDO's

- **Procesos fisicoquímicos en IQ \Rightarrow cambios dependientes del tiempo.**
- **Modelización \rightarrow EDO's**
- **Técnicas de resolución:**
 - 1) Analíticas (bien conocidas).**
 - 2) Numéricas (fundamentales para resolver ecuaciones diferenciales no lineales y stiff).**



Solución de una EDO de Orden n

- Se lleva a un *sistema de n ecuaciones diferenciales de 1er. Orden* mediante un cambio de variables:

$$x_1 \equiv x(t); x_2 \equiv \frac{dx}{dt}; x_3 \equiv \frac{d^2 x}{dt^2}; \dots; x_n \equiv \frac{d^{(n-1)} x}{dt^{(n-1)}}$$

y

$$x'_1(t) = x_2$$

$$x'_2(t) = x_3$$

\vdots

$$x'_{n-1}(t) = x_n$$

$$x'_n = f(t, x_1, x_2, \dots, x_n)$$



Aproximación a la Solución de EDO's de 1er. Orden

Forma Implícita: $F(t, x, x') = 0$

Forma Explícita: $\frac{dx}{dt} = f(t, x)$

Si $f = f(t)$

$$x(t) = x_0 + \int_{t_0}^t f(t) dt \quad \Rightarrow \quad \text{Métodos de Cuadratura}$$

Si $f = f(x, t)$

$$x(t) = x_0 + \int_{t_0}^t f[x(t), t] dt \quad \Rightarrow \quad \text{Métodos de Integración}$$



Aproximación a la Solución de EDO's de 1er. Orden

- Como primer etapa en el proceso de cálculo en forma numérica lo que se hace es dividir el intervalo de variación de la variable independiente t , $[t_0, t_f]$ en subintervalos o pasos en los que se aproxima la solución verdadera $x(t)$ en $(n+1)$ valores igualmente espaciados de t : $t_0, t_1, t_2, \dots, t_f = t_n$ tal que:

$$h = (t_f - t_0)/n$$

y $t_i = t_0 + ih$, con $i = 0, 1, 2, 3, \dots, n$

- Solución de la EDO: En forma tabular en los $(n+1)$ puntos de división del intervalo.
 - $x(t_i)$: Valor verdadero de la función en t_i
 - x_i : Valor aproximado de la función en t_i
- Entonces: $x'(t_i) \approx f(t_i, x_i) = f_i$
- Si no hay errores de redondeo (o al menos éstos son suficientemente pequeños como para que no influyan en el cálculo), entonces: $|e_i| = |x_i - x(t_i)|$ se denomina *error de discretización* o de *truncamiento*.



Algoritmos Numéricos para Resolver EDOs con Condiciones Iniciales

En una primera aproximación podemos clasificarlos en:

- I. Aquellos que hacen uso directo o indirecto de la expansión en serie de Taylor de la solución $x(t)$.
- II. Fórmulas de integración abiertas o cerradas.

A su vez, los diversos procedimientos pueden clasificarse de manera general como:

- 1) Métodos de paso simple: Se evalúa x_{i+1} con t_i e x_i y/o f_i
- 2) Métodos de paso múltiple: Se evalúa x_{i+1} con valores fuera de $[t_i, t_{i+1}]$.



Aproximación de una EDO Mediante Expansión en Series de Taylor (I)

- Se expande $x(t)$ alrededor de la condición inicial $x(t_0)$:

$$x(t_0 + h) = x(t_0) + h x'(t_0) + \frac{h^2}{2!} x''(t_0) + \frac{h^3}{3!} x'''(t_0) + \dots +$$

- Si se especifica la c.i., $x(t_0)$,

$$x'(t_0) = \left. \frac{dx}{dt} \right|_{t=t_0} = f[t_0, x(t_0)]$$

- Entonces:

$$x(t_0 + h) = x(t_0) + h f[t_0, x(t_0)] + \frac{h^2}{2!} f'[t_0, x(t_0)] + \frac{h^3}{3!} f''[t_0, x(t_0)] + \dots +$$

donde

$$f'[t_0, x(t_0)] = \left. \frac{d}{dt} f[t, x(t)] \right|_{t=t_0}$$

$$f''[t_0, x(t_0)] = \left. \frac{d^2}{dt^2} f[t, x(t)] \right|_{t=t_0}$$

y así sucesivamente.



Aproximación de una EDO Mediante Expansión en Series de Taylor (II)

- Las derivadas de orden superior se obtienen aplicando la regla de la cadena:

$$\frac{d}{dt} f(t, x) = \frac{\partial f}{\partial t} + \frac{\partial f}{\partial x} \frac{dx}{dt}$$

- Para pasar de t_i a $t_{i+1} = t_i + h$ hacemos

$$x(t_i + h) = x(t_i) + h f[t_i, x(t_i)] + \dots + \frac{h^n}{n!} f^{(n-1)}[t_i, x(t_i)] + \frac{h^{n+1}}{(n+1)!} f^{(n)}[\xi_i, x(\xi_i)]$$

con $\xi_i \in (t_i, t_{i+1})$



Aproximación de una EDO Mediante Expansión en Series de Taylor (III)

Algoritmos de Orden h^n

- Lado izquierdo del desarrollo de Taylor:

$$x(t_{i+1}) = x(t_i + h) \approx x_{i+1}$$

- Lado derecho:

$$x(t_i) \approx x_i \quad f^{(n)}[t_i, x(t_i)] \approx f^{(n)}(t_i, x_i) = f_i^{(n)}$$

- Algoritmo:

$$x_{i+1} = x_i + h f_i + \frac{h^2}{2!} f_i' + \frac{h^3}{3!} f_i'' + \dots + \frac{h^n}{n!} f_i^{(n-1)} + \frac{h^{n+1}}{(n+1)!} f^{(n)}(\xi_i)$$



Aproximación de una EDO Mediante Expansión en Series de Taylor (IV)

Algoritmos de Orden h^n

- Error local de truncamiento:

$$|e| \leq \frac{h^{n+1}}{(n+1)!} E \quad E \geq \left| f^{(n)} [\xi, x(\xi)] \right|_{\text{máx}} ; \xi \in (t_i, t_{i+1})$$

- La derivación de $f[t, x(t)]$ puede llegar a ser muy complicada.
- Sólo se utiliza en para el caso más simple:

$$x_{i+1} = x_i + h f(t_i, x_i)$$

- En general, la expansión en serie de Taylor de $x(t)$ no se utiliza para resolver EDOs.



Métodos Explícitos de Resolución de EDOs

- Evaluación explícita de $f(t,x)$ y sus derivadas.
- Avance paso a paso en el tiempo sin utilizar procedimientos iterativos.
- Métodos más difundidos:
 - ✓ Euler
 - ✓ Runge – Kutta de 4to. orden

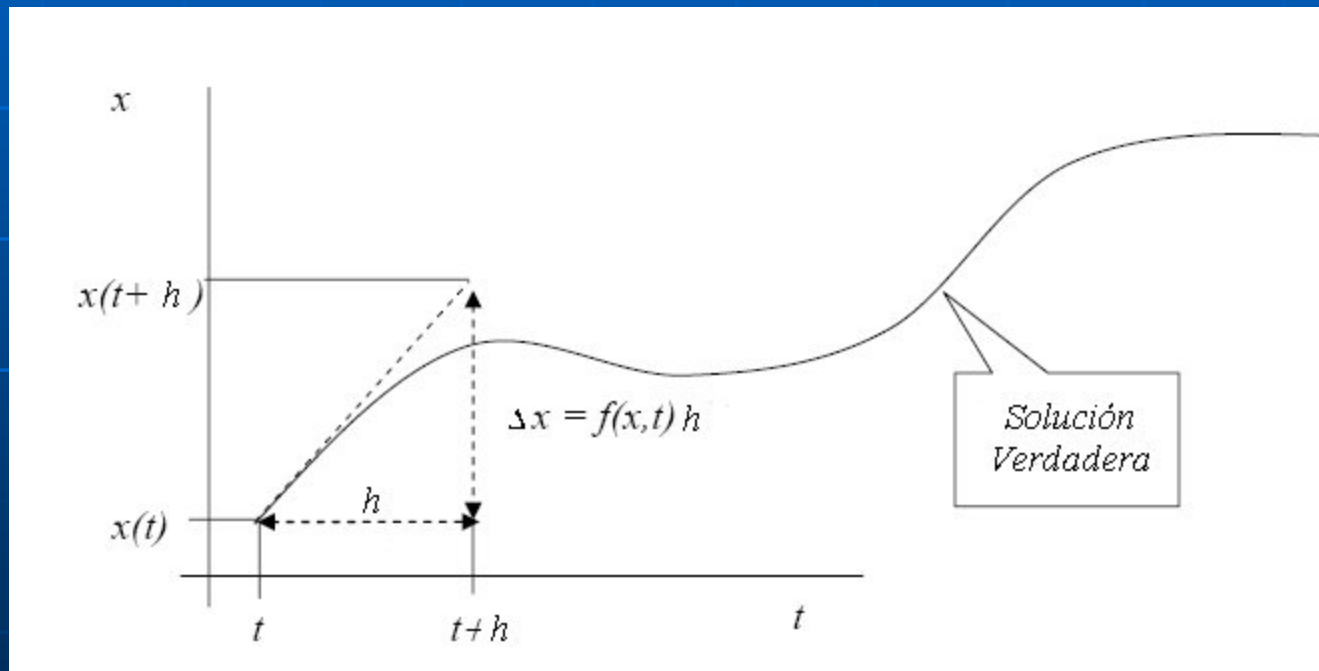


Método Explícito de Euler (I)

(Forward Euler)

Algoritmo: $x_{i+1} = x_i + h f(t_i, x_i)$

h es el paso de integración



Método Explícito de Euler (II)

(Forward Euler)

- Error local de truncamiento:

$$|e| \leq \frac{h^2}{2} E \text{ con } E \geq \left| f' [\xi, x(\xi)] \right|_{\max} ; \xi \in (t_i, t_{i+1})$$

- Error global (el acumulado en todos los pasos en el intervalo de integración):

$$|e|_{\text{global}} \leq \frac{M h}{2K} e^{(t_f - t_0)K}$$

donde M es una cota superior de $f'(x,t)$ en $[t_0, t_f]$ y K satisface:

$$\left| f(t, x^*) - f(t, x) \right| = \left| \frac{\partial f(t, \alpha)}{\partial x} \right| |x^* - x| \leq K |x^* - x| ; x < \alpha < x^*$$



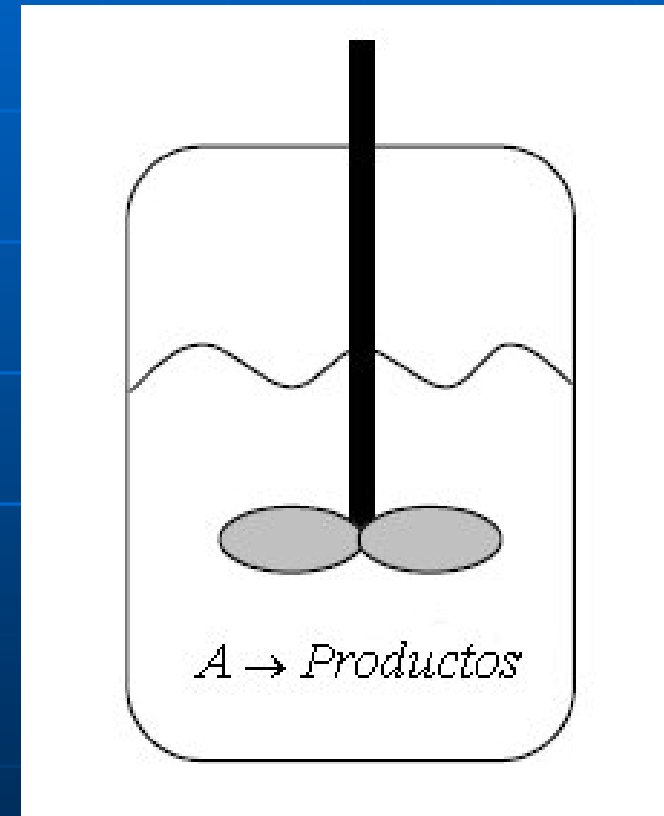
Ejemplo 01

Método Explícito de Euler

Reacción de 1er. Orden en un Reactor Batch

- El reactivo es consumido por una reacción de 1er. orden.
- La velocidad de reacción por unidad de volumen de fluido en el reactor es $r = -k c$. donde k es constante, y c es la concentración del reactivo en el reactor.
- El balance de materia en el reactor suministra la siguiente ecuación:

$$V \frac{dc}{dt} = -kcV$$



Ejemplo 01

Método Explícito de Euler

- La concentración del reactivo está determinada por:

$$V \frac{d(c/c_0)}{dt} = -k(c/c_0)V$$

donde c es la concentración al tiempo t , y c_0 es la concentración inicial.

- Si establecemos que $x=c/c_0$, entonces:

$$\frac{dx}{dt} = -kx$$

y $x(t_0=0)=1$.

- Una solución analítica generará un decaimiento exponencial con un tiempo constante, $\tau = 1/k$.
- El problema es de valores iniciales porque hemos suministrado una condición inicial y podemos avanzar en el tiempo a partir de esa condición inicial.



Ejemplo 01

Método Explícito de Euler

Archivo de Comando de MATLAB:

Reactor.m: Archivo de comandos que llama a la función **Euler.m**.

Método:

Euler.m: Función que implementa el algoritmo explícito de Euler.

Funciones:

TestFunction.m: Función que implementa el lado derecho de la EDO explícita (llamada por **Euler.m**).



Ejemplo 01

Método Explícito de Euler

Ejecución del Programa en MATLAB

>> Reactor

Ingrese el valor inicial de $x(t)$: 1

Ingrese el tiempo inicial t_0 : 0

Ingrese el tiempo final t_f : 10

Ingrese el número de pasos: 100

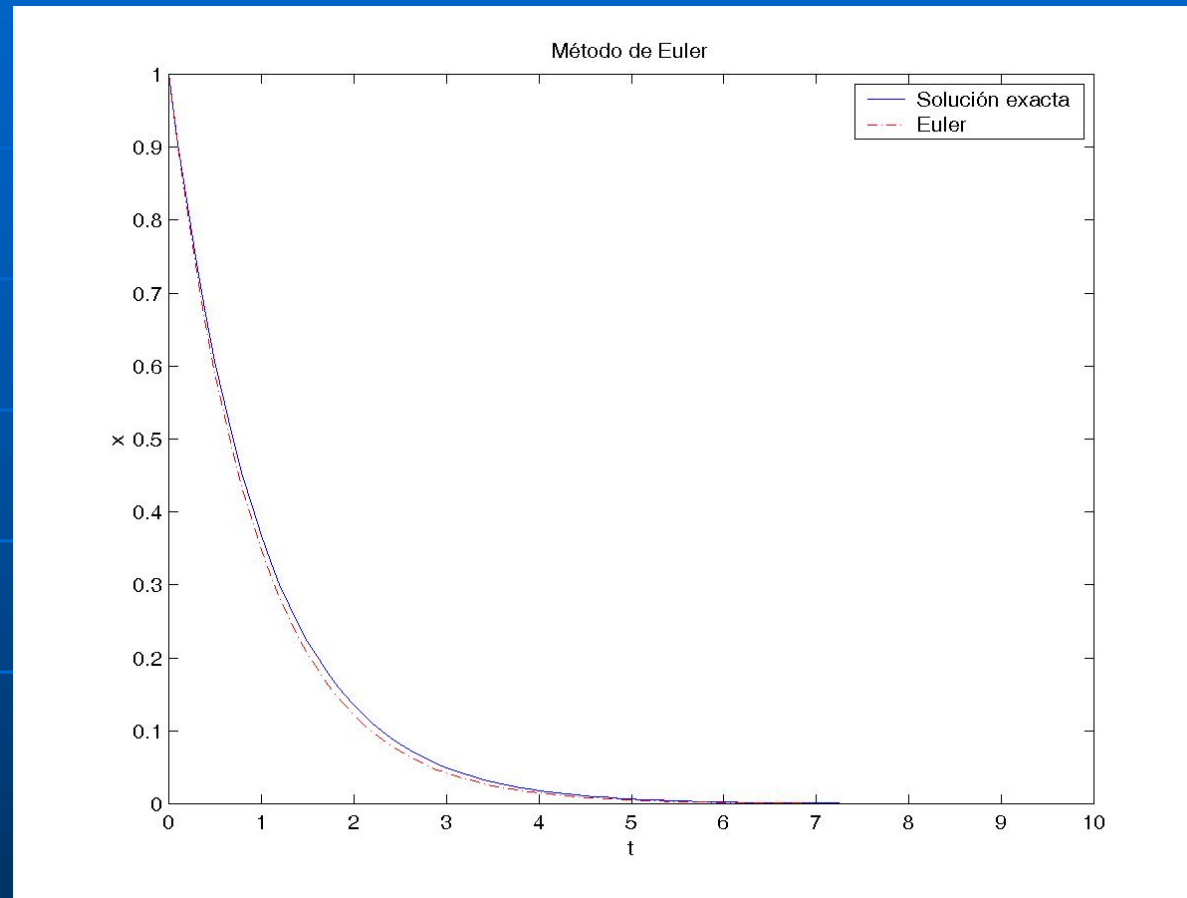
Ingrese la función de prueba: 'TestFunction'

La solución numérica de nuestro problema se grafica a continuación utilizando diferentes números de pasos en el rango de 0 a 10 segundos.



Ejemplo 01

Método Explícito de Euler

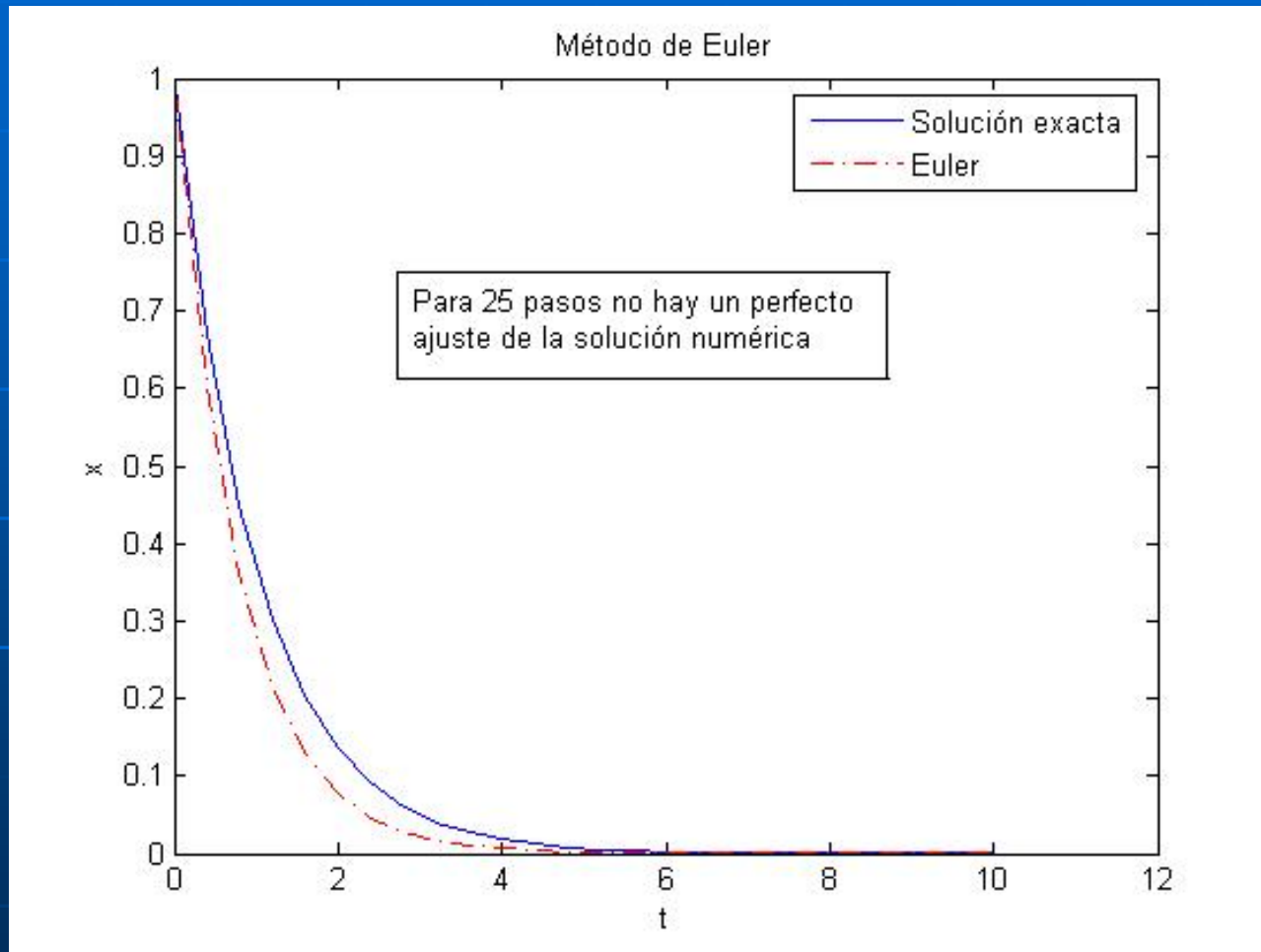


Solución del problema utilizando 100 pasos sobre 10 seg.



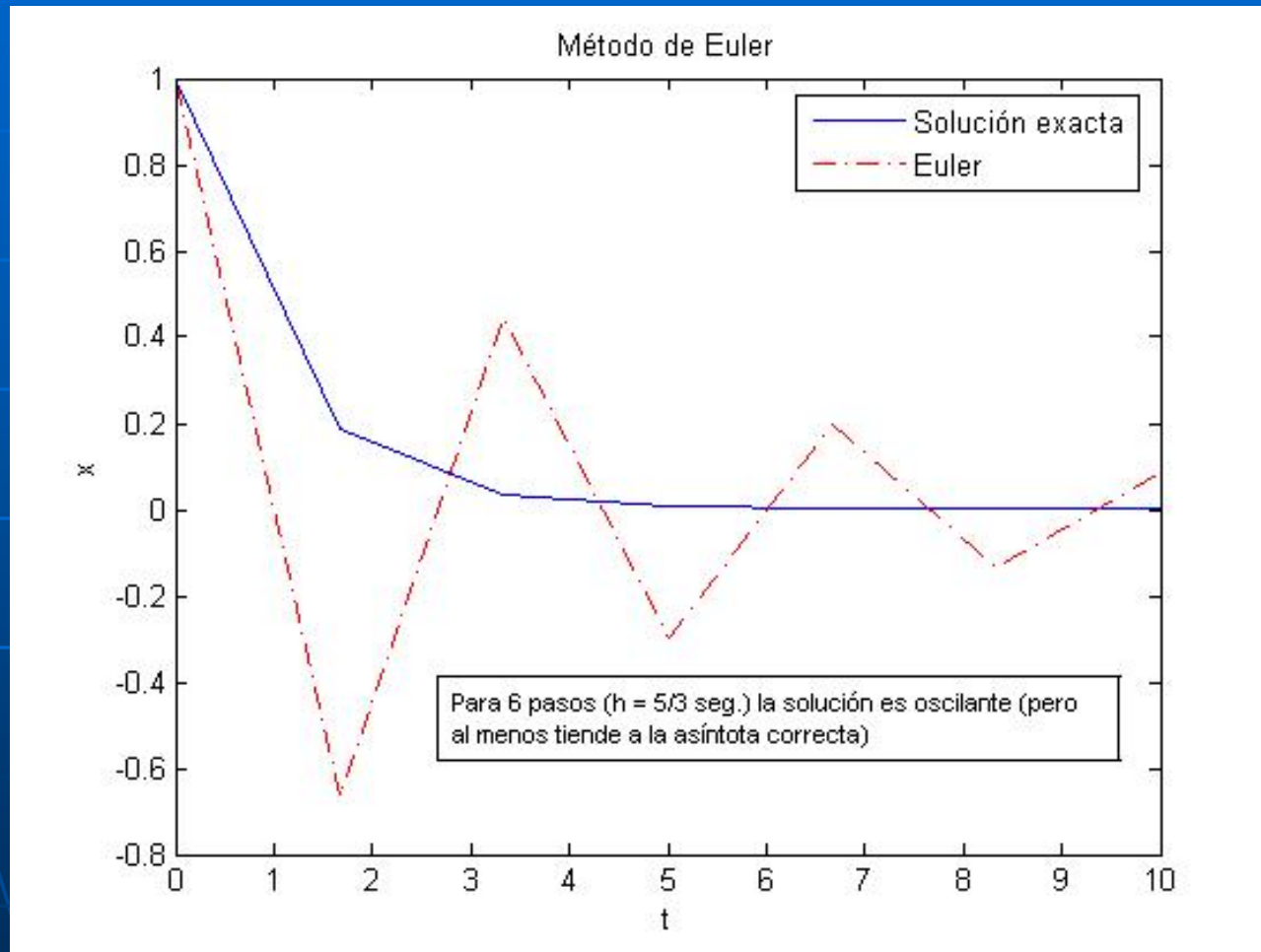
Ejemplo 01

Método Explícito de Euler



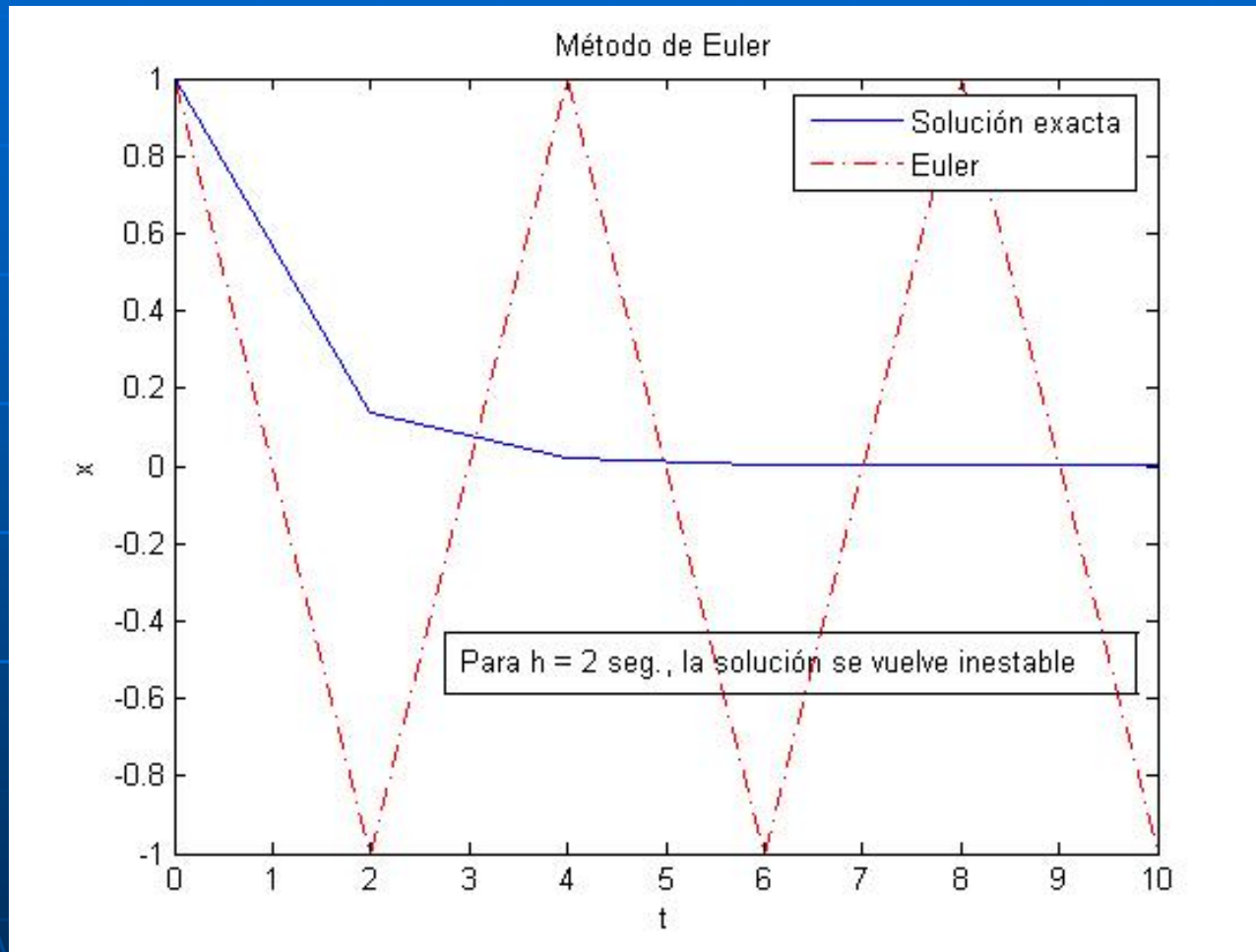
Ejemplo 01

Método Explícito de Euler



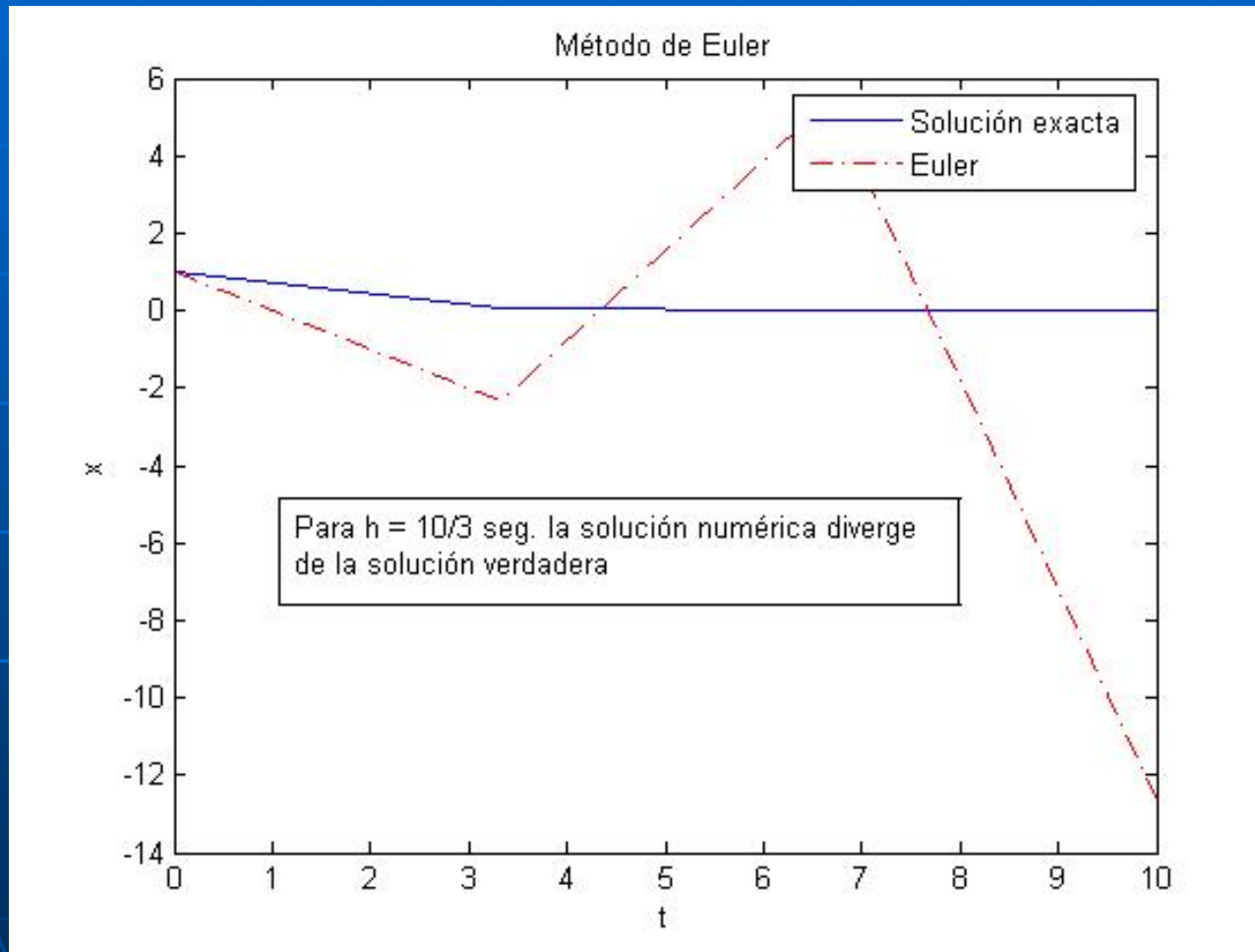
Ejemplo 01

Método Explícito de Euler



Ejemplo 01

Método Explícito de Euler



Estabilidad del Método de Euler (I)

- La solución exacta de la ecuación:

$$\frac{dx}{dt} = \lambda x \quad \text{para } x(t=0) = 1$$

es:

$$x^{exacta} = e^{\lambda t}$$

- Para $\lambda < 0$, la solución debería decaer a cero. Deseamos que nuestra solución numérica haga lo mismo:

$$x_{i+1} = x_i + h f_i = x_i + \lambda h x_i = x_i (1 + \lambda h)$$

- Para que la solución numérica tienda a cero, entonces se debe verificar que:

$$|1 + \lambda h| < 1 \Rightarrow -2 < \lambda h < 0$$



Estabilidad del Método de Euler (II)

Conclusión: En nuestro ejemplo, el método se vuelve inestable si:

$$h > 2 \text{ seg. } (\lambda = -1)$$



Estabilidad del Método de Euler (III)

	$\lambda < 0$	$\lambda > 0$
<p>Error Absoluto</p> $e_{i+1} \approx e_i (1 + \lambda h) - \frac{h^2}{2} \lambda^2 x(t_i)$	<p>El error absoluto decrecerá si :</p> $ (1 + \lambda h) < 1$	<p>El error absoluto se incrementa sin límites.</p>
<p>Error Relativo</p> $e_{i+1}^r \approx e_i^r \left[(1 + \lambda h) - \frac{h^2}{2} \lambda^2 \right] e^{-\lambda h}$	<p>El error relativo se vuelve muy grande. Esto se debe a que la solución tiende a cero. El error relativo no es relevante.</p>	<p>El error relativo decrece en cada iteración. Esto se debe a que la solución crece exponencialmente.</p>



Estabilidad del Método de Euler (IV)

Extensión del Análisis de Errores y de Estabilidad a Ecuaciones No Lineales

- Generalmente una EDO tendrá la forma: $\frac{dx}{dt} = f(x)$
- Podemos aproximar el lado derecho utilizando el desarrollo en serie de Taylor:

$$\frac{d(x - x_0)}{dt} = f(x_0) + \left(\frac{df}{dx}\right)_{x_0} (x - x_0)$$

- Un cambio de variables a $Y = x - x_0$ dará:

$$\frac{dY}{dt} = \left(\frac{df}{dx}\right)_{x_0} Y + cte$$

- Localmente, ahora la ecuación luce como nuestro problema previo, pero con:

$$\lambda = \left(\frac{df}{dx}\right)_{x_0}$$



Estabilidad del Método de Euler (V)

Concluimos entonces que:

- 1) El Método Explícito de Euler no es muy exacto (*low order method*).
- 2) Es condicionalmente estable (la solución numérica divergerá si el paso de integración es muy pequeño).
- 3) El intervalo de estabilidad viene dado por: $-2 < h \lambda < 0$.
- 4) Otros métodos explícitos como el de Euler también requerirán de pasos de integración pequeños.
- 5) Para ecuaciones diferenciales *stiff* ($\lambda \ll -1$) el paso de integración no tendría que ser demasiado grande.



Técnicas Explícitas de Orden Superior

Métodos de Runge-Kutta (I)

- Como vimos previamente, la solución de una EDO mediante la expansión directa en serie de Taylor de la función objeto no es práctica.
- Las derivadas de orden superior al primero que se necesitan conservar son, en general, bastante complicadas. Por consiguiente, cuando se desee trabajar con errores de orden superior no se pueden desarrollar algoritmos de cálculo análogos al de Euler directamente de la expansión en serie de Taylor.
- Sin embargo, es posible desarrollar métodos de paso simple que involucren evaluaciones de derivadas de 1er. Orden, pero que produzcan resultados equivalentes en exactitud a las fórmulas de Taylor de orden superior.



Técnicas Explícitas de Orden Superior

Métodos de Runge-Kutta (II)

- Todos los métodos de Runge-Kutta tienen algoritmos de la forma:

$$x_{i+1} = x_i + h\Phi(t_i, x_i, h)$$

donde Φ , denominada *función incremento*, representa una aproximación adecuada de $f(t, x)$ en el intervalo $[t_i, t_{i+1}]$.

- La estructura del algoritmo nos dice que son métodos explícitos.



Técnicas Explícitas de Orden Superior

Método R-K de 2do. Orden (I)

Algoritmo:

$$x_{i+1} = x_i + h\Phi$$

$$\Phi = (ak_1 + bk_2)$$

$$k_1 = f(t_i, x_i)$$

$$k_2 = f(t_i + ph, x_i + qhk_1)$$

donde a , b , p y q deben elegirse de manera que generen una solución cuya exactitud sea equivalente a la serie truncada de Taylor que retiene términos hasta el orden h^2 .



Técnicas Explícitas de Orden Superior

Método R-K de 2do. Orden (II)

Identificamos dos casos:

1) Caso $b=1/2$. En este caso, $a=1/2$, $p=1$ y $q=1$. Luego:

$$x_{i+1} = x_i + \frac{h}{2}(k_1 + k_2)$$

$$k_1 = f(t_i, x_i)$$

$$k_2 = f(t_i + h, x_i + hk_1)$$

2) Caso $b=1$. En este, $a=0$, $p=1/2$ y $q=1/2$

$$x_{i+1} = x_i + hk_2$$

$$k_1 = f(t_i, x_i)$$

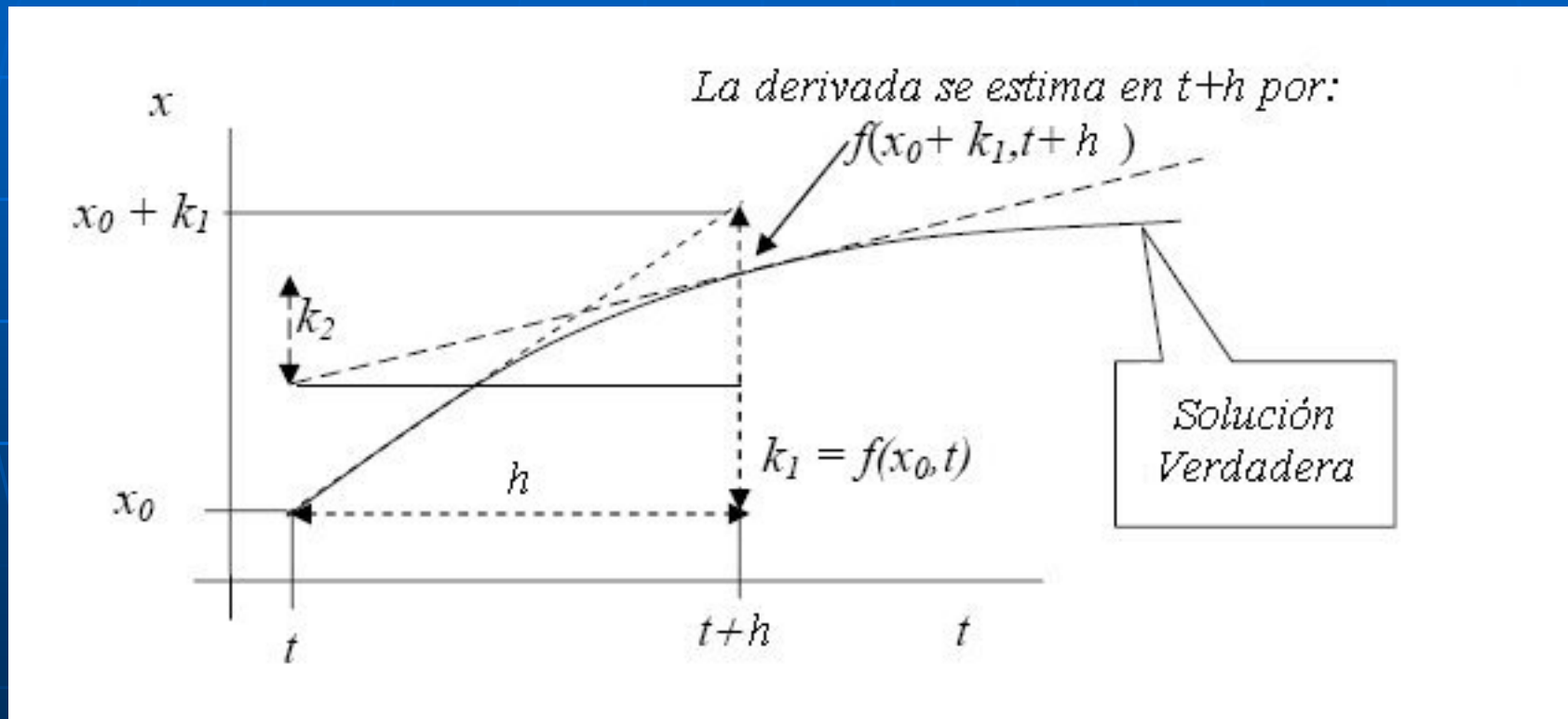
$$k_2 = f\left(t_i + \frac{h}{2}, x_i + \frac{h}{2}k_1\right)$$



Técnicas Explícitas de Orden Superior

Método R-K de 2do. Orden (III)

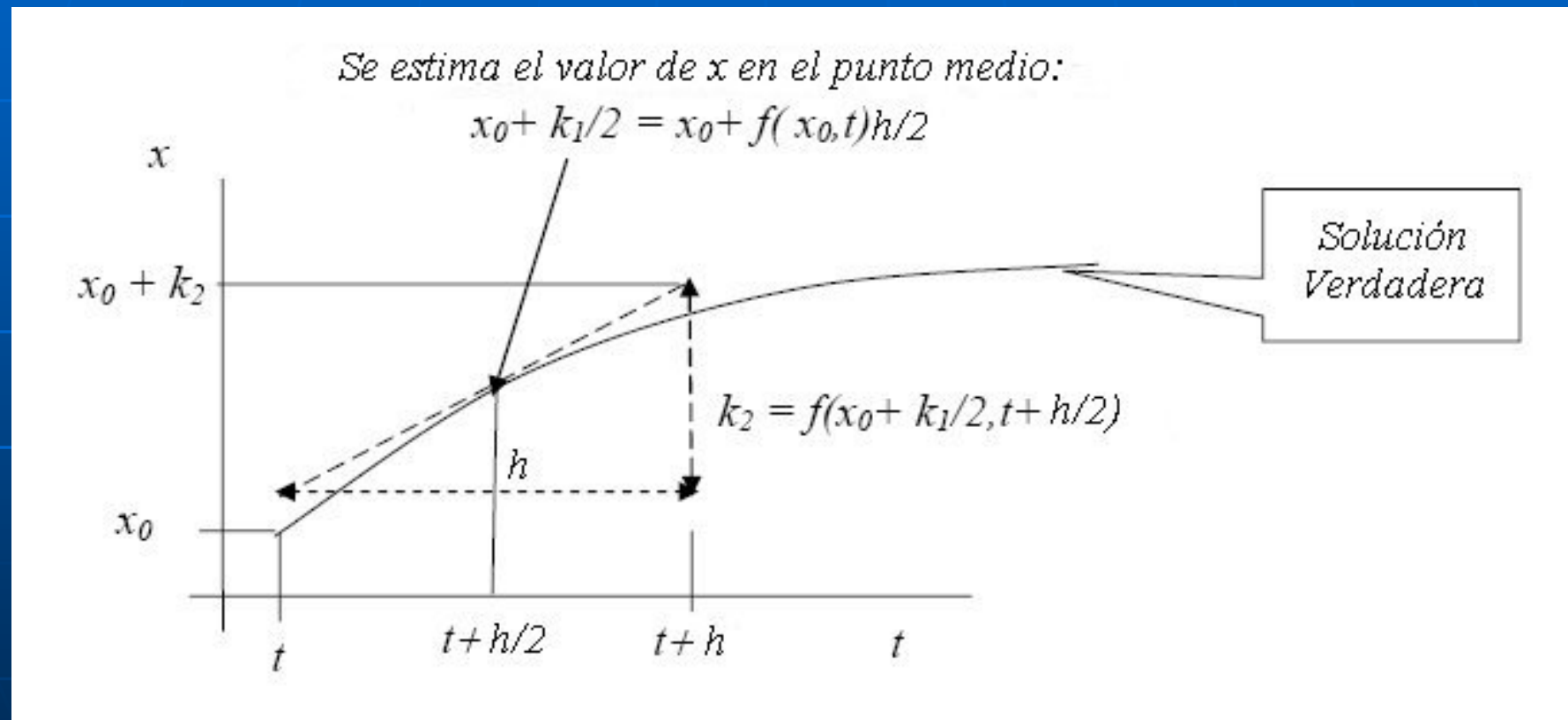
Caso $a=b=1/2$ (Esquema Crank – Nicholson)



Técnicas Explícitas de Orden Superior

Método R-K de 2do. Orden (IV)

Caso $b=1$ y $a=0$ (Esquema del punto medio)



Técnicas Explícitas de Orden Superior

Método R-K de 3er. Orden (I)

Algoritmo:

$$x_{i+1} = x_i + h\Phi$$

$$\Phi = (a k_1 + b k_2 + c k_3)$$

$$k_1 = f(t_i, x_i)$$

$$k_2 = f(t_i + ph, x_i + phk_1)$$

$$k_3 = f[t_i + rh, x_i + shk_2 + (r - s)hk_1]$$



Técnicas Explícitas de Orden Superior

Método R-K de 3er. Orden (II)

Kutta seleccionó las constantes de tal manera que el algoritmo adopta la siguiente forma:

$$x_{i+1} = x_i + \frac{h}{6}(k_1 + 4k_2 + k_3)$$

$$k_1 = f(t_i, x_i)$$

$$k_2 = f\left(t_i + \frac{h}{2}, x_i + \frac{h}{2}k_1\right)$$

$$k_3 = f(t_i + h, x_i + 2hk_2 - hk_1)$$



Técnicas Explícitas de Orden Superior

Método R-K de 4to. Orden

De los diversos algoritmos de 4to. Orden, el siguiente es atribuido a Kutta :

$$x_{i+1} = x_i + \frac{h}{6} (k_1 + 2k_2 + 2k_3 + k_4)$$

$$k_1 = f(t_i, x_i)$$

$$k_2 = f\left(t_i + \frac{h}{2}, x_i + \frac{h}{2}k_1\right)$$

$$k_3 = f\left(t_i + \frac{h}{2}, x_i + \frac{h}{2}k_2\right)$$

$$k_4 = f(t_i + h, x_i + hk_3)$$



Técnicas Explícitas de Orden Superior

Variantes del Método R-K de 4to. Orden (I)

$$x_{i+1} = x_i + \frac{h}{6} \left[k_1 + 2 \left(1 - \frac{1}{\sqrt{2}} \right) k_2 + 2 \left(1 + \frac{1}{\sqrt{2}} \right) k_3 + k_4 \right]$$

$$k_1 = f(t_i, x_i)$$

$$k_2 = f \left(t_i + \frac{h}{2}, x_i + \frac{h}{2} k_1 \right)$$

$$k_3 = f \left[t_i + \frac{h}{2}, x_i + \left(-\frac{1}{2} + \frac{1}{\sqrt{2}} \right) h k_1 + \left(1 - \frac{1}{\sqrt{2}} \right) h k_2 \right]$$

$$k_4 = f \left[t_i + h, x_i - \frac{h}{\sqrt{2}} k_2 + \left(1 + \frac{1}{\sqrt{2}} \right) h k_3 \right]$$



Técnicas Explícitas de Orden Superior

Variantes del Método R-K de 4to. Orden (II)

$$x_{i+1} = x_i + \frac{h}{8}(k_1 + 3k_2 + 3k_3 + k_4)$$

$$k_1 = f(t_i, x_i)$$

$$k_2 = f\left(t_i + \frac{h}{3}, x_i + \frac{h}{3}k_1\right)$$

$$k_3 = f\left(t_i + \frac{2h}{3}, x_i - \frac{h}{3}k_1 + hk_2\right)$$

$$k_4 = f(t_i + h, x_i + hk_1 - hk_2 + hk_3)$$



Técnicas Explícitas de Orden Superior

Errores de Truncamiento del Método R-K (I)

- Dado que los algoritmos RK de orden m se generaron requiriendo que la expresión $x_{i+1} = x_i + h \Phi(t_i, x_i, h)$ coincidiese con la expansión en serie de Taylor de la solución $x(t)$ hasta los términos de orden h^m , el error local de truncamiento viene expresado a través de la siguiente expresión:

$$|e| = K h^{m+1} + O(h^{m+2}).$$

- La constante K depende (usualmente de una manera complicada) de $f(t, x)$ y sus derivadas parciales de orden superior.
- Si uno supone que h es suficientemente pequeño como para que el error sea dominado por el primer término, entonces es posible encontrar cotas para K .
- En general tales cotas dependen de las cotas para $f(t, x)$ y sus derivadas parciales y del particular algoritmo R-K utilizado.



Técnicas Explícitas de Orden Superior

Errores de Truncamiento del Método R-K (II)

- Ralston demostró que particulares elecciones de los parámetros libres en las ecuaciones indeterminadas para las constantes del algoritmo minimizan el límite superior de K .
- Para poder elegir un tamaño razonable de paso de integración, se necesita efectuar una estimación del error que se comete en la integración a través de un paso.
- El tamaño del paso debería ser:
 - 1) Suficientemente pequeño como para alcanzar la exactitud requerida por la solución.
 - 2) Suficientemente grande como para mantener bajo control los errores de redondeo.

Este último factor es muy importante, particularmente cuando la EDO es complicada y cada evaluación de la derivada requiere un tiempo sustancial de cálculo.



Técnicas Explícitas de Orden Superior

Métodos de Runge - Kutta

Conclusiones:

- Las técnicas de Runge - Kutta no son únicas.
- El método de RK de 2do. Orden es más exacto que el Método Explícito de Euler.
- Fehlberg utilizó una técnica RK de 6 evaluaciones de la derivada (regla de 5to. orden), y mostró que 4 de esas evaluaciones podían recombinarse para suministrar una regla de 4to. Orden.
- Esto es muy útil debido a que la diferencia entre las dos estimaciones determina un error que puede utilizarse en el control adaptativo del tamaño de paso.
- La técnica de Fehlberg se halla implementada en la rutina de MATLAB **ode45.m**

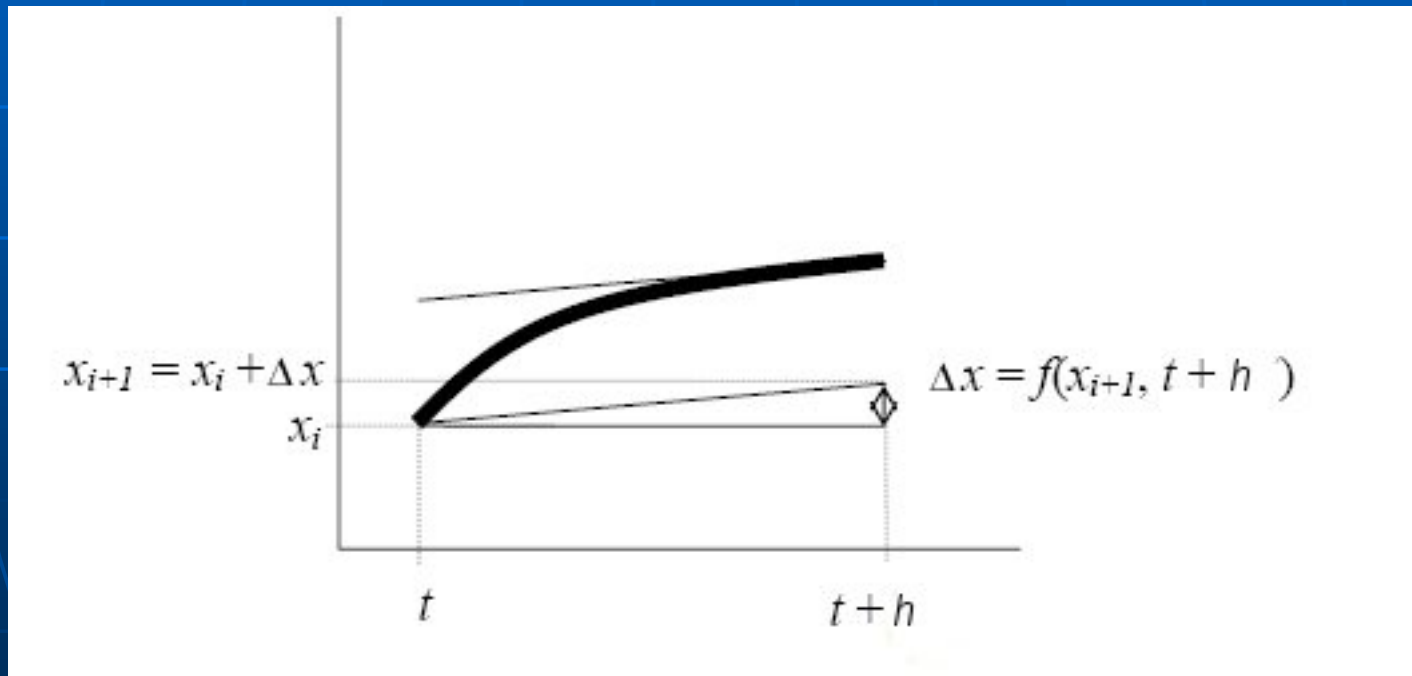


Método Implícito de Euler (I)

(Backward Euler)

Algoritmo: $x_{i+1} = x_i + h f(t_{i+1}, x_{i+1})$

h es el paso de integración



Método Implícito de Euler (II)

(Backward Euler)

- Para este caso se requiere llamar a un *solver (fsolve)* para obtener el nuevo valor de $x(t)$ al final de cada paso de integración.
- Sin embargo esto conduce a un esquema espaciado/integración que es *incondicionalmente estable* si $\lambda < 0$, y *condicionalmente estable* si $\lambda > 0$.
- La solución de nuestra problema de prueba utilizando el esquema implícito de Euler se muestra a continuación, para $h = 2$ ($\lambda = -1$).
- Si comparamos esta solución con la obtenida a través del esquema explícito de Euler, vemos que si bien los resultados no son particularmente exactos, al menos el esquema numérico es al menos estable.



Ejemplo 01

Método Implícito de Euler

Archivo de Comando de MATLAB:

Reactor.m: Archivo de comandos que llama a la función **EulerImplicit.m**.

Método:

EulerImplicit.m: Función que implementa el algoritmo implícito de Euler.

Funciones:

TestFunction.m: Función que implementa el lado derecho de la EDO explícita (llamada por **EulerImplicit.m**).

funToSolve.m: Función que calcula el residuo en el esquema implícito (llamada por **EulerImplicit.m**).

fsolve.m: Función que resuelve la ecuación implícita para obtener el valor actualizado de x .



Ejemplo 01

Método Implícito de Euler

Ejecución del Programa en MATLAB

>> Reactor

Ingrese el valor inicial de $x(t)$: 1

Ingrese el tiempo inicial t_0 : 0

Ingrese el tiempo final t_f : 10

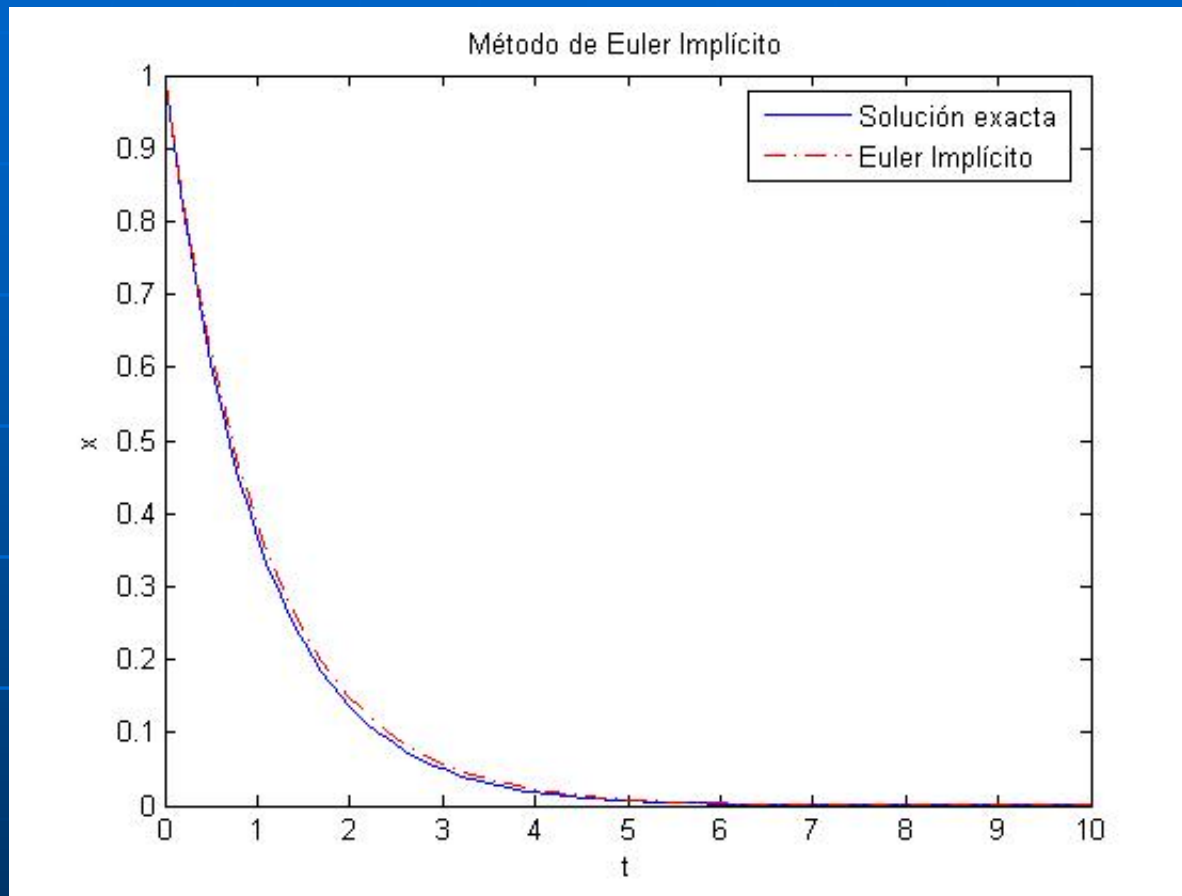
Ingrese el número de pasos: 100

Ingrese la función de prueba: 'TestFunction'



Ejemplo 01

Método Implícito de Euler

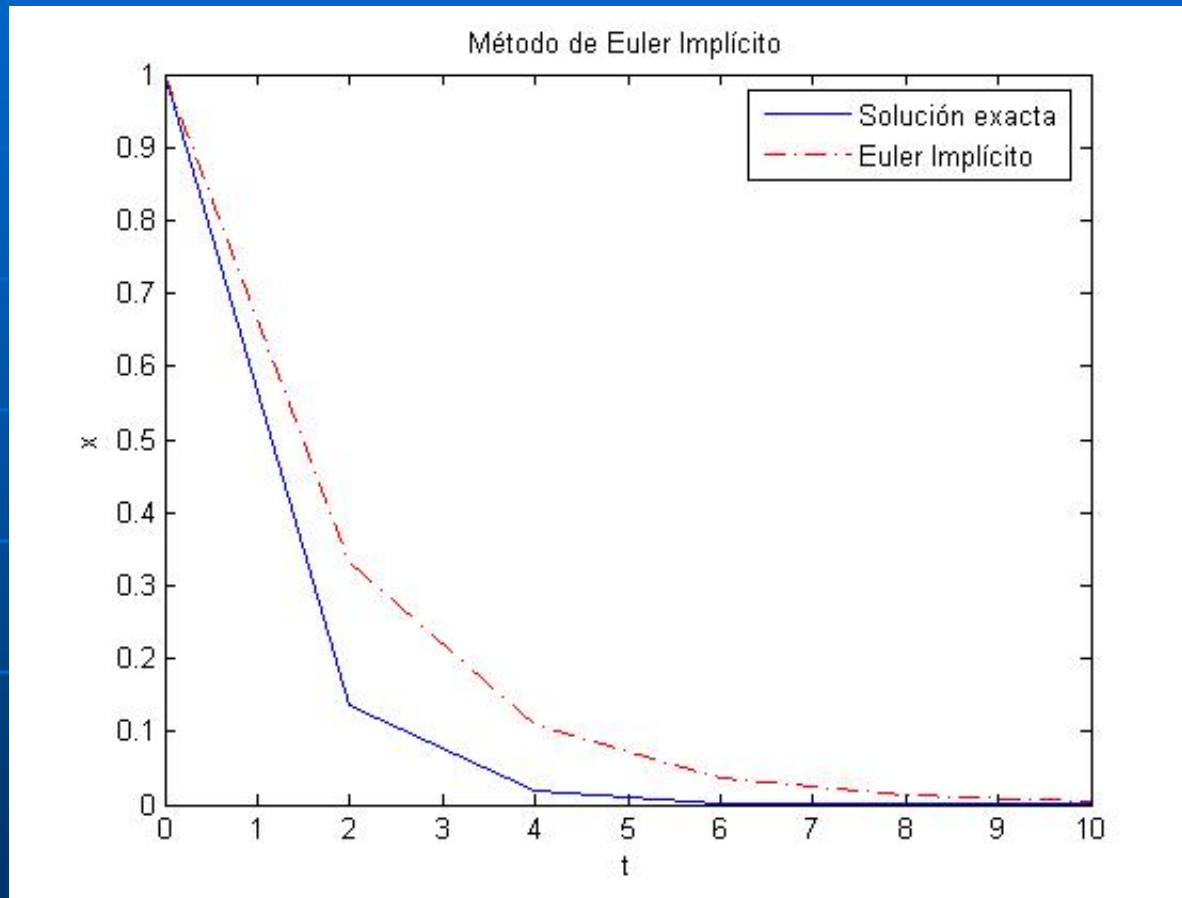


Solución del problema utilizando 100 pasos sobre 10 seg.



Ejemplo 01

Método Implícito de Euler



Solución del problema utilizando 5 pasos sobre 10 seg. mediante el esquema implícito de Euler.



Estabilidad del Método Implícito de Euler (I)

- Nuevamente, la solución exacta de la ecuación:

$$\frac{dx}{dt} = \lambda x \text{ para } x(t=0) = 1$$

es:

$$x^{exacta} = e^{\lambda t}$$

- Para $\lambda < 0$, la solución debería decaer a cero. Deseamos que nuestra solución numérica haga lo mismo. Para el método implícito de Euler el algoritmo viene dado por:

$$x_{i+1} = x_i + h f_{i+1} = x_i + \lambda h x_{i+1}$$

- Para que la solución numérica tienda a cero, entonces se debe verificar que:

$$x_{i+1} = x_i \frac{1}{(1 - \lambda h)}$$

- Por lo tanto, para $\lambda < 0$, la solución numérica siempre tenderá a cero para cualquier tamaño de paso.



Estabilidad del Método Implícito de Euler (II)

- Un desarrollo más exacto de la solución alrededor de t_{i+1} suministra:

$$x(t_i) = x(t_{i+1}) - h \left(\frac{dx}{dt} \right)_{t_{i+1}, x(t_{i+1})} + \frac{h^2}{2} \left(\frac{d^2 x}{dt^2} \right)_{t_{i+1}, x(t_{i+1})} + O(h^3)$$

donde $O(h^3)$ es el error local de truncamiento.

- El error en la iteración $i+1$ es: $e_{i+1} = x_{i+1} - x(t_{i+1})$
de donde resulta:

$$e_{i+1} = \frac{e_i + \frac{h^2}{2} \lambda^2 x(t_{i+1})}{(1 - \lambda h)} \approx \frac{1}{(1 - \lambda h)} e_i$$

- Si el denominador tiene módulo mayor que la unidad el error decaerá.
- Esto se cumplirá siempre que $\lambda < 0$; sin embargo esa condición se cumplirá además para $\lambda > 0$ cuando $\lambda h > 2$.



Estabilidad del Método Implícito de Euler (III)

Concluimos entonces que el método es:

- 1) No muy exacto (*low order method*).
- 2) Incondicionalmente estable si $\lambda < 0$. El método es estable cualquiera sea el valor de h .
- 3) Condicionalmente estable para $\lambda > 0$ si $\lambda h > 2$.
- 4) Por lo tanto el esquema implícito es mucho más estable que el esquema explícito.



Técnicas Implícitas de Orden Superior

Método Euler – Gauss (I)

- Modificación del Método de Euler debida a Gauss.
- La pobre aproximación de los métodos de Euler (explícito e implícito) se puede mejorar si se evalúa la integral de $f(t,x)$ mediante la siguiente fórmula:

$$x_{i+1}^{EGM} = x_i + \frac{h}{2} [f(t_i, x_i) + f(t_{i+1}, x_{i+1})] = \frac{1}{2} (x_{i+1}^{FEM} + x_{i+1}^{BEM})$$

- Para determinar el valor de esta expresión se requiere conocer $f(t_{i+1}, x_{i+1})$ por consiguiente se trata de un *método implícito*.



Técnicas Implícitas de Orden Superior

Método Euler – Gauss (II)

Algoritmo Predictor - Corrector:

1) Etapa predictora:

$$x_{i+1}^{(p)} = x_i^{(c)} + h f(t_i, x_i^{(c)})$$

2) Etapa correctora:

$$x_{i+1}^{(c)} = x_i^{(c)} + \frac{h}{2} \left[f(t_i, x_i^{(c)}) + f(t_{i+1}, x_{i+1}^{(p)}) \right] ; i = 1, 2, \dots,$$



Técnicas Implícitas de Orden Superior

Método Euler – Gauss

- El error absoluto en cada iteración es:

$$e_{i+1}^{EGM} = e_i^{EGM} \left(\frac{1 + \lambda h}{1 - \lambda h} \right) + O(h^3)$$

- Como en el Método Implícito de Euler (BEM), el Método de Euler-Gauss (EGM) es *incondicionalmente estable* para todo $\lambda < 0$, pero el factor de amplificación tenderá a -1 cuando $\lambda \rightarrow -\infty$, lo cual nos indica que se está cerca de la inestabilidad.
- Por lo tanto el EGM puede no ser mejor que el BEM para ecuaciones suficientemente *stiff* ($\lambda \ll -1$).



Técnicas Implícitas de Orden Superior

Métodos de Múltiples Evaluaciones (I)

- Si extendemos el algoritmo Euler-Gauss a órdenes superiores podríamos perder el intervalo infinito de estabilidad.
- Para obtener un algoritmo implícito de orden superior al Método Euler-Gauss debemos utilizar un enfoque diferente.
- Consideremos la expansión en serie de Taylor de la solución $x(t)$:

$$x(t_{i+1}) = x(t_i) + hx'(t_i) + \frac{h^2}{2!}x''(t_i) + \frac{h^3}{3!}x'''(t_i) + \dots +$$

- El enfoque de paso múltiple implica seguir la trayectoria de cada uno de los términos de la serie (la función y sus derivadas) en cada etapa del proceso de integración.



Técnicas Implícitas de Orden Superior

Métodos de Múltiples Evaluaciones (II)

- Definimos el siguiente vector:

$$\underline{\hat{x}}(t_i) = \begin{pmatrix} x(t_i) \\ hx'(t_i) \\ \frac{1}{2}h^2 x''(t_i) \\ \frac{1}{6}h^3 x'''(t_i) \end{pmatrix}$$

- Queremos estimar $\underline{\hat{x}}(t_{i+1})$ dado que conocemos $\underline{\hat{x}}(t_i)$.



Técnicas Implícitas de Orden Superior

Métodos de Múltiples Evaluaciones (III)

- Podemos expandir cada una de esas derivadas en series de Taylor:

$$h x'(t_{i+1}) = h x'(t_i) + h^2 x''(t_i) + \frac{h^3}{2} x'''(t_i) + \frac{h^4}{6} x^{(IV)}$$

$$\frac{h^2}{2} x''(t_{i+1}) = \frac{h^2}{2} x''(t_i) + \frac{h^3}{2} x'''(t_i) + \frac{h^4}{4} x^{(IV)}$$

$$\frac{h^3}{6} x'''(t_{i+1}) = \frac{h^3}{6} x'''(t_i) + \frac{h^4}{6} x^{(IV)}$$

donde hemos ignorado los términos de $O(h^5)$.



Técnicas Implícitas de Orden Superior

Métodos de Múltiples Evaluaciones (IV)

- Podemos escribir este conjunto de ecuaciones en forma vectorial:

$$\underline{\hat{x}}(t_{i+1}) = \underbrace{\begin{pmatrix} 1 & 1 & 1 & 1 \\ 0 & 1 & 2 & 3 \\ 0 & 0 & 1 & 3 \\ 0 & 0 & 0 & 1 \end{pmatrix}}_{\underline{\underline{B}}} \hat{x}(t_i) + \begin{pmatrix} 1 \\ 4 \\ 6 \\ 4 \end{pmatrix} \frac{h^4}{24} x^{(IV)}$$



Técnicas Implícitas de Orden Superior

Métodos de Múltiples Evaluaciones (V)

➤ Definamos ahora:

$$\underline{\hat{x}}_{i+1}^{(p)} \equiv \underline{\underline{B}} \cdot \underline{\hat{x}}(t_i)$$

➤ Luego tenemos:

$$\underline{\hat{x}}_{i+1} = \underline{\hat{x}}_{i+1}^{(p)} + \begin{pmatrix} 1 \\ 4 \\ 6 \\ 4 \end{pmatrix} \underbrace{\frac{h^4}{24} x^{(IV)}}_{a_n}$$



Técnicas Implícitas de Orden Superior

Métodos de Múltiples Evaluaciones (VI)

- Desafortunadamente, la cantidad: $\frac{h^4}{24} x^{(IV)}$ es desconocida.
- La consideraremos constante y la elegiremos de manera que satisfaga nuestra ecuación diferencial.
- Recordemos que $x' = f(t, x)$ es nuestra EDO. Por consiguiente la segunda componente del vector debe satisfacer esta ecuación, por lo tanto:

$$\begin{pmatrix} \hat{x}_{i+1} \\ \underline{x}_{i+1} \end{pmatrix}_2 \equiv \begin{pmatrix} \hat{x}_{i+1}^{(p)} \\ \underline{x}_{i+1} \end{pmatrix}_2 + 4a_n = hf(t_{i+1}, x_{i+1})$$



Técnicas Implícitas de Orden Superior

Métodos de Múltiples Evaluaciones (VII)

- De aquí obtenemos:

$$4a_n = hf_{i+1} - \left(\underline{x}_{i+1}^{(p)} \right)_2$$

- Si introducimos esta cantidad en la expresión para \underline{x}_{i+1} obtenemos:

$$\hat{\underline{x}}_{i+1} \equiv \underline{\underline{B}} \cdot \hat{\underline{x}}_i + \underline{r} \left[hf_{i+1} - \left(\hat{\underline{x}}_{i+1}^{(p)} \right)_2 \right]$$

donde:

$$\underline{r} = \begin{pmatrix} 1/4 \\ 1 \\ 3/2 \\ 1 \end{pmatrix}$$



Técnicas Implícitas de Orden Superior

Métodos de Múltiples Evaluaciones (VIII)

- Dado que hemos ignorado los términos de $O(h^5)$ en las series de Taylor, el error local es $O(h^5)$ y el método es de 4to. Orden.
- El método es implícito en \underline{x}_{i+1} .
- Numéricamente siempre es *inestable*.
- La particular elección del vector \underline{r} determina la regla.
- Dos elecciones de \underline{r} comúnmente utilizadas son:
 - **Adams Moulton:** $\underline{r} = (3/8, 1, 6/11, 1/6)^T$
 - **Gear:** $\underline{r} = (6/11, 1, 6/11, 1/11)^T$



Técnicas Implícitas de Orden Superior

Métodos de Múltiples Evaluaciones (IX)

Conclusiones:

- **Adams Moulton** posee una región finita de estabilidad y es bueno para problemas *no - stiff*.
- **Gear** tiene una región infinita de estabilidad (no obstante es menos exacto que **Adams Moulton**) y es el método frecuentemente utilizado para problemas stiff.



Control Adaptativo del Tamaño de Paso (I)

- Como en el caso de los métodos de cuadratura (integración de funciones conocidas), queremos elegir un tamaño de paso que nos permita alcanzar una exactitud deseada de la solución.
- Desafortunadamente, el tamaño requerido del paso puede cambiar de un punto a otro a medida que la integración avanza.
- Esto conduce a la necesidad de tener que utilizar esquemas adaptativos para controlar el tamaño del paso.
- Para poder cambiar el tamaño del paso necesitamos efectuar una estimación del error local de integración.



Control Adaptativo del Tamaño de Paso (II)

- Consideremos el método de Runge - Kutta de 2do. Orden (esquema de Crank – Nicholson):

$$x_{i+1} = x_i + \frac{h}{2}(k_1 + k_2)$$

$$k_1 = f(t_i, x_i)$$

$$k_2 = f(t_i + h, x_i + hk_1)$$

- El error local de esta regla es $O(h^3)$.
- Si efectuamos una evaluación más de la derivada en el punto medio del intervalo de integración, resulta:

$$k_3 = f\left[t_i + \frac{h}{2}, x_i + \frac{h}{4}(k_1 + k_2)\right]$$



Control Adaptativo del Tamaño de Paso (III)

- Podemos combinar esta evaluación de la derivada con las otras dos para obtener una regla de tercer orden:

$$x_{i+1} = x_i + \frac{h}{6}(k_1 + 4k_3 + k_2)$$

que es similar a la Regla de Simpson de cuadratura.

- El error local es menor que la diferencia entre estas dos estimaciones:

$$\text{error} \approx \frac{h}{2}(k_1 + k_2) - \frac{h}{6}(k_1 + 4k_3 + k_2) = \frac{1}{3}(k_1 - 2k_3 + k_2)$$

- Si el error es más grande que algún valor umbral elegido, entonces procedemos a reducir el tamaño del paso. Si el error es menor que la tolerancia, podemos incrementar el tamaño del paso.



Control Adaptativo del Tamaño de Paso (IV)

¿Cómo determinamos el tamaño del paso óptimo?

- 1) El error local en la regla de 2do. Orden (que domina la estimación del error) es $O(h^3)$.
- 2) Si la estimación del error local es Δ y el máximo error permitido es τ , entonces:

$$\left(\frac{h_{opt}}{h}\right)^3 \approx \frac{\tau}{\Delta}$$

por consiguiente:

$$h_{opt} \approx h \left(\frac{\tau}{\Delta}\right)^{1/3}$$

- 3) En la práctica podemos elegir nuestro nuevo h un poco más pequeño que este valor, digamos:

$$h_{opt} = 0.9h \left(\frac{\tau}{\Delta}\right)^{1/3}$$



Control Adaptativo del Tamaño de Paso (V)

¿Cómo determinamos el tamaño del paso óptimo?
(cont.)

- 4) También debemos establecer un límite superior y uno inferior para h . El límite superior será alguna fracción del intervalo total de integración. El límite inferior, sirve como *flag* para advertirnos cuando nos aproximamos a la singularidad.
- 5) Este método se halla implementado en la rutina *ode23.m* de MATLAB.



Métodos de Paso Múltiple (I)

- Existen otras familias de métodos que involucran en el cálculo de x_{i+1} el valor de la derivada en puntos anteriores: t_i, t_{i-1}, t_{i-2} , etc. A estos métodos se los denomina también *métodos explícitos* o *métodos predictores*.
- Sin embargo, existen otras familias de métodos que involucran en el cálculo de x_{i+1} el valor de la derivada en el punto t_{i+1} , esto es $f_{i+1} = f(t_{i+1}, x_{i+1})$, que depende a su vez del valor de la solución en el punto que se desea calcular, x_{i+1} , el cual no se conoce. Ello implica que debe recurrirse a métodos iterativos.
- Es por eso que a estos métodos se los denomina también *implícitos* a diferencia de los explícitos anteriormente mencionados.



Métodos de Paso Múltiple (II)

- Para una integración que involucra $k+1$ intervalos que terminan en el punto t_{i+1} , aplicando sucesivamente el Método Explícito de Euler obtenemos:

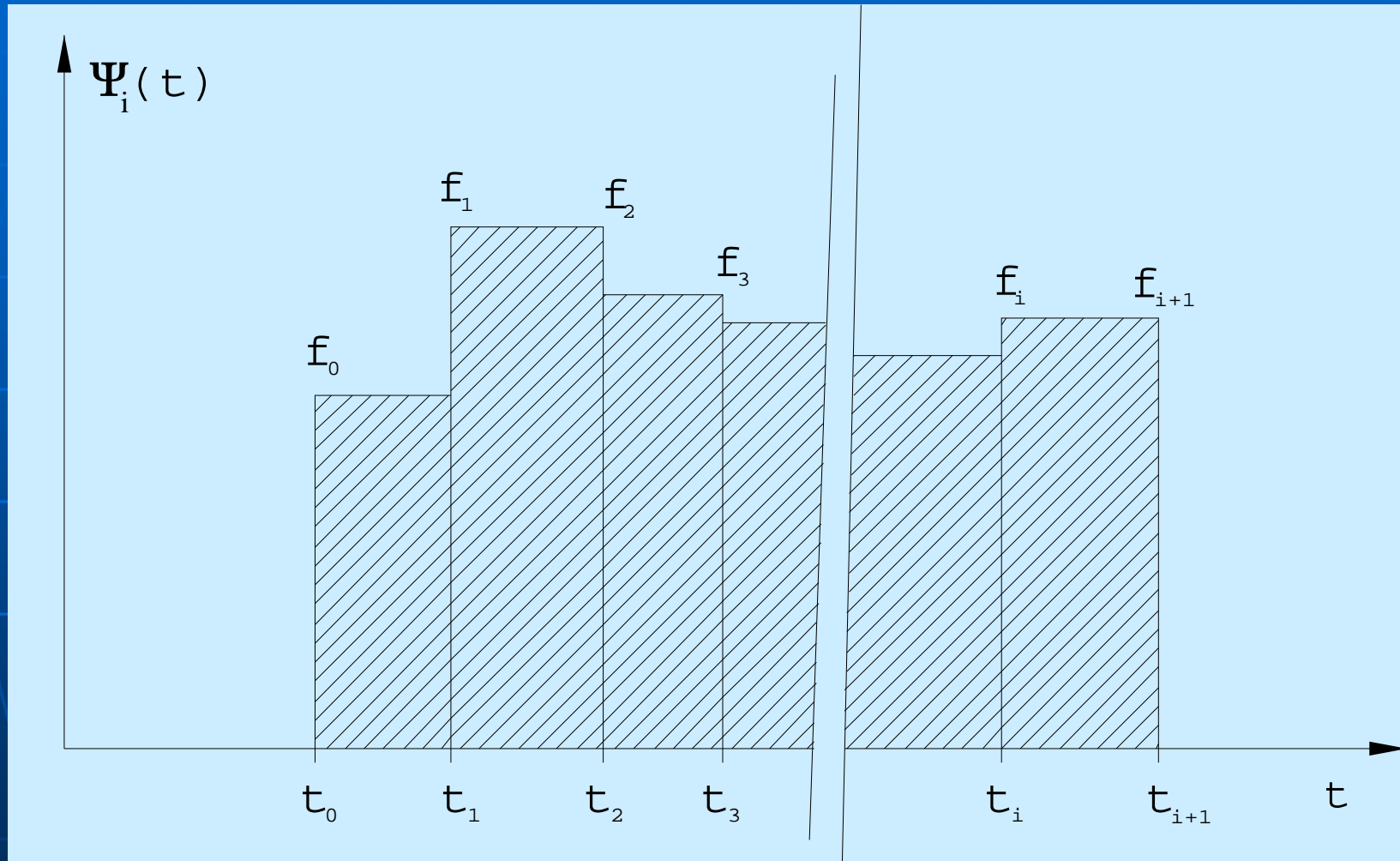
$$x_{i+1} = x_{i-k} + \int_{t_{i-k}}^{t_{i+1}} \Psi_i(t) dt$$

donde $\Psi_i(t)$ es una función escalonada con ordenadas $f_j = f(t_j, x_j)$, sobre los intervalos semiabiertos $[t_j, t_{j+1})$; $j = i - k, \dots, i$.

- La integral puede asimilarse como el área bajo la curva indicada en la Figura siguiente:



Métodos de Paso Múltiple (III)



Métodos de Paso Múltiple (IV)

- Todos los métodos de paso múltiple pueden describirse a través de la siguiente ecuación:

$$x_{i+1} = x_{i-k} + \int_{t_{i-k}}^{t_{i+1}} \Phi(t) dt$$

donde $\Phi(t)$ es un polinomio de interpolación, que pasa por los puntos (t_i, f_i) , apropiado para efectuar la integración aproximada.

- Existen varias metodologías (Lagrange, Newton, etc.), basadas, en general, en tres estrategias características para determinar los coeficientes del polinomio de grado p :

$$\Phi = \sum_{n=0}^p a_n t^n$$

- 1) Interpolación por diferencias hacia adelante.
- 2) Interpolación por diferencias hacia atrás.
- 3) Interpolación por diferencias centradas.



Métodos de Paso Múltiple (V)

- Si se suponen a todos los puntos igualmente espaciados ($\Delta t = h$), cuando utilizamos diferencias hacia adelante, comenzamos en el extremo inferior del intervalo, tomando $\Delta t_1 = (t_1 - t_0)$, $\Delta t_2 = (t_2 - t_1)$, y así sucesivamente.
- Conocidos los puntos t_0, t_1, \dots, t_i , se busca el polinomio que pase por todos los puntos, *extrapolando* el valor de (t_{i+1}, f_{i+1}) .
- A esta fórmula de integración se la conoce como *fórmula de integración abierta*, y es muy sencillo demostrar (aplicando la fórmula de interpolación de Newton para polinomios de grado p) las siguientes expresiones:

1) Para $p=3$:

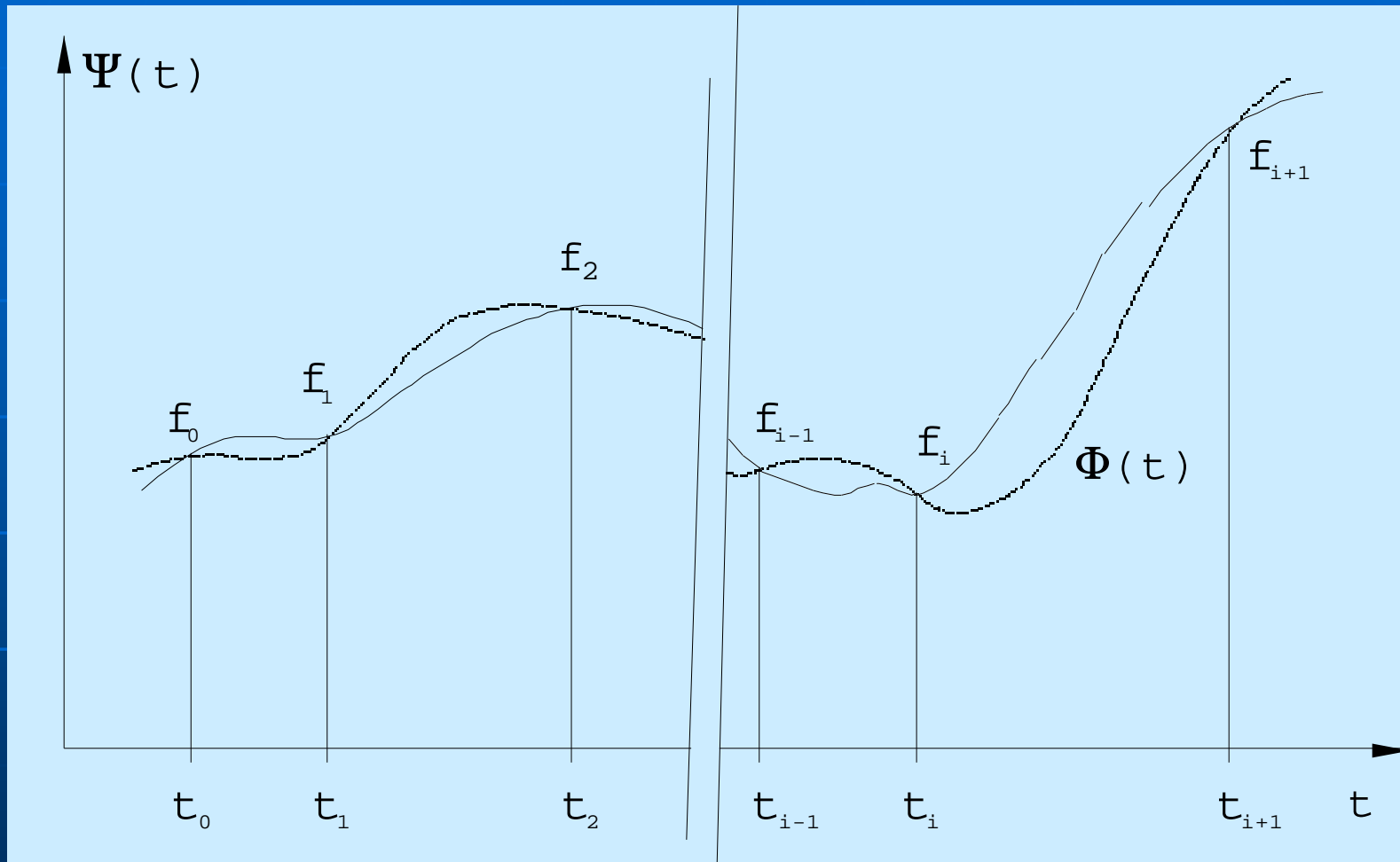
$$x_{i+1} = x_i + \frac{h}{24} (55 f_i - 59 f_{i-1} + 37 f_{i-2} - 9 f_{i-3}) ; \varepsilon = O(h^5)$$

2) Para $p=5$:

$$x_{i+1} = x_{i-5} + \frac{3}{10} h (11 f_i - 14 f_{i-1} + 26 f_{i-2} - 14 f_{i-3} + 11 f_{i-4}) ; \varepsilon = O(h^7)$$



Métodos de Paso Múltiple (VI)



Integración abierta paso múltiple.



Métodos de Paso Múltiple (VII)

- Si en cambio utilizamos *fórmulas de integración cerradas*, el polinomio de interpolación ahora no sólo debe pasar por los puntos conocidos: $t_0, t_1, t_2, \dots, t_i$ sino además por (t_{i+1}, f_{i+1}) .
- Aquí utilizamos las fórmulas de diferencias hacia atrás, basándonos en el extremo derecho del intervalo $\Delta t_1 = (t_{i+1} - t_i)$, $\Delta t_2 = (t_i - t_{i-1})$ y así sucesivamente.
- Nuevamente, es fácil demostrar a partir del procedimiento de Newton para interpolación de un polinomio de grado p usando diferencias finitas (propagación hacia atrás), las siguientes fórmulas para la estimación:

1) Para $p=3$:

$$x_{i+1} = x_i + \frac{h}{24}(9f_{i+1} + 19f_i - 5f_{i-1} + f_{i-2}) ; \varepsilon = O(h^5)$$

2) Para $p=5$:

$$x_{i+1} = x_i + \frac{2}{45}h(7f_{i+1} + 32f_i + 12f_{i-1} + 32f_{i-2} + 7f_{i-3}) ; \varepsilon = O(h^7)$$



Métodos de Paso Múltiple (VIII)

- Como puede observarse, en estas dos últimas expresiones para el cálculo de x_{i+1} , debemos disponer en el miembro derecho, de los valores de f_{i+1} , que a su vez dependen de x_{i+1} .
- Esto nos conduce a la necesidad de un método iterativo, que implica más esfuerzo de cálculo, pero permite, en general, reducir el error de integración.



