

## TEMA 3:

# PROBLEMAS ORDINARIOS DEL VALOR INICIAL EN INGENIERÍA QUÍMICA: ESTUDIO Y COMPARACIÓN DE MÉTODOS NUMÉRICOS DE RESOLUCIÓN.

1. MÉTODOS NUMÉRICOS APLICABLES A PROBLEMAS ODE-IVP: PRESENTACIÓN Y GENERALIDADES
2. MÉTODOS EXPLÍCITOS APLICABLES A ODE-IVP
  - 2.1. Métodos de una etapa
    - 2.1.1. Algoritmo de Euler explícito
    - 2.1.2. Métodos explícitos de orden superior: Algoritmos de Runge-Kutta.
  - 2.2. Métodos de Etapas Múltiples
3. MÉTODOS IMPLÍCITOS APLICABLES A ODE-IVP
  - 3.1. Algoritmo de Euler implícito
  - 3.2. Algoritmo del trapecio (método de Euler modificado)
4. ESTRATEGIAS DE CONTROL DEL ERROR EN LAS APROXIMACIONES
5. ESTABILIDAD Y EXACTITUD EN LOS MÉTODOS NUMÉRICOS IVP
6. BIBLIOGRAFÍA RECOMENDADA

---

Asignatura: Cálculo Avanzado de Procesos Químicos.  
Titulación: Ingeniería Química  
Curso: Cuarto  
Cuatrimestre: Primero

## 1. MÉTODOS NUMÉRICOS APLICABLES A PROBLEMAS ODE-IVP: PRESENTACIÓN Y GENERALIDADES.

Dentro del área de Ingeniería Química, los problemas del tipo ODE-IVP representan casos de sistemas en los que UNA sola propiedad (Temperatura, concentración, etc.) varía en función de UNA sola variable (normalmente tiempo aunque también puede ser posición, por ejemplo, longitud de un reactor).

En estos problemas se conoce: a) cómo es la variación de la propiedad respecto a la variable:  $y' = f(t, y)$  y b) el valor de la propiedad en el primer valor de la variable (punto inicial del intervalo de integración):  $y(0) = y_0$ . Se busca la función que relacione el valor de la propiedad con cada valor de la variable:  $y = f(t)$ .

### Ejemplo 3.1.

El modelado matemático de reactores químicos es un área de I.Q. en la que generalmente se conoce la concentración de los reactivos al comienzo de la reacción (discontinuos) o a la entrada del reactor y se busca la concentración de salida. Ejemplos de este tipo de problemas son:

Reactores batch o semi-batch en estado dinámico en los que se lleven a cabo reacciones isotermas (Riggs, 1994):

$$\begin{cases} \frac{dn_a}{dt} = Q_0 C_{a0} - \frac{kn_a^2}{Q_0 t + v_0} \\ n_a(0) = 0 \end{cases}$$

Reactor flujo pistón isoterma y sin dispersión en estado estacionario en el que se lleva a cabo una reacción con velocidad  $r = -kC^2$  (Rice, 1995):

$$\begin{cases} u \frac{dC}{dz} = -kC^2 \\ C(0) = C_0 \end{cases}$$

Reactor de lecho fijo isoterma y sin dispersión en estado estacionario con velocidad de reacción:

$$-r = Pk_0 K_0 T \exp\left[\frac{(-Q - E_0)}{R_g T}\right] * C \quad (\text{Davis, 1984}):$$

$$\begin{cases} \frac{dC}{dz} = \rho \theta Pk_0 K_0 T \exp\left[\frac{(-Q - E_0)}{R_g T}\right] * C \\ C(0) = C_0 \end{cases}$$

Los problemas relacionados con el control de procesos químicos también se modelan mediante ecuaciones ODE-IVP ya que se busca la función que define el comportamiento de una variable desde el momento en que es sometida a una perturbación. Este tipo de problemas implica a ODEs tanto de primer como de segundo orden ( $y'' = f(t, y, y')$ ) por lo que se verán en el tema 4.

Existen múltiples métodos numéricos aplicables a problemas ordinarios del valor inicial (ODE-IVP), sin embargo todos ellos deben cumplir una serie de características comunes:

1. Se aplican siempre sobre ecuaciones de primer orden. Las ecuaciones de orden superior a uno deben transformarse en sistemas de ecuaciones de primer orden para poder ser resueltas.
2. Se aplican sobre la ecuación ( o ecuaciones) expresada en forma  $y' = f(x, y)$
3. Es necesario conocer el valor exacto de la variable dependiente en el primer punto del intervalo de integración. Este valor es conocido como condición inicial.
4. Las condiciones 2 y 3 definen la forma que debe tener un problema ODE-IVP para poder ser resuelto numéricamente:

$$\begin{cases} y' = f(x, y) \\ y(x_0) = y_0 \end{cases}$$

5. Los métodos numéricos aplicables a ODE-IVP se clasifican en dos grandes grupos:
  - a) Métodos explícitos.
  - b) Métodos implícitos.

En los métodos explícitos se utiliza información conocida  $(y_i, f(x_i, y_i))$  para determinar la aproximación en un nuevo punto  $(y_{i+1})$ .

En los métodos implícitos se utiliza información no conocida  $(y_{i+1}, f(x_{i+1}, y_{i+1}))$  para determinar la aproximación correspondiente a un nuevo punto  $(y_{i+1})$ .

El objetivo de este tema es presentar los fundamentos de los algoritmos utilizados para la resolución numérica de una ecuación del tipo ODE-IVP. Con estos conocimientos básicos en el tema 4 se presentarán las características de los algoritmos comerciales utilizados para la resolución de sistemas de ecuaciones ODEs y para ODEs de orden mayor de 1.

## 2. MÉTODOS EXPLÍCITOS APLICABLES A ODE-IVP

### 2.1. MÉTODOS EXPLÍCITOS DE UNA ETAPA

Los métodos de una etapa utilizan exclusivamente información contenida entre el nodo  $i$  y el nodo  $(i+1)$  para calcular el valor de la aproximación en el nodo  $(i+1)$ .

#### 2.1.1. Algoritmo de Euler explícito

Este algoritmo es el más sencillo de los métodos para problemas IVP pero tiene nula aplicación para problemas de dificultad media y no aparece en subrutinas comerciales.

Sin embargo se encuentra en todos los libros de texto ya que su mecánica es similar a la de otros algoritmos IVP más sofisticados pero más sencilla y por lo tanto permite una mayor comprensión de las formas de trabajo de estos algoritmos con menor esfuerzo.

Para aplicar el algoritmo de Euler partimos de una ecuación de la forma: 
$$\begin{cases} y' = f(x, y) \\ y(x_0) = y_0 \end{cases}$$

Trabajamos en un intervalo  $x_0 \leq x \leq x_n$  y asumimos que  $\frac{\partial f}{\partial x}$  es continua en el intervalo  $(x_0 - x_n)$  lo que nos garantiza que existe una única solución.

Si  $y(x)$  es la solución exacta su gráfica será una curva en el plano  $xy$  que pase por el punto  $(x_0, y_0)$ . Una solución numérica de esta ecuación será un conjunto de puntos  $[(x_i, u_i)]_{i=0}^n$  donde  $u_0 = y_0$  y cada punto  $(x_i, u_i)$  es una aproximación al punto  $(x_i, y_i)$  de la curva solución exacta (la solución numérica es sólo un conjunto de puntos y no proporciona información sobre el intervalo de puntos). La figura 3.1. muestra el significado gráfico de esta interpretación.

1. El primer paso para encontrar la solución aproximada mediante el método de Euler consiste en dividir el intervalo  $[x_0, x_n]$  en  $n$  subintervalos tales que:

$$h = \frac{x_n - x_0}{n}$$

$$x_i = x_{i-1} + h = x_0 + (i * h) \quad i = 1, 2, 3, \dots, n$$

donde  $h$  = tamaño de paso.

2. En cada punto  $x_i$  se cumple que si  $y(x)$  es la solución exacta de  $y'=f(x,y)$  utilizando la expresión de Taylor en el punto  $x_i$  se tiene :

$$y_{i+1} = y_i + (x_{i+1} - x_i)y_i' + \frac{(x_{i+1} - x_i)^2}{2!} y''(\epsilon_i), \quad \text{siendo } x_i \leq \epsilon_i \leq x_{i+1}$$

como: 
$$\begin{cases} y_i' = f(x_i, y_i) \\ h = (x_{i+1} - x_i) \end{cases} \Rightarrow y_{i+1} = y_i + h f(x_i, y_i) + \frac{h^2}{2} f''(\epsilon_i, y(\epsilon_i))$$

3. El algoritmo de Euler se obtiene al truncar esta expresión después del segundo término:

$u_{i+1} = u_i + h * f(x_i, u_i)$
$u_0 = y_0$ <span style="float: right;">Algoritmo de Euler Explícito</span>

En este método numérico el error que se comete en el cálculo de un nuevo punto  $u_{i+1}$  supuesto que el anterior es exacto (error de truncamiento local) es proporcional a  $h^2$  ya que el término que eliminamos de la expresión de Taylor para obtener la aproximación es proporcional a  $h^2$ :  $e_{i+1} = \mathcal{O}(h^2)$ .

La notación  $\mathcal{O}(\ )$  denota términos del orden indicado entre paréntesis.

Se dice que un método es exacto de orden P si su error de truncamiento local cumple:

$$e_{i+1} = \mathcal{O}(h^{P+1}),$$

por tanto el algoritmo de Euler es un método exacto de orden uno, lo que es lo mismo tiene una precisión de primer orden.

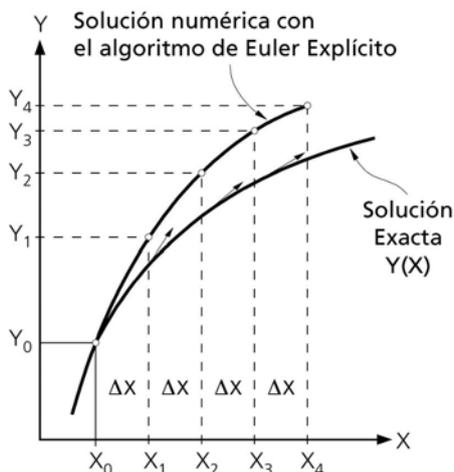


Figura 3.1. Relación entre la solución exacta y numérica de un problema ODE-IVP. [Reelaborado a partir de Riggs, 1994].

**2.1.2. Métodos explícitos de orden superior: Algoritmos de Runge-Kutta.**

Los métodos de Runge-Kutta (RK) son algoritmos explícitos que comprenden la evaluación de la función  $f$  en varios puntos comprendidos entre dos nodos consecutivos  $(x_i - x_{i+1})$ . La serie de métodos RK son los algoritmos más usados para la integración de ODEs.

La fórmula general de los algoritmos RK es:

$$\begin{cases} u_{i+1} = u_i + \sum_{j=1}^v w_j k_j \\ k_j = h f\left(x_i + c_j h, u_i + \sum_{l=1}^{j-1} a_{jl} k_l\right) \\ c_1 = 0 \end{cases}$$

Donde:

$w_j$  = constantes que ponderan la influencia de cada evaluación (pendientes).

$k_j$  = pendientes determinadas en diversos puntos del intervalo  $x_i - x_{i+1}$ .

$v$  = orden del método de Runge Kutta empleado (su valor determina la exactitud y complejidad de cálculo del método empleado).

$c_j$  = constantes

$a_{jl}$  = constantes

$h$  = tamaño de paso

**Ejemplo 3.2.:**

Un método de RK de orden dos consistirá en un algoritmo en el que:

$$v=2, \quad j=1, 2, \quad l=1$$

Por lo tanto:  $u_{i+1} = u_i + w_1 k_1 + w_2 k_2 \Rightarrow$  El valor de  $u_{i+1}$  se determina a partir del valor de  $u_i$  más DOS evaluaciones de valores de la función (pendientes  $k_1, k_2$ ) en dos puntos intermedios del intervalo de integración. Dichas pendientes contribuyen mediante los factores de ponderación  $w_1, w_2$  al valor de  $u_{i+1}$ .

El valor de las pendientes se determina:

$$k_1 = h * f(x_i + 0, u_i + 0)$$

$$k_2 = h * f(x_i + c_2 h, u_i + a_{21} k_1)$$

Se necesita determinar 4 constantes ( $w_1, w_2, c_2, a_{21}$ ) para aplicar un método de RK de orden 2.

Los diferentes algoritmos de RK provienen de realizar aproximaciones a la expansión en series de Taylor de la solución exacta:

$$y_{i+1} = y_i + h * f(x_i, y_i) + \frac{h^2}{2!} * f'(x_i, y_i) + \mathcal{O}(h^3)$$

Para crear un método de RK:

1. Aproximar la expansión de Taylor cortando la expansión en diferentes términos. El punto de corte determina el orden del método de RK.
2. Determinar los coeficientes  $w$ ,  $c$ ,  $a$ . Para ello se igualan los términos de la expansión de Taylor a los de la expresión de RK. Se obtienen así sistemas de  $n$  ecuaciones con  $m$  incógnitas donde  $m > n$ , por lo tanto para resolverlo es necesario dar valores arbitrarios a algunas incógnitas para que  $n=m$ . De esta forma se pueden generar múltiples algoritmos de RK del mismo orden de exactitud en función de los valores arbitrarios designados.

### Ejemplo 3.3.:

Obtener las expresiones correspondientes a los algoritmos de RK de orden 2 para los casos a) ( $c_2=0.5$ ) y b) ( $c_2=1$ ) así como la interpretación gráfica de los mismos.

Expresión RK para orden 2  $\Rightarrow$

$$u_{i+1} = u_i + w_1 k_1 + w_2 k_2$$

$$k_1 = h * f(x_i + 0, u_i + 0)$$

$$k_2 = h * f(x_i + c_2 h, u_i + a_{21} k_1)$$

Serie de Taylor truncada en el 2º término:  $y_x = y_{x0} + y'_{x0}(x - x_0) + y''_{x0} \frac{(x - x_0)^2}{2!} + \dots$

Coefficientes a determinar:  $w_1, w_2, c_2, a_{21}$

Para determinar estos coeficientes se igualan los términos del algoritmo general a la expansión de Taylor truncada en el segundo término obteniéndose tras un tratamiento matemático (que sobrepasa los objetivos de un curso de aplicación) un sistema de la forma:

$$\left. \begin{array}{l} w_1 + w_2 = 1 \\ w_2 * c_2 = 0,5 \\ w_2 * a_{21} = 0,5 \end{array} \right\} \text{3 ecuaciones y 4 incógnitas} \Rightarrow \text{se asignan valores arbitrarios a una de ellas.}$$

$$w_2 = 1$$

caso a) se asigna  $c_2 = 0,5 \Rightarrow$  entonces:  $w_1 = 0$ , entonces el algoritmo se expresa:

$$a_{21} = 0,5$$

$$u_{i+1} = u_i + 1 \cdot k_2$$

$$u_0 = y_0$$

$$k_1 = h \cdot f(x_i, u_i)$$

$$k_2 = h \cdot f\left(x_i + \frac{1}{2} \cdot h, u_i + \frac{1}{2} k_1\right) = h \cdot f\left(x_i + \frac{1}{2} \cdot h, u_i + \frac{1}{2} h f_i\right)$$

La interpretación gráfica de este algoritmo de RK se muestra en la figura 3.2.:

caso b) se asigna  $c_2 = 1 \Rightarrow$  entonces:  $\begin{cases} w_2 = 0,5 \\ w_1 = 0,5 \\ a_{21} = 1 \end{cases}$ , el algoritmo se expresa:

$$u_{i+1} = u_i + \frac{1}{2} \cdot k_1 + \frac{1}{2} \cdot k_2 = u_i + \frac{1}{2} h [f_i + f(x_i + h, u_i + h f_i)]$$

$$u_0 = y_0$$

$$k_1 = h \cdot f(x_i, u_i)$$

$$k_2 = h \cdot f(x_i + h, u_i + k_1) = h \cdot f(x_i + h, u_i + h f_i)$$

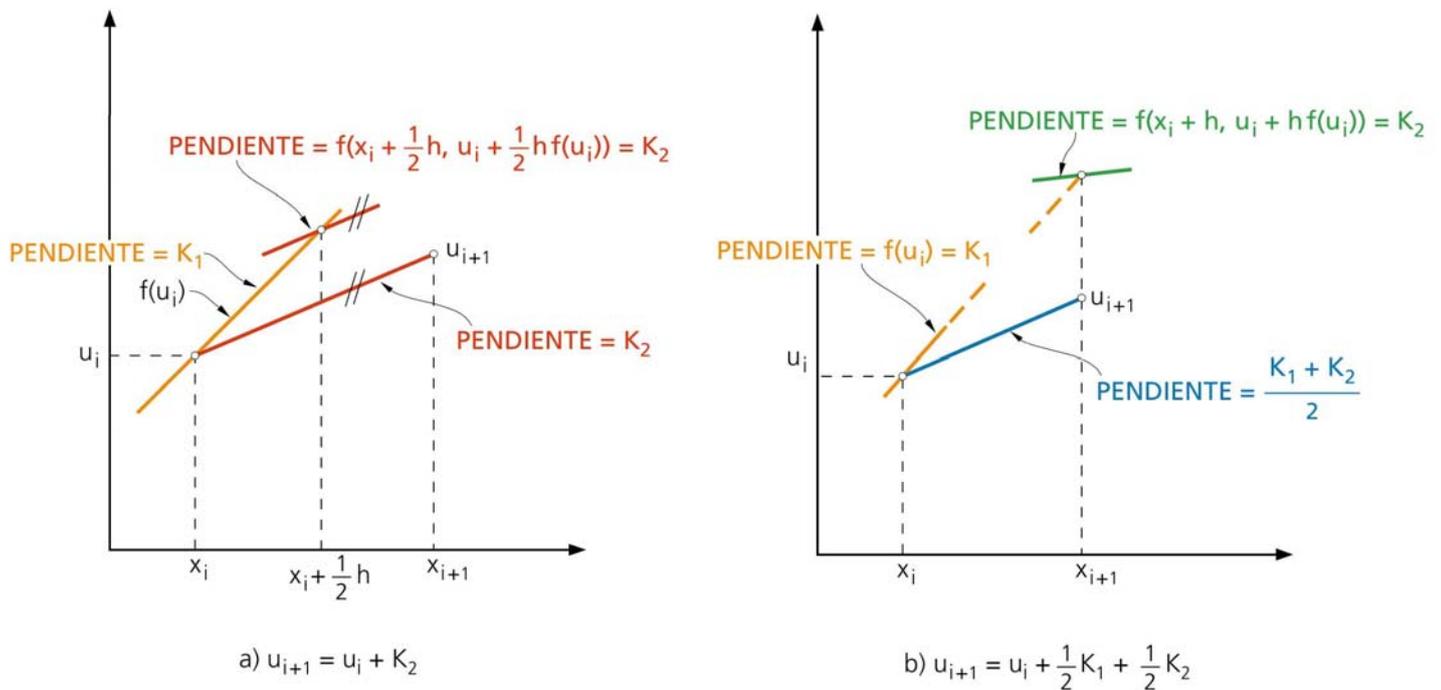


Figura 3.2. Interpretaciones gráficas de algoritmos de RK de orden 2, casos a) y b). [Reelaborado a partir de Davis, 1990].

Los algoritmos de RK más utilizados son los orden 2 y los de orden 4. Los primeros presentan bastante simplicidad de uso y suficiente exactitud para problemas no excesivamente complicados, los segundos presentan muy buena exactitud para problemas exigentes y no tienen excesivo coste de cálculo. En los algoritmos de orden superior a 4 no se compensa el aumento de exactitud ya que el cálculo se complica y aumenta el coste de computación.

La forma general de los algoritmos de RK de orden 4 es:

$$u_{i+1} = u_i + w_1 k_1 + w_2 k_2 + w_3 k_3 + w_4 k_4$$

$$u_0 = y_0$$

$$\text{siendo} \begin{cases} k_1 = h * f(x_i, u_i) \\ k_2 = h * f(x_i + c_2 h, u_i + a_{21} k_1) \\ k_3 = h * f(x_i + c_3 h, u_i + a_{31} k_1 + a_{32} k_2) \\ k_4 = h * f(x_i + c_4 h, u_i + a_{41} k_1 + a_{42} k_2 + a_{43} k_3) \end{cases}$$

Emparejando estos coeficientes con los correspondientes a la expansión de Taylor (y tras operaciones matemáticas) se obtiene un sistema de 11 ecuaciones con 13 incógnitas, se hace necesario dar valores arbitrarios a dos de las incógnitas. De los muchos algoritmos de RK de orden 4 a los que se puede llegar por combinación de valores arbitrarios se presentan los dos de mayor uso en algoritmos comerciales:

Algoritmo 1:

$$u_{i+1} = u_i + \frac{1}{6}(k_1 + 2k_2 + 2k_3 + k_4) + \mathcal{O}(h^5)$$

$$u_0 = y_0$$

$$\text{siendo} \begin{cases} k_1 = h * f(x_i, u_i) \\ k_2 = h * f\left(x_i + \frac{1}{2} h, u_i + \frac{1}{2} k_1\right) \\ k_3 = h * f\left(x_i + \frac{1}{2} h, u_i + \frac{1}{2} k_2\right) \\ k_4 = h * f(x_i + h, u_i + k_3) \end{cases}$$

Algoritmo 2: Runge-Kutta-Gill

$$u_{i+1} = u_i + \frac{1}{6}(k_1 + k_4) + \frac{1}{3}(bk_2 + dk_3)$$

$$u_0 = y_0$$

$$\text{siendo } \begin{cases} k_1 = h * f(x_i, u_i) \\ k_2 = h * f\left(x_i + \frac{1}{2}h, u_i + \frac{1}{2}k_1\right) \\ k_3 = h * f\left(x_i + \frac{1}{2}h, u_i + ak_1 + bk_2\right) \\ k_4 = h * f(x_i + h, u_i + ck_2 + dk_3) \\ a = \frac{\sqrt{2}-1}{2}, b = \frac{2-\sqrt{2}}{2}, c = -\frac{\sqrt{2}}{2}, d = 1 + \frac{\sqrt{2}}{2} \end{cases}$$

El uso de las fórmulas de RK mejora la exactitud del algoritmo pero la estabilidad también se ve limitada a un rango de valores:

$$\text{RK 2º orden : } -2,0 \leq \lambda h \leq 0 \quad (\lambda \text{ real}) \quad \text{RK 4º orden: } -2,8 \leq \lambda h \leq 0 \quad (\lambda \text{ real})$$

Se pueden encontrar ejemplos de algoritmos de RK de otros órdenes de exactitud en diferentes textos y librerías matemáticas.

## 2.2. MÉTODOS EXPLÍCITOS DE ETAPAS MÚLTIPLES.

Todos los métodos vistos hasta ahora pertenecen al grupo denominado de **paso simple** o de **un sólo paso** porque para el cálculo de cada nodo se utiliza únicamente información del nodo previo. Sin embargo, una vez calculados varios nodos sería posible utilizar la información de más de un nodo previo para obtener el valor de la aproximación en el siguiente nodo. Como ejemplo se describe el **método de los cuatro pasos de Adams-Bashforth-Moulton**:

En este algoritmo se necesitan los valores de  $u_{i-3}, u_{i-2}, u_{i-1}, u_i$  para calcular  $u_{i+1}$ ,

Al comienzo necesitaremos conocer por adelantado los primeros cuatro puntos  $u_0, u_1, u_2, u_3$  para poder generar la sucesión  $\{(x_i, u_i): i \geq 4\}$ . Para ello se puede empezar aplicando un método de orden similar (por ejemplo RKG) para obtener las cuatro primeras aproximaciones para pasar después a aplicar el método ABM.

Este método además es un método *predictor-corrector*. En este tipo de métodos la aproximación en cada nodo se realiza en dos etapas, en la primera se obtiene una aproximación para el nodo denominada valor *predictor*,  $p_{i+1}$ , que se utiliza en la segunda para obtener la aproximación definitiva,  $u_{i+1}$ , denominada valor *corrector*. El par de algoritmos de este método es:

$$p_{i+1} = u_i + \frac{h}{24} [-9f_{i-3} + 37f_{i-2} - 59f_{i-1} + 55f_i], \quad \mathcal{O}(h^5)$$

$$u_{i+1} = u_i + \frac{h}{24} [f_{i-2} - 5f_{i-1} + 19f_i + 9f(x_{i+1}, p_{i+1})], \quad \mathcal{O}(h^5)$$

El método ABM se clasifica entonces como:

- **De etapas múltiples:** porque emplea información de más de un nodo anterior al que buscamos.
- **Predictor-corrector:** porque para cada nodo utiliza dos etapas, en vez de una. Con la primera se obtiene una aproximación que se utiliza en la segunda para obtener la aproximación definitiva.

Seguimos dentro de los métodos explícitos que tienen limitaciones en el rango de estabilidad

### 3. MÉTODOS IMPLÍCITOS APLICABLES A ODE-IVP

Los métodos implícitos evalúan la función en nodos posteriores al nodo que se está calculando, por lo tanto se complica la resolución de las ecuaciones algebraicas a las que dan lugar los algoritmos, en función del tipo de ecuación diferencial de partida (lineal o no lineal) las ecuaciones algebraicas que tendremos que resolver serán a su vez lineales o no lineales y podrán requerir de métodos iterativos para su resolución lo que aumentará mucho el coste de cálculo en cada nodo del intervalo de integración. A cambio de este aumento en la dificultad de cálculo estos métodos aumentan la ESTABILIDAD frente a los métodos explícitos.

#### 3.1. Algoritmo de Euler implícito

Partiendo de la expresión de Taylor truncada en el segundo término:

$$y_i = y_{i+1} - (x_{i+1} - x_i) y'_{i+1} + \frac{(x_{i+1} - x_i)^2}{2!} y''(\varepsilon_i), \quad \text{siendo } x_i \leq \varepsilon_i \leq x_{i+1}$$

llegamos a: 
$$u_{i+1} = u_i + h * f(x_{i+1}, u_{i+1}) + \mathcal{O}(h^2)$$

Esta aproximación se conoce como fórmula de Euler implícita. Es implícita porque para determinar el valor de  $u_{i+1}$  es necesario evaluar la función en el punto  $(x_{i+1}, u_{i+1})$ . El método implícito no aumenta la exactitud del método de Euler ya que el error sigue siendo proporcional a  $h^2$ , pero aumenta la estabilidad del método ya que un método implícito es incondicionalmente estable.

#### 3.2. Algoritmo del trapecio (método de Euler modificado)

Otro método implícito utilizado habitualmente es el denominado método del trapecio. El algoritmo de este método es:

$$\begin{cases} u_{i+1} = u_i + \frac{h}{2} [f_i + f_{i+1}] + \mathcal{O}(h^3) \\ u_0 = y_0 \end{cases}$$

Este método, además de ser estable (por implícito) presenta orden de exactitud 2 porque se genera a partir de la diferencia entre las expansiones de Taylor correspondientes a  $y_i$  e  $y_{i+1}$  truncadas en el término  $\mathcal{O}(h^3)$ .

#### 4. ESTRATEGIAS DE CONTROL DEL ERROR EN LAS APROXIMACIONES

Hasta ahora hemos ilustrado los métodos numéricos aplicándolos a problemas que tienen una solución analítica por lo que sus errores son reconocibles y cuantificables.

**En un problema práctico no se conoce la solución analítica por lo que no pueden hacerse comparaciones para determinar el error de cálculo. Deben construirse estrategias alternativas para calcular el error.**

Un método intuitivo para la estimación del error local puede ser el cálculo de la diferencia entre  $u_{i+1}^*$  y  $u_{i+1}$  donde  $u_{i+1}$  se calcula usando un tamaño de paso  $h$  y  $u_{i+1}^*$  un tamaño  $h/2$ . Como la exactitud del método numérico depende del tamaño de paso,  $u_{i+1}^*$  será una mejor aproximación de  $y(x_i)$ , por tanto:

$$\left| y_{i+1} - u_{i+1}^* \right| < \left| y_{i+1} - u_{i+1} \right| \text{ y por lo tanto } \boxed{e_{i+1} = \left( y_{i+1} \approx u_{i+1}^* - u_{i+1} \right)}$$

Para las fórmulas de RK, el uso del procedimiento de una etapa más dos medias etapas puede resultar muy costoso, por lo tanto se ha desarrollado un mejor procedimiento, **el método de Fehlberg**, el cual calcula para cada paso  $u_{i+1}^*$  y  $u_{i+1}$  con el mismo tamaño de paso pero usando una fórmula de RK de orden de exactitud más elevado para calcular  $u_{i+1}^*$  que la que se usa para calcular  $u_{i+1}$ . De esta manera  $u_{i+1}^*$  tiene mayor exactitud que  $u_{i+1}$ . El par de fórmulas del método RKF de cuarto orden es:

$$u_{i+1} = u_i + \left[ \frac{25}{216} k_1 + \frac{1408}{2565} k_3 + \frac{2197}{4104} k_4 - \frac{1}{5} k_5 \right] \quad e_{i+1} = \mathcal{O}(h^5)$$

$$u_{i+1}^* = u_i + \left[ \frac{16}{135} k_1 + \frac{6656}{12825} k_3 + \frac{28561}{56430} k_4 - \frac{9}{50} k_5 + \frac{2}{55} k_6 \right] \quad e_{i+1} = \mathcal{O}(h^6)$$

$$k_1 = h * f(x_i, u_i)$$

$$k_2 = h * f\left(x_i + \frac{1}{4}h, u_i + \frac{1}{4}k_1\right)$$

$$k_3 = h * f\left(x_i + \frac{3}{8}h, u_i + \frac{3}{32}k_1 + \frac{9}{32}k_2\right)$$

Donde:

$$k_4 = h * f\left(x_i + \frac{12}{13}h, u_i + \frac{1932}{2197}k_1 + \frac{7200}{2197}k_2 + \frac{7296}{2197}k_3\right)$$

$$k_5 = h * f\left(x_i + h, u_i + \frac{439}{216}k_1 - 8k_2 + \frac{3680}{513}k_3 - \frac{845}{4104}k_4\right)$$

$$k_6 = h * f\left(x_i + \frac{1}{2}h, u_i - \frac{8}{27}k_1 + 2k_2 - \frac{3544}{2565}k_3 + \frac{1859}{4104}k_4 - \frac{11}{40}k_5\right)$$

Una primera mirada a la aproximación RKF lo muestra como complicado pero se puede programar de una manera muy sencilla. Utilizando esta aproximación, en cada nodo se realizarán 6 evaluaciones de las funciones mientras que, por ejemplo, con el método RKG, que es del mismo orden de exactitud y controlando el error en cada paso, se necesitan 11 evaluaciones en cada etapa.

La forma de utilizar la aproximación de RKF en un algoritmo consiste en calcular para cada nodo las aproximaciones  $u_{i+1}^*$  y  $u_{i+1}$  y comprobar si la diferencia es mayor o menor que la tolerancia establecida. Si el error es menor se pasa al siguiente nodo, si el error es mayor volveremos a calcular las aproximaciones en ese nodo con un tamaño de paso menor.

#### Estrategias de control del error:

- Trabajar con diferentes tamaños de paso y el mismo orden de exactitud en el método. Se tomará como control del error la aproximación realizada con el menor tamaño de paso.
- Trabajar con métodos de diferente exactitud utilizando el mismo tamaño de paso. Se tomará como control del error la aproximación realizada con el mayor orden de exactitud.

## 5. ESTABILIDAD Y EXACTITUD EN LOS MÉTODOS NUMÉRICOS IVP

La aplicación práctica de los algoritmos para problemas IVP implica una serie de errores:

1. Error de redondeo: se produce cuando  $u_0 \neq y_0$  ( $e_0$ ). Este error se da cuando el ordenador o el lenguaje de programación utilizado redondea las cifras decimales.
2. Error de truncamiento local: error que se produce en cada paso supuesto que el anterior es exacto ( $e_i$ ). Es un error intrínseco al método numérico y depende de:
  - a) orden de precisión del método.
  - b) Tamaño de paso utilizado ( $h$ ).
3. Error global: Diferencia entre la solución exacta y la calculada con el método para el nodo correspondiente ( $\varepsilon_i$ ).

La figura 3.3. muestra gráficamente la influencia de los diferentes tipos de error en un método numérico.

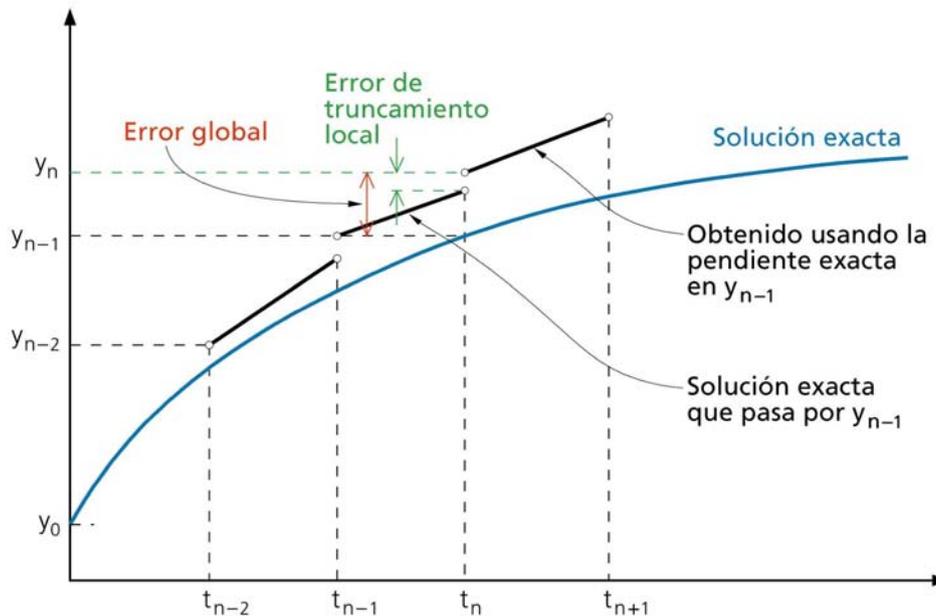


Figura 3.3. Efecto de los errores local y global en la resolución de un problema IVP mediante un algoritmo numérico. [Reelaborado a partir de Rice y Do, 2012]

**El OBJETIVO a cumplir en cualquier problema es que el error global no aumente en cada paso. Si esto se cumple el algoritmo se considera ESTABLE.**

Las condiciones para que un problema sea estable son diferentes en cada caso pero se puede hacer un estudio general para el caso más simple: problema lineal resuelto mediante el método de Euler:

**Ejemplo 3.4:**

Consideremos el problema  $\begin{cases} y' = \lambda y \\ y(x_0) = y_0 \end{cases}$  donde  $\lambda$  es un número real.

Este problema tiene solución analítica (exacta):  $y = y_0 * e^{\lambda x}$ , para un punto  $x_{i+1} \Rightarrow y_{i+1} = y_0 * e^{\lambda x_{i+1}} = y_0 * e^{(i+1)h\lambda}$ .

Aplicando el algoritmo de Euler podemos obtener una aproximación numérica de la forma:  
 $u_{i+1} = u_i + (\lambda * h * u_i) = u_i(1 + \lambda h) = u_0(1 + \lambda h)^{i+1}$

Comparando ambas soluciones se observa que al aplicar el algoritmo de Euler lo que se hace es aproximar:

$$y_0 * e^{(i+1)h\lambda} = u_0(1 + \lambda h)^{i+1}, \text{ como } y_0 = u_0 \Rightarrow \boxed{e^{h\lambda} = (1 + h\lambda)}$$

Esta aproximación es el error inherente al método. No es posible trabajar sin este error al aplicar el método de Euler.

Supongamos además que  $y_0 \neq u_0$ , entonces:  $u_{i+1} = (1 + h\lambda)^{i+1} (y_0 - e_0)$ , el error global será entonces:

$$\varepsilon_{i+1} = y_{i+1} - u_{i+1} = y_0 (e^{h\lambda})^{i+1} - (1 + h\lambda)^{i+1} (y_0 - e_0), \text{ reordenando:}$$

$$\boxed{\varepsilon_{i+1} = \underbrace{[e^{(i+1)h\lambda} - (1 + h\lambda)^{i+1}]}_A y_0 + \underbrace{(1 + h\lambda)^{i+1}}_B e_0}$$

A = Error que resulta de la aproximación del método de Euler.  
 B = Efecto de propagación del error inicial.

Caso 1: Si  $|1 + h\lambda| > 1$  el término B crecerá con cada término  $i > (i-1)$  porque está elevado a  $(i+1)$  y sin importar cual sea el valor de  $e_0$  se convertirá en el término dominante al llegar al nodo  $(i+1)$ .

Caso 2: Si  $|1 + h\lambda| \leq 1$  El término B disminuirá con cada nodo  $i > (i-1)$  y por tanto el término B será menos importante que A al llegar al nodo  $(i+1)$ .

Por lo tanto para mantener unidos los efectos de propagación de errores se requiere que  $|1 + h\lambda| \leq 1$  para el algoritmo de Euler.

Se define la REGIÓN DE ESTABILIDAD ABSOLUTA por el conjunto de tamaños de paso,  $h$  (real, no negativo), y los valores de  $\lambda$  para los cuales una perturbación en un simple valor  $u_i$  producirá en los siguientes nodos un cambio que no aumenta paso a paso.

Si  $\lambda$  es real la región de estabilidad (para el algoritmo de Euler) será:  $-2 \leq h\lambda \leq 0$

Dentro de esta región si  $-2 \leq h\lambda \leq -1$  el error global oscilará de signo.

El estudio anterior es aplicable a ecuaciones lineales en las que  $\lambda$  es constante en el intervalo de integración. Para poder aplicar el mismo estudio en una ecuación no lineal el habría que linealizar la ecuación en cada nodo.

En resumen, el algoritmo de Euler presenta dos problemas que pueden mejorarse:

- a) Exactitud
- b) Estabilidad.

Implícito en estas categorías está el costo del cálculo para mejorar la estabilidad y la exactitud. Con este método se requieren tamaños de paso muy pequeños lo que lleva a un gran número de evaluaciones de la función incrementando así el coste del cálculo.

Las estrategias para mejorar estos problemas son:

- a) aumentar el orden de exactitud del método manteniendo el carácter explícito del mismo.
- b) Utilizar un método implícito.

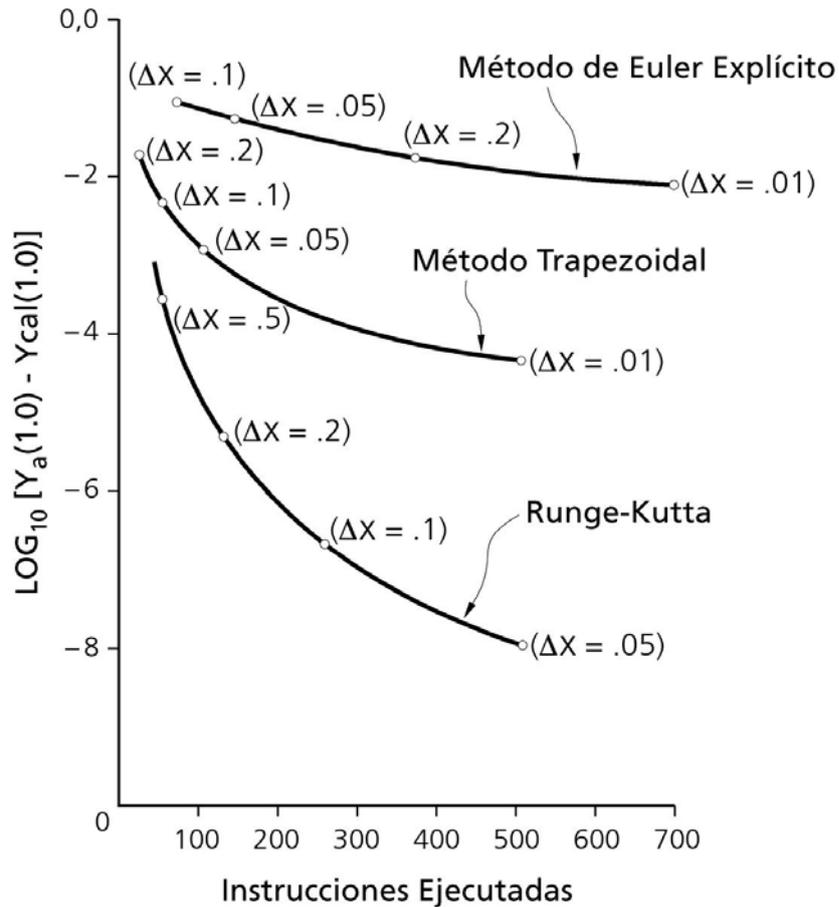
En la Tabla 3.1. se presentan las regiones de estabilidad de algunos algoritmos numéricos para problemas IVP. Los límites recogidos en esta Tabla solo son válidos para ecuaciones lineales con  $\lambda$  reales, pero indican una referencia de las regiones de estabilidad para cada método en cualquier tipo de problema.

**Tabla 3.1. Regiones de estabilidad para algunos algoritmos de ODE-IVP**

Método	Orden de aproximación	Región de estabilidad
Euler explícito	1	$-2 \leq \lambda h \leq 0$
Euler implícito	1	No límite
Euler modificado	2	No límite
Runge-Kutta 2º orden	2	$-2 \leq \lambda h \leq 0$
Runge-Kutta 3º orden	3	$-2,5 \leq \lambda h \leq 0$
Runge-Kutta 4º orden	4	$-2,78 \leq \lambda h \leq 0$
Runge-Kutta 5º orden	5	$-5,7 \leq \lambda h \leq 0$
Adams-Moulton	4	$-1,28 \leq \lambda h \leq 0$

**Ejemplo 3.5:**

La Figura 3.4. permite comparar i) la influencia del orden de exactitud y ii) del tamaño de paso de la integración de los algoritmos IVP aplicados a un problema ejemplo: 
$$\begin{cases} y' = x^2 y \\ y(0) = 1 \end{cases}$$



**Figura 3.4. Comparación de los métodos Euler explícito, Runge-Kutta y Trapezoidal. [Reelaborado a partir de Riggs, 1994].**

En la figura se muestra la relación entre el número de operaciones realizadas y la diferencia entre la solución analítica y la respectiva aproximación alcanzada para el problema IVP.

## 6. BIBLIOGRAFÍA RELACIONADA

**Davis, M.E.;** *Métodos y Modelos Numéricos para Ingenieros Químicos.* CAPÍTULO 1. Compañía Editorial Continental de C.V. México, México D.F. 1990.

**Riggs, J.B.;** *An Introduction to Numerical Methods for Chemical Engineers.* CAPÍTULO 4. Texas Tech University Press, Lubbock, Texas. 1994.

**Rice, R.G., Do, D.D.;** *Applied Mathematics and Modeling for Chemical Engineers, 2<sup>nd</sup> Edition.* CAPÍTULO 7. John Wiley & Sons, Hoboken, New Jersey, USA. 2012.

**Zill, D.G.** *Ecuaciones Diferenciales con Aplicaciones de Modelado.* TeEd Int. Thomson 6ªEd. 1997.

**Butcher, J.C.** *Numerical Methods for Ordinary Differential Equations.* Wiley, Chichester, UK. 2003.

**Griffiths, D.F., Higham, D.J.** *Numerical Methods for Ordinary Differential Equations.* Springer, Berlín, Alemania. 2010.