



Departamento de Automática

**Métodos numéricos para la
solución de ecuaciones diferenciales
ordinarias (*EDO*)**

Prof. Víctor M. Alfaro

Febrero de 2002

Rev. Diciembre de 2005

TABLA DE CONTENIDO

1. Introducción.....	1
1.1 Planteamiento del problema.....	1
2. Métodos del tipo Runge-Kutta.....	3
2.1 Método de Euler (Runge-Kutta de 1er orden).....	3
2.2 Método de Euler modificado (Runge-Kutta de 2° orden).....	5
2.3 Método de Euler-Cauchy modificado (Runge-Kutta de 2° orden).....	6
2.4 Método de la regla de Simpson (Runge-Kutta de 3er orden).....	7
2.5 Método de Heun (Runge-Kutta de 3er orden).....	7
2.6 Runge-Kutta de 4° orden (“clásico”).....	8
2.7 Runge-Kutta-Simpson de 4° orden.....	9
2.8 Métodos Runge-Kutta de paso variable.....	9
2.8.1 Runge-Kutta-Fehlberg de 2° orden	10
2.8.2 Runge-Kutta-Fehlberg de 4° orden	11
2.9 Error por truncamiento y error por redondeo.....	12
2.10 Sistemas rígidos.....	12
2.11 Efecto del paso de integración sobre la exactitud de la solución.....	13
3. Métodos de integración numérica.....	15
3.1 Fórmulas directas, Métodos de Adams-Bashforth.....	15
3.1.1 Método de segundo orden	16
3.1.2 Método de cuarto orden	16
3.2 Fórmulas implícitas, Métodos de Adams-Moulton.....	16
3.2.1 Método de segundo orden	16
3.2.2 Método de cuarto orden	16
3.3 Métodos predictor - corrector.....	16
3.3.1 Método trapezoidal modificado (2° orden)	17
3.3.2 Método de Adams-Bashforth-Moulton de 4° orden	17

3.3.3 Método de Milne de 4° orden	18
3.3.4 Método de Milne de 6° orden	18
3.4 Algoritmo de solución.....	18
4. Ejemplos del uso de algunos métodos.....	20
4.1 Método de la regla de Simpson en MATLAB.....	20
4.2 Método de Adams-Bashforth de 2° orden en Scilab.....	21
4.3 Solución de una ecuación diferencial de segundo orden.....	23
4.4 “ODE Solvers” en MATLAB y Simulink.....	25
4.4.1 Métodos de paso variable	25
4.4.2 Métodos de paso fijo (solo en Simulink)	26
4.4.3 Ejemplo	26
4.5 “ODE Solvers” en Scilab.....	27
4.6 Métodos de integración en VisSim.....	29
5. Comparación de los métodos de solución de EDO.....	30
Bibliografía.....	32

ESCUELA DE INGENIERIA ELECTRICA
UNIVERSIDAD DE COSTA RICA

1. Introducción

El estudio de los procesos dinámicos y sus sistemas de control, debe iniciarse con la obtención de una representación matemática de las relaciones existentes entre las diferentes variables involucradas en el proceso a controlar, a la que usualmente se denomina *modelo* del sistema.

El proceso de modelado de un sistema dinámico, puede llevar a la obtención de una representación para el mismo por medio de una ecuación diferencial de orden alto, o por un conjunto de ecuaciones diferenciales de primer orden no lineales, cuya solución se debe obtener para conocer la respuesta temporal del sistema, a partir un conjunto de condiciones iniciales y una entrada dada.

La solución analítica de una ecuación diferencial lineal puede ser fácil, de varias ya presenta dificultades y de muchas es prácticamente imposible. Si las ecuaciones diferenciales son no lineales, el resolver una sola es muy difícil y varias o muchas es imposible por medios analíticos.

Como es normal que el modelo obtenido para el sistema que se desea analizar, esté constituido por varias ecuaciones diferenciales no lineales, este solamente puede resolverse con la ayuda de un programa de simulación digital.

Para el desarrollo de un programa de simulación de sistemas dinámicos, es necesario entonces contar con un método de solución de ecuaciones diferenciales.

Se presentarán adelante en forma breve, algunos de los métodos numéricos de solución de *ecuaciones diferenciales ordinarias (EDO)*, más empleados en la simulación digital de los sistemas dinámicos.

1.1 Planteamiento del problema

Se requiere solucionar la ecuación diferencial de primer orden

$$\frac{dy}{dt} = f(y, t), \quad y(t_0) = y_0 \quad (1)$$

para encontrar y en una secuencia de valores de la variable independiente $t \in \{t_i\}$ dentro de un intervalo de solución $[t_0, t_f]$, donde $f(y, t)$ es una función no lineal cualquiera.

La obtención de la solución de (1) es conocida como el *problema del valor inicial* en la solución de ecuaciones diferenciales y para esto, se dispone de dos tipos de métodos de solución:

1. Métodos en los cuales $f(y, t)$ será evaluada solamente en los puntos (y_i, t_i) , donde y_i es el valor de y en $t = t_i$ y que se denominan **Métodos de integración numérica**
2. Métodos en los cuales $f(y, t)$ será evaluada además, en puntos distintos de (y_i, t_i) y que se denominarán **Métodos del tipo Runge-Kutta**.

Si se define a $Y(t)$ como la solución exacta de (1) y a $y(t)$ como la solución calculada, entonces

$$Y_n = Y(t_n), \quad \dot{Y}_n = \left(\frac{dY}{dt} \right)_{t=t_n} = f(Y_n, t_n) \quad (2)$$

$$y_n = y(t_n), \quad \dot{y}_n = f(y_n, t_n) \quad (3)$$

$$\Delta t = t_{n+1} - t_n \quad (4)$$

Como Y es la solución verdadera, $f(Y_n, t_n)$ es igual a $dY/dt|_{t=t_n}$, sin embargo $y(t)$ solamente existe en los instantes $n = 1, 2, 3, \dots$

El intervalo de tiempo entre dos instantes consecutivos de la solución, denominado usualmente *paso de integración* Δt , puede permanecer constante sobre un determinado número de intervalos de la solución, o variarse cuando consideraciones de error lo hagan deseable.

En la obtención de los métodos numéricos para la solución de de las ecuaciones es importante considerar entonces:

1. cuanto error se comete en cada paso del cálculo y como afecta este los pasos siguientes, esto es, cómo se propaga el error
2. la habilidad del método para estimar el error en una etapa de cálculo, en función de los resultados obtenidos
3. la iniciación del método (se conoce la condición inicial y_0 pero como se verá, algunos métodos numéricos requieren conocer además, los valores de y en más de un punto anterior para calcular el siguiente) y
4. la velocidad del método.

En la presentación siguiente de los diferentes métodos de solución de ecuaciones diferenciales, se considerará la solución de una sola ecuación diferencial no lineal de primer orden, sin embargo todos ellos son fácilmente extensibles al caso de un conjunto de ecuaciones diferenciales de primer orden no lineales simultáneas, considerando a todas las variables y ecuaciones como vectores. Si el modelo está representado por una ecuación diferencial de orden alto, es necesario convertirla primero en un conjunto de ecuaciones diferenciales de primer orden simultáneas para su solución.

UNIVERSIDAD DE COSTA RICA

2. Métodos del tipo Runge-Kutta

La base de todos los métodos del tipo Runge-Kutta es expresar la diferencia entre los valores de y en t_{n+1} y t_n como

$$y_{n+1} - y_n = \sum_{i=1}^m \gamma_i k_i \quad (5)$$

donde los γ_i son constantes y los

$$k_i = \Delta t f \left(y_n + \sum_{j=1}^{i-1} \beta_{ij} k_j, t_n + \alpha_i \Delta t \right) \quad (6)$$

con $\alpha_1 = 0$. Dadas las constantes γ_i , α_i y β_{ij} , la solución es directa.

Los métodos del tipo Runge-Kutta se caracterizan por

1. ser auto-iniciables
2. requerir solamente información del punto n para calcular la solución en el punto $n+1$
3. evaluar en cada iteración la función (derivada) tantas veces como el orden del método, y
4. por no poseer forma de estimar en error cometido, a menos que se utilicen simultáneamente dos métodos de distinto orden.

En los métodos del tipo Runge-Kutta el número de veces que es necesario evaluar la ecuación diferencial es igual al orden del método, lo cual repercute directamente sobre su velocidad. Sin embargo a medida que aumenta el orden del método, este tiene una mayor exactitud para un paso de integración dado, por lo que será posible utilizar un paso de integración mayor para un grado de exactitud deseado.

2.1 Método de Euler (Runge-Kutta de 1^{er} orden)

“Integración rectangular”

Se puede calcular el valor de la integral de y en el instante $t_{n+1} = t_n + \Delta t$, mediante la expansión de la ecuación (1) en una serie de Taylor

$$y_{n+1} = y_n + \Delta t \frac{dy}{dt} + \frac{\Delta t^2}{2!} \frac{d^2 y}{dt^2} + \frac{\Delta t^3}{3!} \frac{d^3 y}{dt^3} + \dots + \frac{\Delta t^p}{p!} \frac{d^p y}{dt^p} + \dots \quad (7)$$

Sustituyendo (1) en (7) se obtiene que

$$y_{n+1} = y_n + \Delta t f(y_n, t_n) + \frac{\Delta t^2}{2!} \frac{df}{dt} + \frac{\Delta t^3}{3!} \frac{d^2 f}{dt^2} + \dots + \frac{\Delta t^p}{p!} \frac{d^{p-1} f}{dt^{p-1}} + \dots \quad (8)$$

Si se reduce sustancialmente el valor del paso de integración Δt , se pueden despreciar los términos con Δt^2 y superiores con lo que se obtiene que

$$y_{n+1} = y_n + \Delta t f(y_n, t_n) \quad (9)$$

el cual es el conocido método de *integración rectangular* o *método de Euler*.

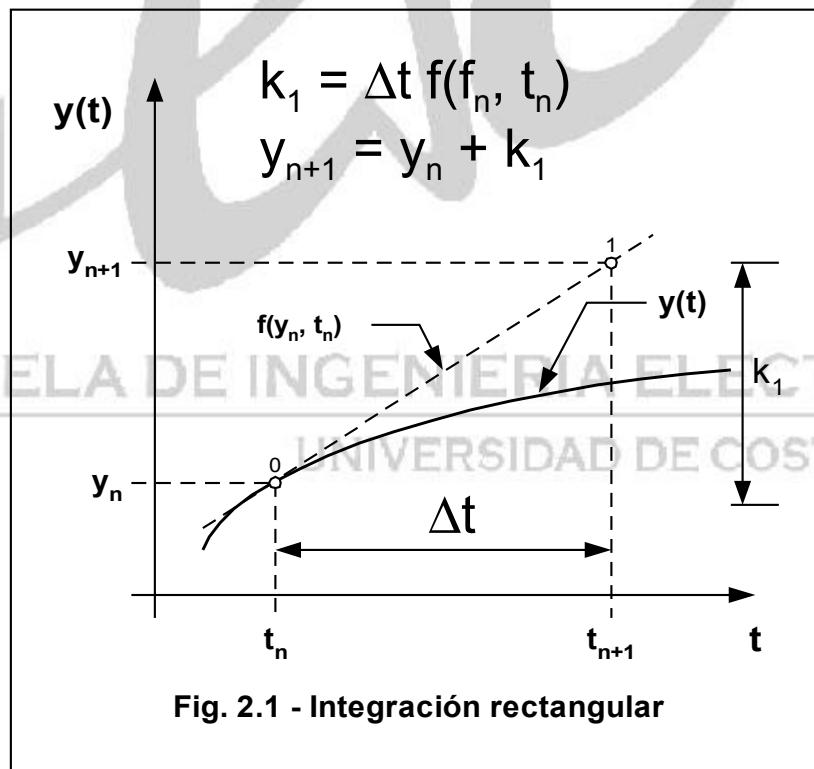
La ecuación (9) se puede reescribir para expresarla en la forma general de los métodos del tipo Runge-Kutta dada por (5) y (6) quedando esta entonces como

$$k_1 = \Delta t f(y_n, t_n) \quad (10)$$

$$y_{n+1} = y_n + k_1$$

Este método se dirá que es de primer orden por haberse truncado los términos que contenían más allá de la primera potencia de Δt . La omisión de estos términos dará el error por truncamiento inherente del método, el cual se dirá en este caso que es $O(\Delta t^2)$ o *del orden de Δt^2* .

El método de Euler estima el valor de la solución en el instante $n + 1$ a partir de la extrapolación del valor de la ecuación diferencial en el instante n , tangente de la solución, por un paso de integración Δt como se muestra en la Figura 2.1.



2.2 Método de Euler modificado (Runge-Kutta de 2° orden)

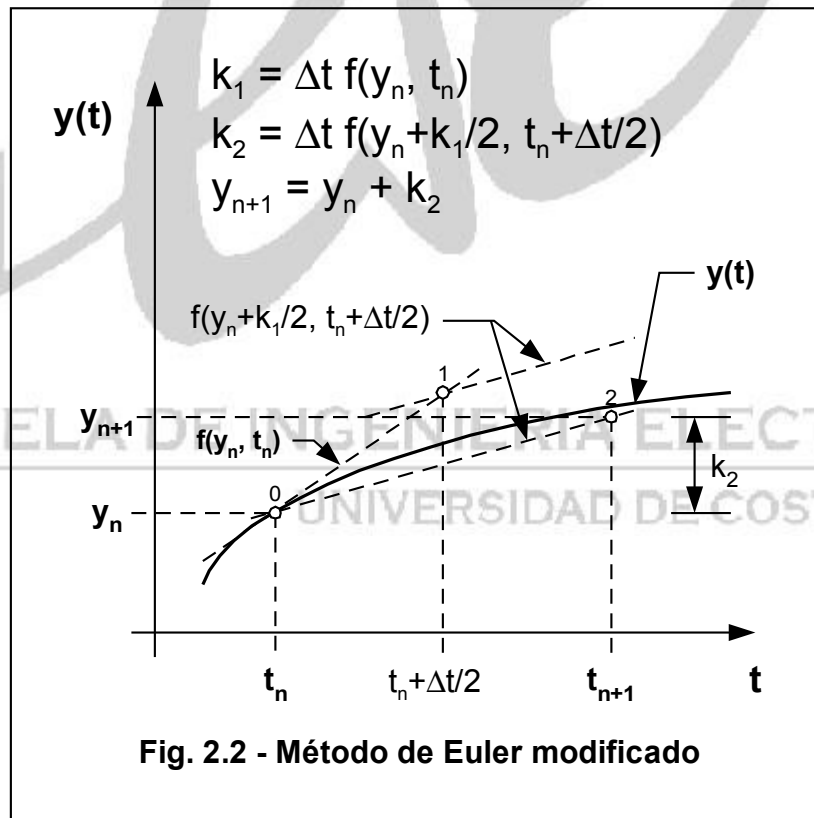
El método de Euler es un método de un paso y el más simple de los métodos del tipo Runge-Kutta. Un método de segundo orden o de dos pasos, está dado por las siguientes ecuaciones

$$\begin{aligned} k_1 &= \Delta t f(y_n, t_n) \\ k_2 &= \Delta t f\left(y_n + \frac{1}{2}k_1, t_n + \frac{1}{2}\Delta t\right) \end{aligned} \quad (11)$$

$$y_{n+1} = y_n + k_2$$

Este método coincide con la serie de Taylor (8) hasta el término Δt^2 por lo que su error por truncamiento será $O(\Delta t^3)$.

El *método de Euler modificado* utiliza el valor de la ecuación diferencial al centro del paso de integración, extendiéndolo a todo su ancho según se aprecia en la Figura 2.2.



2.3 Método de Euler-Cauchy modificado (Runge-Kutta de 2º orden)

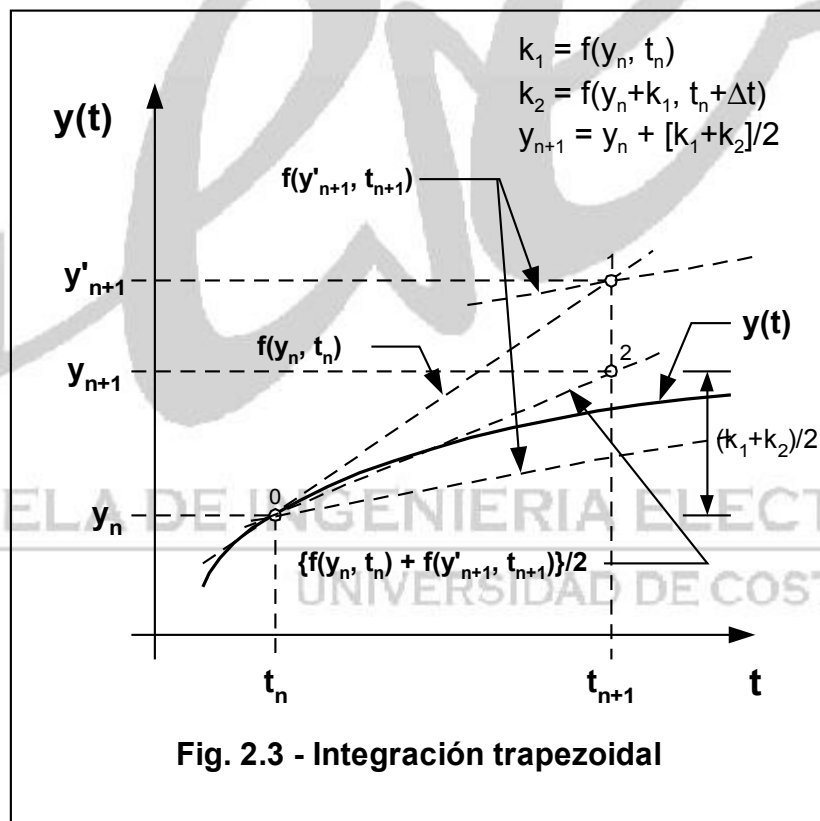
“Integración trapezoidal”

Este es otro método de segundo orden y sus ecuaciones son

$$\begin{aligned} k_1 &= \Delta t f(y_n, t_n) \\ k_2 &= \Delta t f(y_n + k_1, t_n + \Delta t) \end{aligned} \quad (12)$$

$$y_{n+1} = y_n + \frac{1}{2}[k_1 + k_2]$$

El *método de integración trapezoidal* utiliza un promedio del valor de la ecuación diferencial al inicio (n) y final ($n+1$) del paso de integración, como se indica en la Figura 2.3.



2.4 Método de la regla de Simpson (Runge-Kutta de 3^{er} orden)

Este es un método de tres pasos dado por

$$\begin{aligned}k_1 &= \Delta t f(y_n, t_n) \\k_2 &= \Delta t f\left(y_n + \frac{1}{2}k_1, t_n + \frac{1}{2}\Delta t\right) \\k_3 &= \Delta t f(y_n - k_1 + 2k_2, t_n + \Delta t)\end{aligned}\tag{13}$$

$$y_{n+1} = y_n + \frac{1}{6}[k_1 + 4k_2 + k_3]$$

con un error por truncamiento $O(\Delta t^4)$.

2.5 Método de Heun (Runge-Kutta de 3^{er} orden)

Este es otro método de tres pasos y está dado por

$$\begin{aligned}k_1 &= \Delta t f(y_n, t_n) \\k_2 &= \Delta t f\left(y_n + \frac{1}{3}k_1, t_n + \frac{1}{3}\Delta t\right) \\k_3 &= \Delta t f\left(y_n + \frac{2}{3}k_2, t_n + \frac{2}{3}\Delta t\right)\end{aligned}\tag{14}$$

$$y_{n+1} = y_n + \frac{1}{4}[k_1 + 3k_3]$$

con un error por truncamiento $O(\Delta t^4)$

2.6 Runge-Kutta de 4° orden (“clásico”)

El *método de Runge-Kutta* de cuarto orden, es el utilizado con más frecuencia en la simulación de los sistemas dinámicos y sus ecuaciones son

$$\begin{aligned}
 k_1 &= \Delta t f(y_n, t_n) \\
 k_2 &= \Delta t f(y_n + \frac{1}{2} k_1, t_n + \frac{1}{2} \Delta t) \\
 k_3 &= \Delta t f(y_n + \frac{1}{2} k_2, t_n + \frac{1}{2} \Delta t) \\
 k_4 &= \Delta t f(y_n + k_3, t_n + \Delta t) \\
 y_{n+1} &= y_n + \frac{1}{6} [k_1 + 2k_2 + 2k_3 + k_4]
 \end{aligned} \tag{15}$$

El error por truncamiento de este método es $O(\Delta t^5)$, siendo además mucho más estable que los anteriores.

Existen varios métodos Runge-Kutta de 4° orden. Además del anterior, cuyos coeficientes se atribuyen a Kutta, existe el Runge-Kutta-Gill en que se minimiza la memoria utilizada, el Runge-Kutta-Ralston en el que se minimiza el error por truncamiento, el Runge-Kutta-Merson que es una extensión para hacerlo de paso variable, esto a costa de una evaluación adicional de las ecuaciones del sistema para poder estimar el error y tomar decisiones sobre el tamaño del paso y otros más.

ESCUELA DE INGENIERIA ELECTRICA
UNIVERSIDAD DE COSTA RICA

2.7 Runge-Kutta-Simpson de 4° orden

Este es otro método de cuarto orden con un error por truncamiento $O(\Delta t^5)$, cuyas ecuaciones son

$$\begin{aligned}
 k_1 &= \Delta t f(y_n, t_n) \\
 k_2 &= \Delta t f(y_n + \frac{1}{3} k_1, t_n + \frac{1}{3} \Delta t) \\
 k_3 &= \Delta t f(y_n - \frac{1}{3} k_1 + k_2, t_n + \frac{2}{3} \Delta t) \\
 k_4 &= \Delta t f(y_n + k_1 - k_2 + k_3, t_n + \Delta t) \\
 y_{n+1} &= y_n + \frac{1}{8} [k_1 + 3k_2 + 3k_3 + k_4]
 \end{aligned} \tag{16}$$

2.8 Métodos Runge–Kutta de paso variable

Los métodos del tipo Runge-Kutta no tienen forma de evaluar el error cometido en cada iteración, sin embargo es posible combinar dos métodos orden diferente o utilizar un método con dos pasos de integración diferentes, para estimar el error y en base a este, decidir si es necesario o no cambiar el tamaño del paso de integración (reducirlo o incrementarlo), teniéndose entonces un método de paso variable.

Por ejemplo, los métodos Runge-Kutta-Fehlberg son métodos del tipo Runge-Kutta en los cuales se combinan dos métodos de diferente orden, para poder estimar el error cometido en cada iteración de la solución.

2.8.1 Runge-Kutta-Fehlberg de 2º orden

Este método está dado por las siguientes ecuaciones

$$\begin{aligned}
 k_1 &= \Delta t' f(y_n, t_n) \\
 k_2 &= \Delta t' f\left(y_n + \frac{1}{2}k_1, t_n + \frac{1}{2}\Delta t'\right) \\
 k_3 &= \Delta t' f\left(y_n + \frac{1}{256}k_1 + \frac{255}{256}k_2, t_n + \Delta t'\right) \\
 y_{n+1} &= y_n + \frac{1}{512}k_1 + \frac{255}{256}k_2 + \frac{1}{512}k_3
 \end{aligned} \tag{17}$$

donde $\Delta t'$ es el paso de integración variable, el cual debe dividirse a la mitad si el

$$Error > ErrAbs + ErrRel|y_{n+1}| \tag{18}$$

y puede duplicarse si el

$$Error < (ErrAbs + ErrRel|y_{n+1}|)/100 \tag{19}$$

donde

$$Error = \left| \frac{(k_1 - k_3)}{512} \right| \tag{20}$$

es el error por truncamiento estimado y $ErrAbs$ y $ErrRel$ son el error absoluto y relativo utilizados para controlar el paso de integración.

ESCUELA DE INGENIERIA ELECTRICA
UNIVERSIDAD DE COSTA RICA

2.8.2 Runge-Kutta-Fehlberg de 4° orden

En este método se utiliza un método de quinto orden para estimar el error cometido por un método de cuarto orden y tomar decisiones sobre el tamaño del paso de integración y está dado por las siguientes ecuaciones

$$\begin{aligned}
 k_1 &= \Delta t' f(y_n, t_n) \\
 k_2 &= \Delta t' f\left(y_n + \frac{1}{4} k_1, t_n + \frac{1}{4} \Delta t'\right) \\
 k_3 &= \Delta t' f\left(y_n + \frac{3}{32} k_1 + \frac{9}{32} k_2, t_n + \frac{3}{8} \Delta t'\right) \\
 k_4 &= \Delta t' f\left(y_n + \frac{1932}{2197} k_1 - \frac{7200}{2197} k_2 + \frac{7296}{2197} k_3, t_n + \frac{12}{13} \Delta t'\right) \\
 k_5 &= \Delta t' f\left(y_n + \frac{439}{216} k_1 - 8 k_2 + \frac{3680}{513} k_3 - \frac{845}{4104} k_4, t_n + \Delta t'\right) \\
 k_6 &= \Delta t' f\left(y_n - \frac{8}{27} k_1 + 2 k_2 - \frac{3544}{2565} k_3 + \frac{1859}{4104} k_4 - \frac{11}{40} k_5, t_n + \frac{1}{2} \Delta t'\right)
 \end{aligned} \tag{21}$$

donde $\Delta t'$ es el paso de integración variable.

Se calcula

$$R = \frac{1}{\Delta t'} \left| \frac{1}{360} k_1 - \frac{128}{4275} k_3 - \frac{2197}{75240} k_4 + \frac{1}{50} k_5 + \frac{2}{55} k_6 \right| \tag{22}$$

siendo este

$$R = \frac{\Delta t'}{2} |y_{n+1(5)} - y_{n+1(4)}| \tag{23}$$

y

$$\delta = 0.84 (TolE/R)^{1/4} \tag{24}$$

Si $R \leq TolE$, donde $TolE$ es la tolerancia o error permitido en la solución, se calcula el nuevo punto de la solución

$$y_{n+1} = y_n + \frac{25}{216} k_1 + \frac{1408}{2565} k_3 + \frac{2197}{4104} k_4 - \frac{1}{5} k_5 \tag{25}$$

y el nuevo paso de integración como $\Delta t' = \delta \Delta t'$, si $\delta \geq 4$ el paso solo se cuadruplica ($\Delta t' = 4 \Delta t'$), y se continua con la solución a partir del punto $n+1$.

Si $R > TolE$, se calcula el nuevo paso de integración como $\Delta t' = \delta \Delta t$, si $\delta \leq 0.1$ el paso se reduce solo a su décima parte ($\Delta t' = 0.1 \Delta t$), y se parte del punto n anterior.

2.9 Error por truncamiento y error por redondeo

Se indicó anteriormente que el error producido cuando el paso de integración Δt no es suficientemente pequeño para representar la solución de la ecuación diferencial en forma exacta, se denomina *error por truncamiento*, al haber truncado la serie de Taylor y no incluir términos más allá de un cierto Δt^k . Sin embargo, por otra parte consideraciones numéricas así como el tiempo total de solución, requieren que el Δt no se haga demasiado pequeño. La razón para esto es que el *error por redondeo* es mayor para pasos de integración pequeños debido al gran número de iteraciones requeridas para obtener la solución. El error por redondeo es el error resultante de la imposibilidad del computador digital de representar un número con más de un número de cifras significativas y esto dependerá del “largo de palabra” utilizado para la representación numérica.

Cuando el paso de integración es grande el error dominante es el error por truncamiento, mientras que para pasos pequeños, es más importante el error por redondeo.

2.10 Sistemas rígidos

Se dice que un sistema es “*rígido*” cuando presenta características dinámicas lentas y rápidas en forma simultánea, esto es que contiene unos elementos dinámicos con constantes de tiempo pequeñas y otros con constantes de tiempo grandes, como podría ser el caso de un sistema electromecánico en donde las constantes de tiempo de los componentes eléctricos, pueden ser muy pequeñas en comparación con las constantes de tiempo de los componentes mecánicos.

La presencia de dinámicas lentas y rápidas dentro del mismo sistema de ecuaciones diferenciales, introduce dificultades al momento de obtener su solución numérica.

Para garantizar la estabilidad y exactitud de la solución de las ecuaciones de la parte dinámica rápida, se puede requerir un paso de integración pequeño, el cual puede ser demasiado pequeño para la solución de la parte dinámica lenta, extendiendo innecesariamente la obtención de la solución en todo el intervalo de solución deseado, si el paso se mantiene constante.

Una posible solución al problema planteado por la presencia de constantes de tiempo muy diferentes, puede ser la utilización de métodos de paso variable, los cuales pueden ajustar el paso de integración de acuerdo a los requerimientos del sistema, utilizando pasos pequeños inicialmente y luego que la parte de dinámica rápida haya desaparecido, utilizando entonces paso de integración mayor, sin embargo esto tiene un costo extra por la necesidad de efectuar evaluaciones adicionales de las ecuaciones diferenciales y comparaciones para determinar el ajuste del paso.

Para la solución de los sistemas rígidos son usualmente preferidos los métodos numéricos de solución de ecuaciones diferenciales basados en las fórmulas de diferencias retrógradas implícitas.

2.11 Efecto del paso de integración sobre la exactitud de la solución

El tamaño del paso de integración utilizado para la solución de la ecuación diferencial, afecta directamente la exactitud de la misma. Normalmente se desea emplear el paso de integración mayor posible para obtener una solución rápida, pero no tan grande que introduzca errores apreciables en esta. Al utilizar los métodos de integración de paso fijo, es responsabilidad del usuario la selección del paso de integración adecuado.

En la Figura 2.11-1 se muestra la solución de la ecuación diferencial $\frac{dy}{dx} = \cos(x)$; $y(0)=0$ utilizando el método de Euler (Runge-Kutta de primer orden), con cuatro pasos de integración diferentes, uno lo suficientemente pequeño para considerar la solución como “exacta”. Se puede apreciar fácilmente como se deteriora la solución a medida que se aumenta el paso de integración y la necesidad de utilizar pasos suficientemente pequeños para tener una solución con un error mínimo.

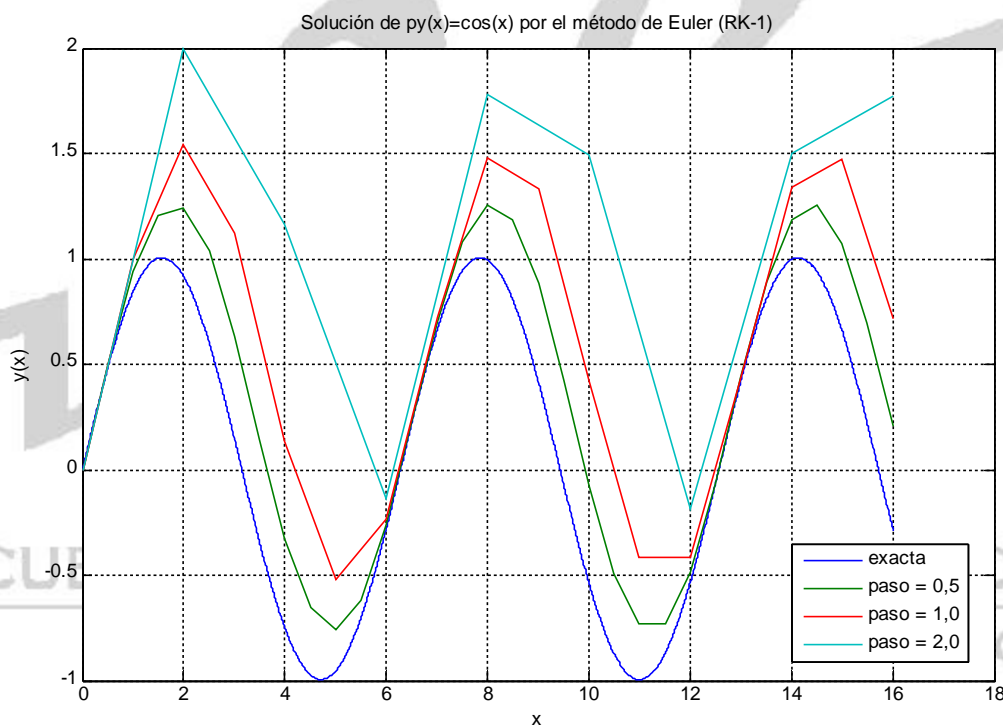


Figura 2.11-1 Efecto del paso de integración – método de Euler

En la Figura 2.11-2 se muestra la solución de la misma ecuación diferencial anterior pero utilizando ahora el método de Simpson (Runge-Kutta de tercer orden) y los mismos pasos de integración empleados anteriormente. En este caso se puede apreciar que para un mismo tamaño de paso, la solución con el método de Simpson es mucho más exacta que la obtenida con el método de Euler.

Aunque el método de Simpson requiere realizar tres veces más evaluaciones de la ecuación diferencial en cada paso de integración que el método de Euler, normalmente permite emplear pasos de integración bastante mayores, reduciéndose incluso el tiempo total requerido para la solución.

El tamaño del paso de integración necesario para obtener una solución satisfactoria depende entonces de la exactitud del método de integración y de la ecuación diferencial a resolver. Entre más preciso, mayor orden, sea el método de integración utilizado, se pueden emplear paso mayores, sin embargo debe tenerse también en cuenta que, en los casos en que se desea mostrar la solución de la ecuación diferencial en forma gráfica, el tamaño máximo del paso estará determinado usualmente por la resolución deseado de este gráfico.

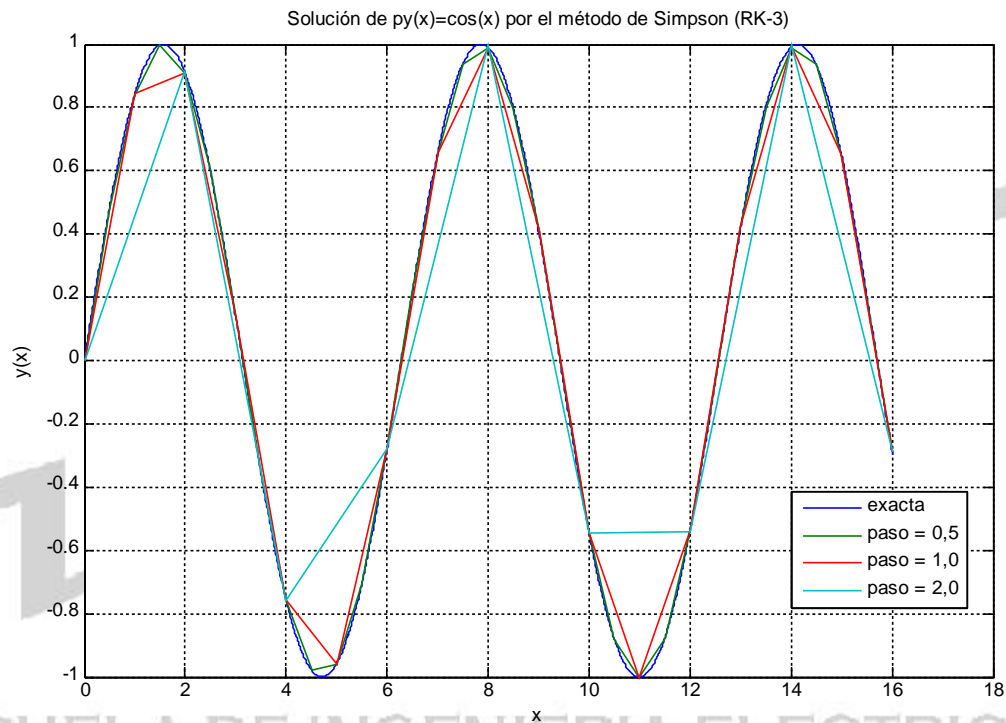


Figura 2.11-1 Efecto del paso de integración – método de Simpson

3. Métodos de integración numérica

La expresión general de los métodos de integración numérica está dada por

$$y_{n+1} = \sum_{i=0}^p a_i y_{n-i} + \Delta t \sum_{i=-1}^p b_i \dot{y}_{n-i} \quad (26)$$

en donde el último valor de t en que se calculó y es t_n y el número de valores pasados usados para calcular y_{n+1} es $p+1$.

Se considerará que en (26) algunos de los a_i y b_i pueden ser cero, pero se supondrá que a_p o b_p no son cero.

Si en (26) $b_{-1}=0$, y_{n+1} está expresado como una combinación lineal de los valores conocidos de y y \dot{y} y es fácil de calcular. Las fórmulas con $b_{-1}=0$ se denominan *fórmulas de integración directa* o abiertas

Si $b_{-1} \neq 0$, la ecuación (26) es una función implícita de y_{n+1} porque $\dot{y}_{n+1} = f(y_{n+1}, t_{n+1})$ y se puede resolver solamente por procedimientos iterativos. Las fórmulas con $b_{-1} \neq 0$ se denominan *fórmulas de integración implícitas* o cerradas.

Como regla general, para fórmulas del mismo orden, las fórmulas implícitas son sustancialmente más exactas que las fórmulas directas.

En comparación con los métodos de integración del tipo Runge-Kutta, los métodos de integración numérica se caracterizan entonces por:

1. no ser autoiniciables
2. requerir aparte de la información del punto n , información de uno o más puntos anteriores para calcular la solución en el punto $n+1$
3. evaluar la función (derivada) una sola vez por cada aplicación de la fórmula (el número de evaluaciones de la derivada en el caso de los métodos predictor - corrector dependerá del número de veces que sea necesario aplicar la fórmula correctora) y
4. por que el uso combinado de fórmulas explícitas e implícitas permite estimar y corregir los errores por truncamiento locales. Esta estimación del error se puede utilizar para variar el tamaño del paso de integración.

3.1 Fórmulas directas, Métodos de Adams-Bashforth

Las fórmulas directas (*abiertas*) pueden utilizarse solas. Al uso de las fórmulas directas de Adams solas, se les llama métodos o predictores de Adams-Bashforth.

Los *métodos de Adams-Bashforth* más utilizados son los de segundo y cuarto orden. Como se podrá apreciar en sus ecuaciones, estas requieren del conocimiento de la derivada evaluada en uno o hasta tres instantes anteriores respectivamente, por lo que la solución de la ecuación

diferencial debe iniciarse con un procedimiento del tipo Runge-Kutta del mismo orden para obtener la información inicial requerida por el método de Adams.

3.1.1 Método de segundo orden

Este método está dado por la ecuación

$$y_{n+1} = y_n + \frac{1}{2} \Delta t [3 \dot{y}_n - \dot{y}_{n-1}] \quad (27)$$

cuyo error por truncamiento es $O(\Delta t^3)$.

3.1.2 Método de cuarto orden

Esta fórmula directa está dada por la ecuación

$$y_{n+1} = y_n + \frac{1}{24} \Delta t [55 \dot{y}_n - 59 \dot{y}_{n-1} + 37 \dot{y}_{n-2} - 9 \dot{y}_{n-3}] \quad (28)$$

con un error por truncamiento $O(\Delta t^5)$.

3.2 Fórmulas implícitas, Métodos de Adams-Moulton

Las fórmulas implícitas (*cerradas*) no pueden utilizarse solas y se les llamará *métodos o correctores de Adams-Moulton*.

3.2.1 Método de segundo orden

Este método está dado por la ecuación

$$y_{n+1} = y_n + \frac{1}{2} \Delta t [\dot{y}_{n+1} - \dot{y}_n] \quad (29)$$

el cual es comúnmente llamado método trapezoidal cerrado.

3.2.2 Método de cuarto orden

Esta fórmula está dada por la ecuación

$$y_{n+1} = y_n + \frac{1}{24} \Delta t [9 \dot{y}_{n+1} + 19 \dot{y}_n - 5 \dot{y}_{n-1} + \dot{y}_{n-2}] \quad (30)$$

3.3 Métodos predictor - corrector

Si bien las fórmulas directas se pueden utilizar solas, proveyéndoles la información de arranque necesaria, las fórmulas implícitas no pueden emplearse solas.

Para poder hacer uso de las fórmulas de integración numérica para el cálculo de y_{n+1} cuando $b_{-1} \neq 0$, se debe obtener primero una estimación del valor de y_{n+1} denominado $y_{n+1}^{(0)}$, calcular luego $f(y_{n+1}^{(0)}, t_{n+1})$ y utilizar (26) para calcular $y_{n+1}^{(1)}$.

Se debe entonces *predecir* primero el valor de y_{n+1} utilizando una fórmula de integración directa y luego *corregir* el valor predicho con una fórmula de integración implícita, creándose así los métodos del tipo *predictor – corrector*, los cuales emplean una fórmula directa como predictor y una fórmula implícita como corrector, ambas con errores por truncamiento del mismo orden.

3.3.1 Método trapezoidal modificado (2° orden)

Uno de los métodos más sencillos del tipo predictor - corrector se obtiene utilizando los métodos de Adams de 2° orden (el de Adams-Bashforth como predictor y el de Adams – Moulton, integración trapezoidal cerrada, como corrector). En su versión más simple el corrector se utilizaría una sola vez en cada iteración, haciendo del mismo un método de paso fijo cuyas ecuaciones están dadas por

- Predictor

$$y_{n+1}^{(0)} = y_n + \frac{1}{2} \Delta t [3 \dot{y}_n - \dot{y}_{n-1}] \quad (31a)$$

$$\dot{y}_{n+1}^{(0)} = f(y_{n+1}^{(0)}, t_n + \Delta t)$$

- Corrector

$$y_{n+1} = y_n + \frac{1}{2} \Delta t [\dot{y}_{n+1}^{(0)} + \dot{y}_n] \quad (31b)$$

3.3.2 Método de Adams-Bashforth-Moulton de 4° orden

Al uso de una fórmula de Adams abierta (predictor) junto con una fórmula de Adams cerrada (corrector) se le conoce como *método de Adams-Bashforth-Moulton*, siendo el de 4° orden

- Predictor

$$y_{n+1}^{(0)} = y_n + \frac{1}{24} \Delta t [55 \dot{y}_n - 59 \dot{y}_{n-1} + 37 \dot{y}_{n-2} - 9 \dot{y}_{n-3}] \quad (32a)$$

$$\dot{y}_{n+1}^{(0)} = f(y_{n+1}^{(0)}, t_n + \Delta t)$$

- Corrector

$$y_{n+1}^{(j+1)} = y_n + \frac{1}{24} \Delta t [9 \dot{y}_{n+1}^{(j)} + 19 \dot{y}_n - 5 \dot{y}_{n-1} + \dot{y}_{n-2}] \quad (32b)$$

3.3.3 Método de Milne de 4° orden

Este es otro método predictor – corrector con un error $O(\Delta t^5)$ cuyas ecuaciones son

- Predictor

$$y_{n+1}^{(0)} = y_{n-3} + \frac{4}{3} \Delta t [2 \dot{y}_n - \dot{y}_{n-1} + 2 \dot{y}_{n-2}] \quad (33a)$$

$$\dot{y}_{n+1}^{(0)} = f(y_{n+1}^{(0)}, t_n + \Delta t)$$

- Corrector

$$y_{n+1}^{(j+1)} = y_{n-1} + \frac{1}{3} \Delta t [\dot{y}_{n+1}^{(j)} + 4 \dot{y}_n + \dot{y}_{n-1}] \quad (33b)$$

3.3.4 Método de Milne de 6° orden

Este es un método predictor – corrector con un error $O(\Delta t^7)$ cuyas ecuaciones son

- Predictor

$$y_{n+1}^{(0)} = y_{n-5} + \frac{3}{10} \Delta t [11 \dot{y}_n - 14 \dot{y}_{n-1} + 26 \dot{y}_{n-2} - 14 \dot{y}_{n-3} + 11 \dot{y}_{n-4}] \quad (34a)$$

$$\dot{y}_{n+1}^{(0)} = f(y_{n+1}^{(0)}, t_n + \Delta t)$$

- Corrector

$$y_{n+1}^{(j+1)} = y_{n-3} + \frac{2}{45} \Delta t [7 \dot{y}_{n+1}^{(j)} + 32 \dot{y}_n + 12 \dot{y}_{n-1} + 32 \dot{y}_{n-2} + 7 \dot{y}_{n-3}] \quad (34b)$$

3.4 Algoritmo de solución

La utilización combinada de los métodos de integración del tipo Runge-Kutta con los métodos numéricos predictor – corrector, permite desarrollar un algoritmo de solución de ecuaciones diferenciales de paso variable el cual en términos generales comprendería:

1. Utilizar un algoritmo Runge–Kutta para iniciar la solución y obtener la información requerida por el método predictor – corrector
2. En cada iteración
 - Predecir el valor de $y_{n+1}^{(0)}$
 - Corregir para obtener $y_{n+1}^{(1)}$
3. Si la diferencia porcentual entre el valor predicho y el corregido es menor que un valor ϵ arbitrariamente pequeño, pero mayor que un valor δ dado, entonces continuar

4. Si la diferencia anterior no es menor que ϵ , utilizar entonces nuevamente el corrector obteniendo $y_{n+1}^{(2)}$. Si en dos iteraciones del corrector no se logra la precisión deseada, entonces el paso de integración debe dividirse a la mitad y volver a utilizar el método de Runge–Kutta a partir del punto n para continuar la solución, cambiando nuevamente al método predictor – corrector cuando se tenga la información requerida por este
5. Si la diferencia entre el valor predicho y el corregido es menor que δ , entonces el error cometido es muy pequeño y se puede acelerar la solución aumentando el paso de integración al doble, continuar con el método Runge–Kutta y luego el predictor – corrector nuevamente
6. El método continuará doblando o dividiendo por dos el paso de integración de manera de mantener el error por truncamiento local dentro de los límites establecidos y tratando en todo caso de usar el paso de integración mayor posible para obtener una solución rápida.



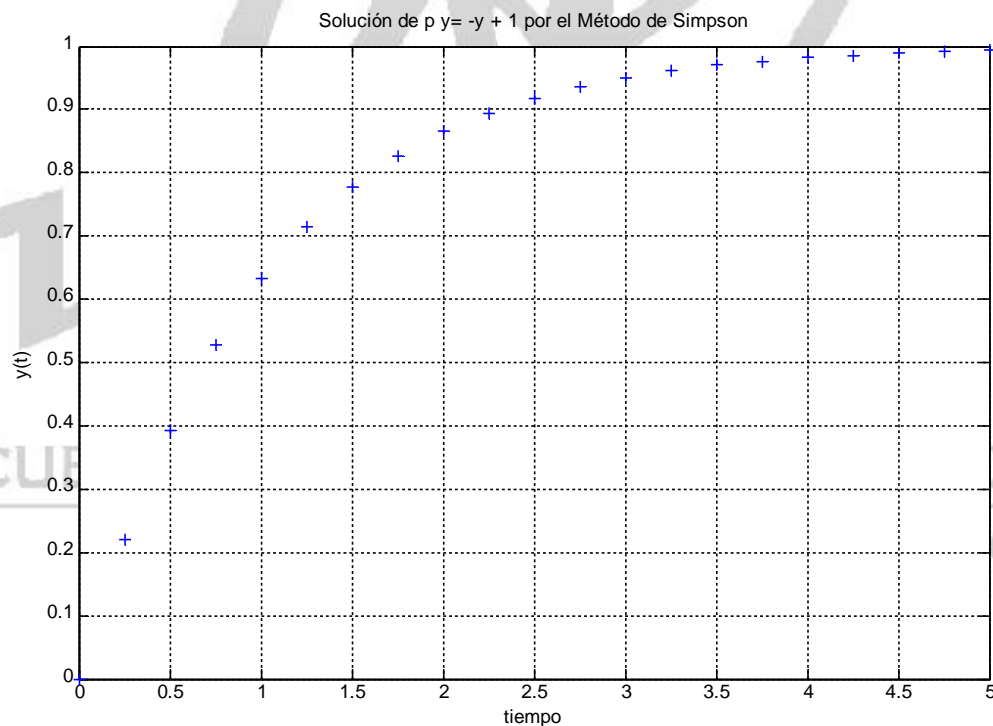
ESCUELA DE INGENIERIA ELECTRICA
UNIVERSIDAD DE COSTA RICA

4. Ejemplos del uso de algunos métodos

Los programas de diseño de sistemas de control asistido por computadora (CACSD) *MATLAB*® y *Scilab*, así como los de simulación digital de sistemas dinámicos *Simulink*® y *VisSim*™, incorporan varias rutinas para la solución de ecuaciones diferenciales ordinarias, o “ODE Solvers” por sus siglas en inglés, por lo que al utilizarlos usualmente no es necesario su programación a menos que se desee emplear un método diferente o realizar comparaciones entre algunos métodos no incluidos en ellos.

4.1 Método de la regla de Simpson en MATLAB

El siguiente es el gráfico de la solución de la ecuación diferencial $\frac{dy}{dt} = -y(t) + 1$ empleando el método de Simpson, un Runge-Kutta de 3^{er} orden, y el correspondiente listado de las instrucciones en *MATLAB*.



```
%Ejemplo 4.1
%Solución de una ecuación diferencial utilizando
%el método de Simpson (Runge-Kuta de 3er orden)
%Ecuación de ejemplo: p y = f(y,t) = -y + 1
%
```

```

clear
%definir la ecuación diferencial a resolver
f=inline('-y+1','y','t');
%
%condición inicial y tiempos
yo=0.;to=0;dt=0.25;tu=5;
y(1)=yo;t(1)=to;
n=0;
%
%solución numérica
while t(n+1) < tu
    n=n+1;
    k1=dt*f(y(n),t(n));
    k2=dt*f(y(n)+k1/2,t(n)+dt/2);
    k3=dt*f(y(n)-k1+2*k2,t(n)+dt);
    y(n+1)=y(n)+(k1+4*k2+k3)/6;
    t(n+1)=t(n)+dt;
end
%
%graficar la solución
plot(t,y,'+b')
grid on
title('Solución de p y= -y + 1 por el Método de Simpson')
xlabel('tiempo'),ylabel('y(t)')

```

4.2 Método de Adams-Bashforth de 2° orden en Scilab

El siguiente es el listado de instrucciones en *Scilab* y el gráfico de la solución de la ecuación diferencial $\frac{dy}{dt} = -0.025 y(t) + t$ empleando el método de Adams-Bashforth de 2° orden.

```

//Ejemplo 4.2
//Solución de una ecuación diferencial utilizando
//el método de Adams-Bashforth de 2° orden
//Ecuación de ejemplo: p y = f(y,t) = -0.025 y + t
//
clear,xset('default'),xbasc()
//definir la ecuación diferencial a resolver
function py=f(y,t)
    py=-0.025*y+t;
endfunction
//
//condición inicial y tiempos
yo=0.;to=0;dt=0.25;tu=10;
y(1)=yo;t(1)=to;
n=0;
//solución numérica
//arranque con integración trapezoidal
n=n+1;
k1=dt*f(y(n),t(n));

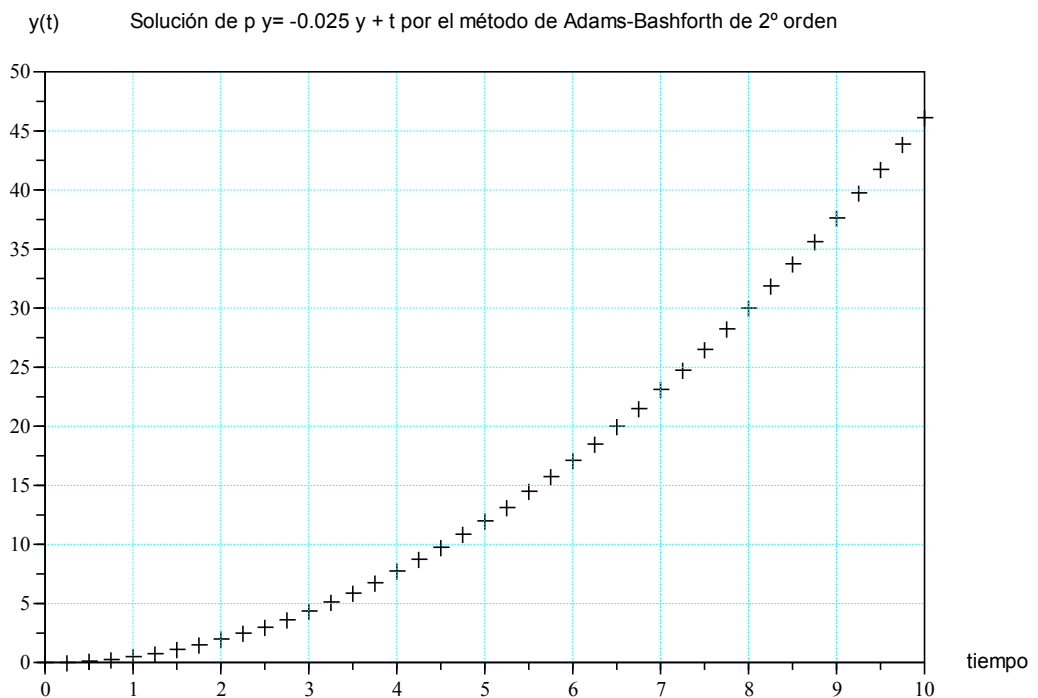
```



```

k2=dt*f(y(n)+k1,t(n)+dt);
y(n+1)=y(n)+(k1+k2)/2;
t(n+1)=t(n)+dt;
//cambio al método de Adams-Bashforth
while t(n+1) < tu
    n=n+1;
    y(n+1)=y(n)+dt*(3*f(y(n),t(n))-f(y(n-1),t(n-1)))/2;
    t(n+1)=t(n)+dt;
end
//
//graficar la solución
xname('Solución numérica de una ecuación diferencial')
xset('color',2),xset('mark size',1,18)
plot2d(t,y,style=-1)
xgrid(4)
xset('color',1),xset('font',2,3)
xtitle('Solución de p y= -0.025 y + t por el método de Adams-Bash-
forth de 2° orden','tiempo','y(t)')

```



4.3 Solución de una ecuación diferencial de segundo orden

Para obtener la solución de una ecuación diferencial de orden n , esta debe convertirse en un conjunto de n ecuaciones diferenciales de primer orden.

Por ejemplo, la ecuación diferencial lineal de segundo orden

$$\frac{d^2 y}{dt^2} + a_1 \frac{dy}{dt} + a_0 y(t) = b_0 u(t) \quad (35)$$

puede convertirse en dos ecuaciones diferenciales de primer orden definiendo

$$x_1(t) = y(t), \quad x_2(t) = dy/dt \quad (36)$$

para obtener

$$\frac{dx_1}{dt} = x_2(t), \quad \frac{dx_2}{dt} = -a_0 x_1(t) - a_1 x_2(t) + b_0 u(t) \quad (37)$$

Para el caso de una ecuación diferencial lineal de orden alto (n) dada por

$$\frac{d^n y}{dt^n} + a_{n-1} \frac{d^{n-1} y}{dt^{n-1}} + \dots + a_1 \frac{dy}{dt} + a_0 y(t) = b_0 u(t) \quad (38)$$

definiendo

$$x_1(t) = y(t), \quad x_2(t) = \frac{dy}{dt}, \dots, x_n = \frac{d^{n-1} y}{dt^{n-1}} \quad (39)$$

se puede convertir en un conjunto de n ecuaciones diferenciales de primer orden dado por

$$\begin{aligned} \dot{x}_1 &= x_2 \\ \dot{x}_2 &= x_3 \\ &\vdots \\ \dot{x}_n &= -a_0 x_1 - a_1 x_2 - \dots - a_{n-1} x_n + b_0 u(t) \end{aligned} \quad (40)$$

En el caso general de un conjunto de n ecuaciones diferenciales de primer orden no lineales se tendría

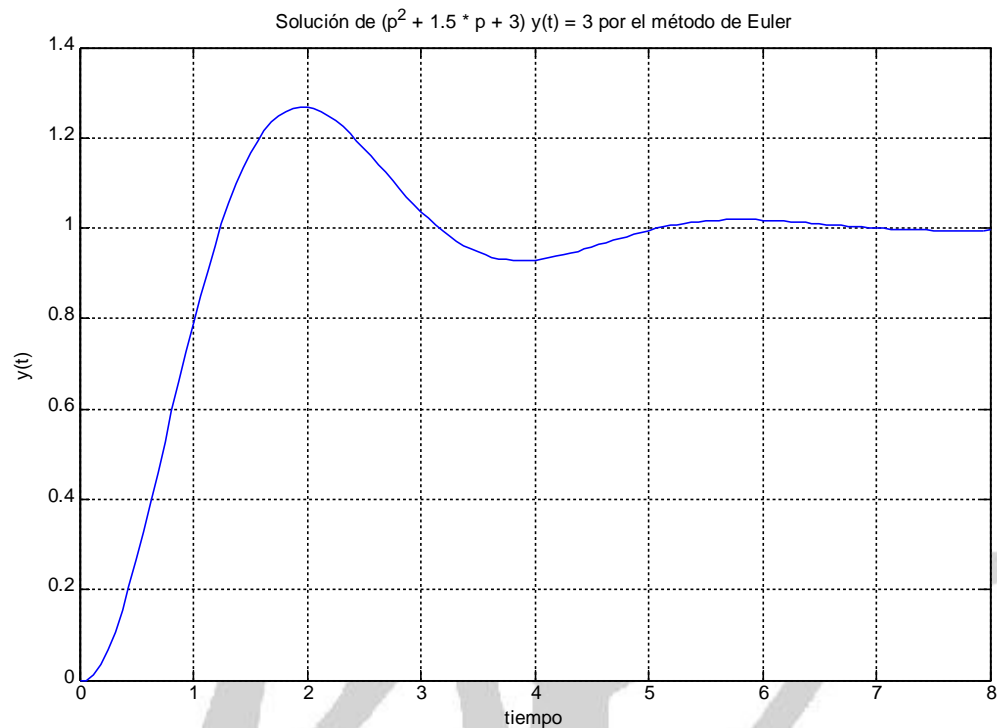
$$\begin{aligned} \dot{x}_1(t) &= f_1(x_1, x_2, \dots, x_n; u; t) \\ \dot{x}_2(t) &= f_2(x_1, x_2, \dots, x_n; u; t) \\ &\vdots \\ \dot{x}_n(t) &= f_n(x_1, x_2, \dots, x_n; u; t) \end{aligned} \quad (41)$$

las cuales deben ser resueltas en forma simultánea.

El siguiente es el gráfico de la solución de la ecuación diferencial de segundo orden $\frac{d^2 y}{dt^2} + 1.5 \frac{dy}{dt} + 3y(t) = 3$ empleando el método de Euler, y el correspondiente listado de las instrucciones en *MATLAB*.

Aunque la respuesta de $y(t)$ se muestra en el gráfico como una línea continua, debe recordarse que la solución de la ecuación diferencial solo existe en los instantes $\{t_i\}$ en los cuales se calculó.

```
%Ejemplo 4.3
%Solución de una ecuación diferencial de segundo orden
%utilizando el método de Euler (Runge-Kutta de 1er orden)
%Ecuación de ejemplo: (p^2+1.5*p+3) y(t) = 3
%
clear
%definir la ecuación diferencial
%como dos ecuaciones diferenciales de 1er orden)
f1=inline('x2','x1','x2','t');
f2=inline('-1.5*x2-3*x1+3','x1','x2','t');
%
%condiciones iniciales y tiempos
x1o=0.;x2o=0;to=0;dt=0.0625;tu=8.0;
x1(1)=x1o;x2(1)=x2o;t(1)=to;
n=0;
%
%solución numérica
while t(n+1) < tu
    n=n+1;
    k11=dt*f1(x1(n),x2(n),t(n));
    k12=dt*f2(x1(n),x2(n),t(n));
    x1(n+1)=x1(n)+k11;
    x2(n+1)=x2(n)+k12;
    t(n+1)=t(n)+dt;
end
%
%graficar la solución
plot(t,x1,'b')
grid on
title('Solución de (p^2 + 1.5 * p + 3) y(t) = 3 por el método de Euler')
xlabel('tiempo'),ylabel('y(t)')
```



4.4 “ODE Solvers” en MATLAB y Simulink

MATLAB y *Simulink* comparten el mismo conjunto de rutinas para la solución de ecuaciones diferenciales ordinarias el cual incluye los siguientes métodos:

4.4.1 Métodos de paso variable

- **Sistemas no rígidos**

- ode45 - fórmula Runge-Kutta (4,5) explícita, Dormand-Prince, exactitud media (método utilizado por omisión)
- ode23 - fórmula Runge-Kutta (2,3) explícita, Bogacki-Shampine, exactitud baja
- ode113 - fórmula de Adams-Bashforth-Moulton (1,13), exactitud baja a alta

- **Sistemas rígidos**

- ode15s - fórmula de diferencias retrógradas, Klopfenstein-Shampine (1,5), exactitud baja a alta
- ode23s - fórmula de modificada de Rosenbrock (2,3), exactitud baja
- ode23t - integración trapezoidal, sistema moderadamente rígidos, exactitud baja
- ode23tb - método híbrido, trapezoidal – diferencias retrógradas, exactitud baja

4.4.2 Métodos de paso fijo (solo en Simulink)

- ode1 - fórmula de Euler
- ode2 - fórmula de Euler modificada
- ode3 - versión de paso fijo del ode23
- ode4 - fórmula Runge-Kutta de 4º orden
- ode5 - versión de paso fijo del ode45

4.4.3 Ejemplo

El llamado general a las rutinas de solución de ecuaciones diferenciales en *MATLAB* es

$$[t,y] = \text{odesolver}('f', \text{intervalo}, y_0)$$

en donde *odesolver* es el nombre del método a utilizar, seleccionado de entre los listados en 4.4.1 o 4.4.2.

El siguiente es el listado de instrucciones y el gráfico correspondiente para la solución de un sistema de control proporcional de una planta de segundo orden utilizando el *ode45* de *Matlab*.

```
%Ejemplo 4.4
%Utilización de la rutina ode45 (Runge-Kutta 4/5)
%las ecuaciones diferenciales (modelo) están contenidas
%en el archivo modelo_44.m
%
clear
%
%Intervalo de solución
to=0;tu=20;
%
%llamado al "ode solver"
[t,y]=ode45('modelo_44',[to tu],[0 0]);
%
%graficar la solución
plot(t,y(:,1))
grid on
title('Ejemplo del uso de la rutina ODE45')
xlabel('tiempo, seg'),ylabel('y(t)')
```

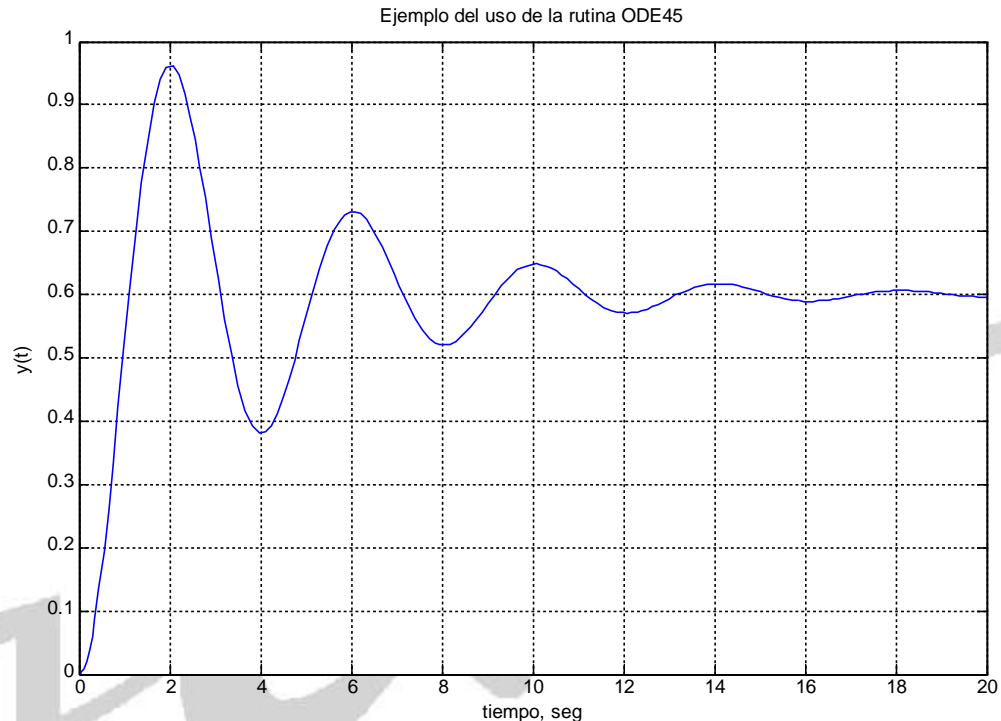
El “modelo” a resolver debe proporcionarse en un archivo separado.

```
function px=modelo(t,x)
%Ejemplo 4.4
%Archivo modelo_44.m con definición del modelo
%(conjunto de ecuaciones diferenciales)
%la función debe devolver un vector columna
%
%Modelo de ejemplo - control P de una planta de segundo orden
r=1;Kc=1.5;
%realimentación
```

```

e=r-x(1);
%controlador P
u=Kc*e;
%planta
px1=x(2);
px2=-x(1)-0.5*x(2)+u;
px=[px1; px2];

```



4.5 “ODE Solvers” en Scilab

Al igual que *MATLAB* el programa *Scilab* provee varias rutinas para la solución de un conjunto de ecuaciones diferenciales ordinarias no lineales.

El llamado general al método de solución es

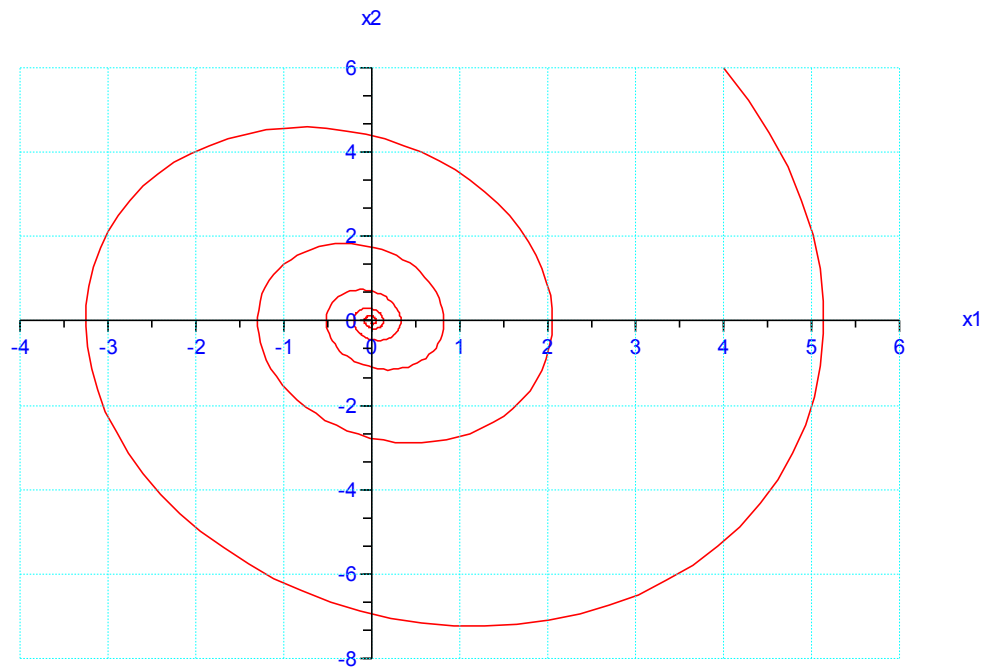
$$y = \text{ode}([\text{tipo}], y_0, t_0, t, f)$$

en donde el “tipo” define la rutina a utilizar teniéndose disponibles las siguientes:

- “adams” - método predictor -corrector de Adams, sistemas no rígidos
- “stiff” - método de diferencias retrógradas, sistemas rígidos
- “rk” - Runge-Kutta de 4 orden, paso variable
- “rkf” - Runge-Kutta-Felhberg (4,5), Shampine-Watts, sistemas no rígidos o de rigidez moderada

- “fix” - igual a rkf, menos argumentos en la llamada
- “root” - rutina de solución con capacidad de encontrar raíces
- “discrete” - simulación en tiempo discreto
- omisión - selecciona automáticamente entre “adams” y “stiff”

El siguiente es el listado de instrucciones en *Scilab* y el gráfico correspondiente, para la obtención del movimiento en el plano de fase de los estados de un sistema de segundo orden libre y con condiciones iniciales diferentes de cero, utilizando la instrucción *ode*.



//Ejemplo 4.5

```
//Utilización de la instrucción ode de Scilab
//Dibujo del movimiento en el plano de fase
//de los estados de un sistema de segundo orden
//Función de ejemplo:  $(p^2 + 0.5p + 3) x(t) = 0$ 
//
clear
t=0:0.05:20;
//definir el sistema como una función
//dos ecuaciones diferenciales de primer orden
function px=modelo(t,x)
    px(1)=x(2)
    px(2)=-3*x(1)-0.5*x(2)
endfunction
//
```

```
//llamado al "ode solver" con el método Runge-Kutta de 4 orden
x=ode('rk',[4;6],0,t,modelo);
//
//dibujar el plano de fase
plot2d(x(1,:),x(2,:),style=5)
xgrid(4),xtitle('','x1','x2')
//
//el cambio de la posición de los ejes coordenados
//y del tamaño y color de las escalas y leyendas,
//se realizó con las facilidades de Scilab
//para la edición de las propiedades de las figuras
```

4.6 Métodos de integración en VisSim

Este programa de simulación digital permite seleccionar entre los siguientes métodos de solución del sistema dinámico:

- Euler
- Trapezoidal
- Runge-Kutta 2° orden (Euler modificado)
- Runge-Kutta de 4° orden
- Runge-Kutta de 5° orden (paso variable)
- Bulirsh-Stoer (polinomios racionales)
- Euler (diferencias retrógradas), para sistemas rígidos

ESCUELA DE INGENIERIA ELECTRICA
UNIVERSIDAD DE COSTA RICA

5. Comparación de los métodos de solución de EDO

Como se ha indicado anteriormente, la exactitud de un método numérico de solución de ecuaciones diferenciales depende de su tipo y orden, y del paso de integración utilizado. Es de utilidad contar entonces con una programa o herramienta que permita comparar las características de la solución de una ecuación diferencial, utilizando diferentes métodos de solución y diferentes pasos de integración.

El programa SysQuake (Calerga, Suiza), es un programa *CACSD* similar en cierta forma al Matlab, pero que incorpora facilidades extra para la interacción entre el usuario y los programas desarrollados en este.

A continuación se presenta la solución de la ecuación diferencial $\frac{dx}{dt} = -0.5x^2 + 2$ obtenida utilizando el método de Euler y el Runge-Kutta de cuarto orden con un paso de integración $\Delta t = 0.80$ y el programa *ucr.ie431.solución_edo.sq*¹ Versión 1.2² desarrollado en SysQuake LE.

Este programa permite seleccionar entre los siguientes métodos de solución de ecuaciones diferenciales:

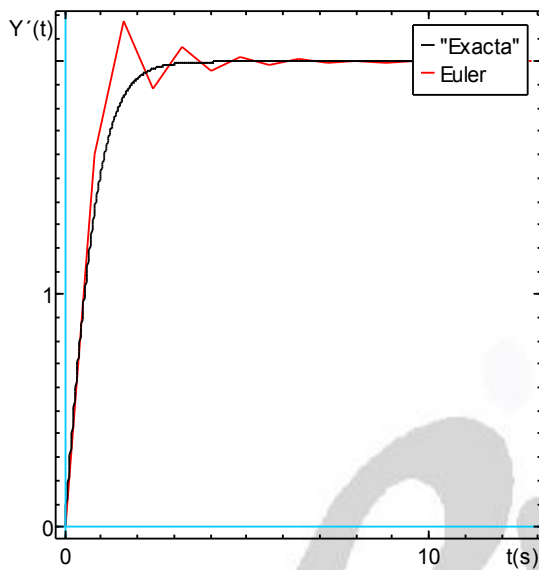
- Euler
- Euler modificado
- Integración trapezoidal
- Regla de Simpson
- Heun
- Runge-Kuta de 4º orden
- Runge-Kutta-Simpson

Se puede variar el paso de integración Δt para ver su efecto en la solución diferencial y también cambiar la ecuación diferencial que se desea resolver.

¹ disponible entre los ejemplos de Sysquake para el curso IE-431 Sistemas de control, en la dirección electrónica <http://www.eie.ucr.ac.cr/htm/pd/alfaro/sqdemos.htm>

² la versión original de este programa forma parte de las rutinas desarrolladas por Aragón^[1]

Solución de la EDO por el método de Euler



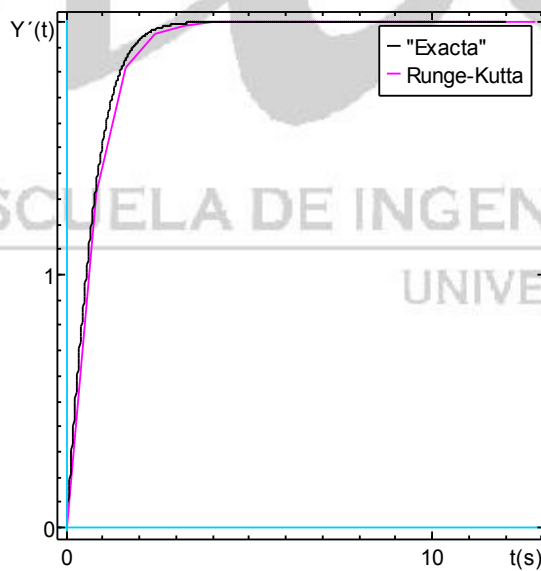
Paso de integración

dt (0.800):

Método de solución numérica

- ☒ Euler
- ☐ Trapezoidal
- ☐ Heun
- ☐ Runge-Kutta-Simpson
- ☐ Euler modificado
- ☐ Simpson
- ☐ Runge-Kutta

Solución de la EDO por Runge-Kutta 4º orden



Paso de integración

dt (0.800):

Método de solución numérica

- ☐ Euler
- ☐ Trapezoidal
- ☐ Heun
- ☐ Runge-Kutta-Simpson
- ☐ Euler modificado
- ☐ Simpson
- ☒ Runge-Kutta

Bibliografía

- Aragón, M. (2003) - “Desarrollo de herramientas en SysQuake para el curso Sistemas de Control”, Proyecto Eléctrico, Escuela de Ingeniería Eléctrica, Universidad de Costa Rica
- Burden, R.L. y J.D. Faires (1998) - Análisis Numérico, Sexta Edición, México, International Thompson Editores
- Calerga Sarl (2003) - “SysQuake 2 User Manual”, Lausana, Suiza
- Gomez, C. (Editor) (1999) - “Engineering and Scientific Computing with Scilab”, New York, NY, EUA, Birkhäuser Boston
- Guy, J.L. (1982) - “Modeling process systems via digital computers”, Chemical Engineering (EUA), Marzo 8
- INRIA Group (2001) - “Scilab Reference Manual”, Institut National de Recherche en Informatique et en Automatique – Project Meta 2, Rocquencourt, Francia
- Ketter, R.L. y S.P. Prawl (1969) – “Modern Methods of Engineering Computation”, New York, NY, EUA, McGraw-Hill, Inc.
- Kochenburger, R.J. (1972) - “Computer Simulation of Dynamic Systems”, Englewood Cliffs, NJ, EUA, Prentice-Hall, Inc.
- Korn, G.A. y J.V. Wait (1978) – “Digital Continuous System Simulation”, Englewood Cliffs, NJ, EUA
- Ralston, A. (1965) - “A First Course in Numerical Analysis”, Tokyo, Japón, McGraw-Hill/Kogakusha Co.
- Shampine, L.F. y M.W. Reichelt (1997) – “The MATLAB ODE Suite”, SIAM Journal of Scientific Computing (EUA), Vol. 18 N° 1, pág. 1-22
- The Matworks, Inc. (2002) - “Using Matlab®- Version 6”, Natick, MA, EUA
- The Matworks, Inc. (2002) - “Using Simulink®- Version 5”, Natick, MA, EUA
- Thomas, B. (1986) - “The Runge-Kutta Methods”, Byte (EUA), Abril
- Visual Solutions, Inc. (1999) - “VisSim™ User Guide – Version 3”, Westford, MA, EUA