Cálculo Numérico

Errores

Tener Presente

Interesa saber cual es el grado de precisión y de que depende.

Muchos métodos se basan en sucesiones infinitas que convergen a la solución.

Finalmente, se debe truncar la sucesión y determinar el error. Hay que estudiar la naturaleza de este error.

También interesa conocer la velocidad de convergencia.

Definiciones

Definición 1:
Sea {Xn} una sucesión que converge a x, en un espacio vectorial normado. El error de truncación n – ésimo será.

$$en = xn - x$$

Sea {βn} una sucesión de números reales positivos que converge a cero, (típicamente βn = 1/np para algún p entero positivo). Se dirá que la sucesión {Xn} converge con una velocidad de convergencia caracterizada por un O(βn) si existe un constante K positiva, tal que a partir de un cierto n, para todo n≥No

$$\| en \| \leq K \beta n$$

Gauss

Que permite resolver sistemas lineales, es un método que no está exento de errores.

Porque los número tienen decimales y dichos números se operan, (+, -, *. /)

Al operar los números se propagan errores

Hay situaciones en que un error pequeño inicial, al manipularse produce un error mayor. Esto se denomina **Inestabilidad**.

Por ejemplo un atraso de 5’ en una ruta de tráfico, produce un retraso de 30’ al llegar al destino.

Sistemas mal Condicionados

La solución de un sistema de dos por dos, geométricamente corresponde a la intersección de dos líneas en el plano

a. Si dos rectas son paralelas podrá ser infinitas soluciones (coinciden) o no tendrá solución (no coinciden). En ambos casos las rectas son linealmente dependientes y no invertibles.

b. Pero si son casi paralelas, entonces habrá una solución única y se mueven un poco la solución se encontrará distante de la primera.

Si se considera el sistema:

$$\begin{bmatrix} 1.000 & 2.000 \\ 0.499 & 1.001 \end{bmatrix} \begin{bmatrix} X \\ Y \end{bmatrix} = \begin{bmatrix} 3.0 \\ 1.5 \end{bmatrix} \quad \text{Su solución} \quad \begin{bmatrix} 1.0 \\ 1.0 \end{bmatrix}$$

Cambiando ligeramente la matriz

$$\begin{bmatrix} 1.000 & 2.000 \\ 0.500 & 1.001 \end{bmatrix} \begin{bmatrix} X \\ Y \end{bmatrix} = \begin{bmatrix} 3.0 \\ 1.5 \end{bmatrix} \quad \text{Su solución} \quad \begin{bmatrix} 3.0 \\ 0.0 \end{bmatrix}$$

Definición 2:

Si x* es un número que se aproxima a x, se define el error absoluto como:

$$E_A(x^*) = |x - x^*|$$

Y el error relativo.

$$E_R(x^*) = |x - x^*| \ / \ |x|$$

Se tiene ε tal que $E_A(x^*) \leq \varepsilon$, así $x \in [x-\varepsilon, x+\varepsilon]$

Intervalo de confianza $x \in [a, b] \longrightarrow x^* = [a + b]/2$ es la mejor aproximación a x.

$$E_A(x^*) \leq (b-a)/2$$

Propagación de Errores

Una expresión de n números $X_1, X_2, \ldots, X_n$ se conocen de manera aproximada por $X^*_1, X^*_2, .., X^*_n$.

Entonces el error para cada una de estas aproximaciones: $\Delta = X_i - X^*_i$, contribuirá al error de la expresión final:

$$Z - Z^* = \Phi(X_1, X_2, \ldots, X_n) - \Phi(X^*_1, X^*_2, .., X^*_n)$$

Si las derivadas parciales de Φ son continuas, por medio de una serie de Taylor.

Existe ξ en el segmento que una ambas n – tuplas $X = (X_1, X_2, \ldots, X_n)^t$ y $X^* = (X^*_1, X^*_2, .., X^*_n)^t$

tal que: $$Z = Z^* + \sum_{j=1}^{n} \frac{\partial \Phi(X^*)}{\partial X_j} \Delta_j + \sum_{i,j=1}^{n} \frac{\partial^2 \Phi(\xi)}{\partial X_i \partial X_j} \Delta_i \Delta_j$$ (0) (errores pequeños cuadráticos)

$$Z - Z^* \approx \sum_{j=1}^{n} \frac{\partial \Phi(X^*)}{\partial X_j} \Delta_j$$

Representación Punto Flotante

Para la representación en un computador de los números con una base β pre-establecida, coda número con una cantidad finita de dígitos p, de la forma:

$0.a_1a_2\ldots a_p\beta^{\alpha}$ representación en punto flotante

$a_i \in [0, \beta-1]$ y $\alpha \in [m,M]$

Así, la precisión del computador quedará caracterizada por:

- Base β
- el largo máximo de palabra p
- el rango del exponente $\alpha \in [m,M]$

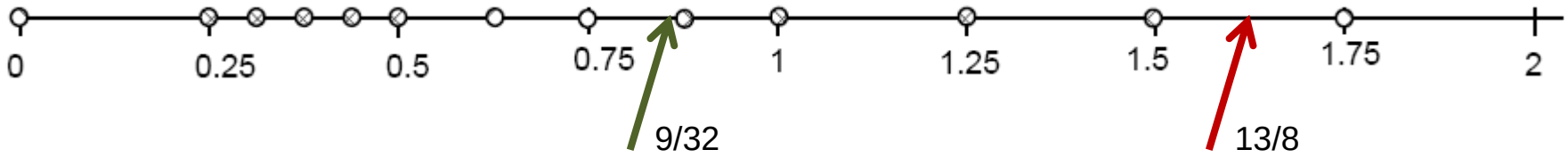
Considerando los números representados por una Máquina

β = 2
p = 3
m= -1
M= 1

Este conjunto es: $A = \left\{ \begin{array}{cccccc} 0 & \pm 0.111 & \pm 0.110 & \pm 0.101 & \pm 0.100 & \pm 0.111 \cdot 2^{-1} \quad \pm 0.1110 \cdot 2^{-1} \\ & \pm 0.111 \cdot 2 & \pm 0.110 \cdot 2 & \pm 0.101 \cdot 2 & \pm 0.100 \cdot 2 & \pm 0.101 \cdot 2^{-1} \quad \pm 0.100 \cdot 2^{-1} \end{array} \right\}$

Corresponde a: $A = \left\{ \begin{array}{cccccc} 0 & \pm\frac{7}{8} & \pm\frac{3}{4} & \pm\frac{5}{8} & \pm\frac{1}{2} & \pm\frac{7}{16} \quad \pm\frac{3}{8} \\ & \pm\frac{7}{4} & \pm\frac{3}{2} & \pm\frac{5}{4} & \pm 1 & \pm\frac{5}{16} \quad \pm\frac{1}{4} \end{array} \right\}$

En la representación de los positivos:



(no hay una distribución uniforme)

Todo número cuya representación sea mayor que M será considerado ∞
(Matlab NaN) Not a Number

Todo número cuya representación sea menor que m será considerado 0

Para representar un número no representable, debido a que su representación tiene más dígitos que p. Por uno representable, el computador debe redondear o truncar en p dígitos.

Sea $X = s*a1*a2..ap*ap+1*\beta^{\alpha}$ $m \leq \alpha \leq M$ y $s \in [-1,1]$ y fl(X) es la representación de punto flotante adecuado.

Entonces el error cometido fl(X):

Redondeo: $E_A(fl(X)) = |X - fl(X)| \leq (1/2)[0, \beta^{\alpha-p}] = \beta^{\alpha-p}/2$

$E_R(fl(X)) = |X - fl(X)| \leq (1/2)[0, \beta^{\alpha-p}/\beta^{\alpha-1}] = \beta^{1-p}/2$

Truncación: $E_A(fl(X)) = |X - fl(X)| \leq (1/2)[0, \beta^{\alpha-p}] = \beta^{\alpha-p}$

$E_R(fl(X)) = |X - fl(X)| \leq (1/2)[0, \beta^{\alpha-p}/\beta^{\alpha-1}] = \beta^{1-p}$

Se estima mediante la expresión, la propagación del error.

Definición 3 $$Z - Z^* \approx \sum_{j=1}^{n} \frac{\partial \Phi(X^*)}{\partial X_j} \Delta j$$

Una evaluación de Φ es una fuente de error que no ha considerado la expresión, en la definición 3.

Por ejemplo si Φ(x1,x2) = x1 + x2, asumiendo que x1 y x2 son representables, es decir x1,x2 Є **A**

no necesariamente la suma será representable y deberá aproximarse.

Sin embargo en vez de obtenerse: $Z^* = \Phi(X^*_1, X^*_2, .., X^*n)$ se obtiene $\tilde{Z} = fl(\Phi(X^*))$

Luego si cada operación hay redondeo, el orden en que hace es importante;

Por ejemplo: Φ(x1,x2,x3) = X1 + X2 + X3, con largo de palabra p= 8.

$X1 = 0.23371258*10^{-4}$
$X2 = 0.33678429*10^{2}$
$X3 = -0.33677811*10^{2}$

$fl(X1 + fl(X2 + X3)) = 0.64137126*10^{-3}$ (Mejor Algoritmo)

$fl(fl(X1 + X2) + X3) = 0.64100000*10^{-3}$

$X1 + X2 + X3 = 0.641371258*10^{-3}$

Para formalizar la influencia del algoritmo utilizado., si Φ se calcula con sucesivas operaciones:

$$Z = \Phi(X) = \Phi^{(r)}\Phi^{(r-1)}\,(\ldots\Phi^{0}(X)\ldots\,) = \Phi^{(r)}\mathrm{o}\Phi^{(r-1)}\,\mathrm{o}\ldots..\,\mathrm{o}\,\Phi^{(0)}(X)$$

La matriz Jacobiana (definición 3) $D\Phi(X^*)$ será el producto de las matrices Jacobianas $\Phi^{(k)}$ en el mismo orden en que se componen estas funciones.

Así, el aporte del paso del algoritmo a la propagación del error:

$$\Delta x^{(k+1)} = \alpha_{k+1} + D\Phi^{(k)}(x^{(k)})\Delta x^{(k)} \qquad \text{donde:} \qquad \alpha_{k+1} = E^{(k+1)}x^{(k+1)}$$

El error acumulado a lo largo de todo el algoritmo se aproxima como:

$$\Delta z \approx \Delta x^{(r+1)} \approx D\Phi^{(r)} \cdot \ldots \cdot D\Phi^{(0)}\Delta x + D\Phi(r) \cdot \ldots \cdot D\Phi^{(1)}\alpha_1 + \ldots + \alpha_{r+1}.$$

De acuerdo al ejemplo:

$\Phi(x1,x2,x3) = X1 + X2 + X3$, con largo de palabra p= 8.

$X1 = 0.23371258*10^{-4}$
$X2 = 0.33678429*10^{2}$
$X3 = -0.33677811*10^{2}$

$fl(X1 + fl(X2 + X3)) = 0.64137126*10^{-3}$ (Mejor Algoritmo)

$fl(fl(X1 + X2) + X3) = 0.64100000*10^{-3}$

$X1 + X2 + X3 = 0.641371258*10^{-3}$

$$X = X^{(0)} = \begin{bmatrix} X1 \\ X2 \\ X3 \end{bmatrix} \xrightarrow{\Phi^{(0)}} X^{(1)} = \begin{bmatrix} X1 \\ X2 + X3 \end{bmatrix} = \begin{bmatrix} U \\ V \end{bmatrix} \xrightarrow{\Phi^{(1)}} X^{(2)} = U + V = Z$$

Por lo cual:

$$D\,\Phi^{(0)} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 1 \end{bmatrix} \quad D\,\Phi^{(1)} = [1,1] \quad E^{(1)} = \begin{bmatrix} 0 & 0 \\ 0 & \varepsilon 1 \end{bmatrix} \longrightarrow \alpha^{(1)} = \begin{bmatrix} 0 \\ \varepsilon 1(X2 + X3) \end{bmatrix} \qquad E^{(2)} = \varepsilon 2 \quad \alpha^{(2)} = \varepsilon_2 * (X1 + X2 + X3)$$

La propagación total del error será:

$$\Delta Z = \begin{bmatrix} 1, 1, 1 \end{bmatrix} \begin{bmatrix} \Delta X1 \\ \Delta X2 \\ \Delta X3 \end{bmatrix} + \begin{bmatrix} 1, 1 \end{bmatrix} \begin{bmatrix} 0 \\ \varepsilon 1(X2 + X3) \end{bmatrix} + \varepsilon 2(X1 + X2 + X3)$$

Error de Cálculo — **Error de propagación del algoritmo**

Error relativo, acotado por: $|\dot{\varepsilon}| \leq |\varepsilon|$, donde $\varepsilon = \beta^{1-p}/2$ donde $\beta = 10$, $p = 8$

Algoritmo 1		Algoritmo 2
$\frac{\lvert X2 + X3\rvert\varepsilon 1}{\lvert X1 + X2 + X3\rvert} + \varepsilon 2$	$<$	$\frac{\lvert X1 + X2\rvert\varepsilon 3}{\lvert X1 + X2 + X3\rvert} + \varepsilon 4$
$\frac{\lvert X2 + X3\rvert\varepsilon 1}{\lvert X1 + X2 + X3\rvert}$	$<$	$\frac{\lvert X1 + X2\rvert\varepsilon 3}{\lvert X1 + X2 + X3\rvert}$

POR ESTE MOTIVO EL ALGORITMO 1, ES MEJOR QUE EL ALGORITMO 2

SER EFICIENTE EN EL ALGORITMO, ES MEJOR

Sistema Ecuaciones Lineales

Resolver sistema de Ecuaciones de la Forma:

$$\begin{cases} a_{11}x_1 + a_{12}x_2 + a_{13}x_3 + \cdots + a_{1n}x_n &= b_1 \\ a_{21}x_1 + a_{22}x_2 + a_{23}x_3 + \cdots + a_{2n}x_n &= b_2 \\ a_{31}x_1 + a_{32}x_2 + a_{33}x_3 + \cdots + a_{3n}x_n &= b_3 \\ \vdots & \vdots \\ a_{n1}x_1 + a_{n2}x_2 + a_{n3}x_3 + \cdots + a_{nn}x_n &= b_n \end{cases}$$

Donde hay n ecuaciones y n incógnitas :

$$\begin{pmatrix} a_{11} & a_{12} & a_{13} & \cdots & a_{1n} \\ a_{21} & a_{22} & a_{23} & \cdots & a_{2n} \\ a_{31} & a_{32} & a_{33} & \cdots & a_{3n} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ a_{11} & a_{12} & a_{13} & \cdots & a_{1n} \end{pmatrix} \begin{pmatrix} x_1 \\ x_2 \\ x_3 \\ \vdots \\ x_n \end{pmatrix} = \begin{pmatrix} b_1 \\ b_2 \\ b_3 \\ \vdots \\ b_n \end{pmatrix}$$

Esto se puede escribir de la forma :

$$AX = b$$

Se pueden dividir en dos grandes grupos:

Métodos Exactos: Algoritmos finitos que permiten obtener la solución del sistema de manera directa. Método de Gauss y una modificación de éste denominado método de Gauss-Jordan

Métodos Aproximados: Utilizan algoritmos iterativos e infinitos y que calculan las solución del sistema por aproximaciones sucesivas. Métodos de Richardson, Jacob y Gauss-Seidel

Métodos de Resolución Exacta:

Sistemas Simples:

Valores sólo en las diagonales

$$\begin{pmatrix} a_{11} & 0 & 0 & \cdots & 0 \\ 0 & a_{22} & 0 & \cdots & 0 \\ 0 & 0 & a_{33} & \cdots & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & 0 & \cdots & a_{1n} \end{pmatrix} \begin{pmatrix} x_1 \\ x_2 \\ x_3 \\ \vdots \\ x_n \end{pmatrix} = \begin{pmatrix} b_1 \\ b_2 \\ b_3 \\ \vdots \\ b_n \end{pmatrix} \Longrightarrow x = \begin{pmatrix} b_1/a_{11} \\ b_2/a_{22} \\ b_3/a_{33} \\ \vdots \\ b_n/a_{nn} \end{pmatrix}$$

Métodos de Resolución Exacta:

Sistemas Simples:

Diagonal Inferior:

$$\begin{pmatrix} a_{11} & 0 & 0 & \cdots & 0 \\ a_{21} & a_{22} & 0 & \cdots & 0 \\ a_{31} & a_{32} & a_{33} & \cdots & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ a_{11} & a_{12} & a_{13} & \cdots & a_{1n} \end{pmatrix} \begin{pmatrix} x_1 \\ x_2 \\ x_3 \\ \vdots \\ x_n \end{pmatrix} = \begin{pmatrix} b_1 \\ b_2 \\ b_3 \\ \vdots \\ b_n \end{pmatrix}$$

Sistema de Sustitución progresiva:

$$x_i = \left(b_i - \sum_{j=1}^{i-1} a_{ij} x_j \right) / a_{ii} \qquad \text{i = 1,2, …,n}$$

Métodos de Resolución Exacta:

Sistemas Simples:

Diagonal Superior:

$$\begin{pmatrix} a_{11} & a_{12} & a_{13} & \cdots & a_{1n} \\ 0 & a_{22} & a_{23} & \cdots & a_{2n} \\ 0 & 0 & a_{33} & \cdots & a_{3n} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & 0 & \cdots & a_{1n} \end{pmatrix} \begin{pmatrix} x_1 \\ x_2 \\ x_3 \\ \vdots \\ x_n \end{pmatrix} = \begin{pmatrix} b_1 \\ b_2 \\ b_3 \\ \vdots \\ b_n \end{pmatrix}$$

Sistema de Sustitución progresiva:

$$x_i = \left(b_i - \sum_{j=i+1}^{n} a_{ij} x_j \right) / a_{ii} \qquad \text{i = 1,2, …,n}$$

La Factorización LU:

Supongamos que *A* se puede factorizar como el producto de una matriz triangular inferior *L* con una matriz triangular superior *U*:

$$A = LU$$

Luego:

$$\mathbf{Ax=b} \longrightarrow \boldsymbol{LUx=b} \quad \boldsymbol{(1)}$$

Si denominamos *z* a la matriz columna de *n* filas resultado del producto de las matrices *Ux*, tenemos que la ecuación se puede reescribir del siguiente modo:

$$\boldsymbol{Lz=b} \quad \boldsymbol{(2)}$$

A partir de las ecuaciones (1) y (2), es posible plantear un algoritmo para resolver el sistema de ecuaciones empleando dos etapas:

Primero obtenemos *z* aplicando el algoritmo de **sustitución progresiva** en la ecuación (2). Posteriormente obtenemos los valores de *x* aplicando el algoritmo de **sustitución regresiva** a la ecuación

$$Ux = z$$

El análisis anterior nos muestra lo fácil que es resolver estos dos sistemas de ecuaciones triangulares y lo útil que resultaría disponer de un método que nos permitiera llevar a cabo la factorización A=LU. Si disponemos de una matriz A de n*n estamos interesados en encontrar aquellas matrices.

$$L = \begin{pmatrix} l_{11} & 0 & 0 & \cdots & 0 \\ l_{21} & l_{22} & 0 & \cdots & 0 \\ l_{31} & l_{32} & l_{33} & \cdots & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ l_{n1} & l_{n2} & l_{n3} & \cdots & l_{nn} \end{pmatrix} \quad U = \begin{pmatrix} u_{11} & u_{12} & u_{13} & \cdots & u_{14} \\ 0 & u_{22} & u_{23} & \cdots & u_{2n} \\ 0 & 0 & u_{33} & \cdots & u_{3n} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & 0 & \cdots & u_{nn} \end{pmatrix}$$

Cuando esto es posible, decimos que *A* tiene una **descomposición** *LU*

$$A = LU$$

Se puede ver que las ecuación anterior no determina de forma única a *L ni* a *U*. De hecho, para cada *i* podemos asignar un valor distinto de cero a l_{ii} o u_{ii} (aunque no ambos).

Por ejemplo, una elección simple es fijar lii=1 para i = 1,2,..n haciendo de esto modo que L sea una matriz triangular inferior unitaria. Otra elección es hacer U una matriz triangular superior unitaria (tomando uii=1 para cada i).

Para deducir un algoritmo que nos permita la factorización *LU* de *A* partiremos de la fórmula para la multiplicación de matrices:

$$a_{ij} = \sum_{s=1}^{n} l_{is} u_{sj} = \sum_{s=1}^{\min(i,j)} l_{is} u_{sj} \qquad (3)$$

en donde se ha valido del hecho de que l_{is}=0 para *s* >*i* y u_{sj}=0 para *s*>*j*.

En este proceso, cada paso determina una nueva fila de U y una nueva columna de L. En el paso k, podemos suponer que ya se calcularon las filas 1,2, .., k-1 de U al igual que las columnas 1,2,..,k-1 de L. Haciendo i=j=k en (3) se obtiene:

$$a_{kk} = l_{kk}u_{kk} \sum_{s=1}^{k-1} l_{ks}u_{sk} \qquad (4)$$

Si especificamos un valor para l_{kk} (o para u_{kk}), a partir de la ecuación (4) es posible determinar un valor para el otro término. Conocidas u_{kk} y l_{kk} y a partir de la ecuación (3) se pueden escribir las expresiones para la *k*-ésima fila (*i*=*k*) y para la *k*-ésima columna (*j*=*k*), respectivamente:

$$a_{kj} = l_{kk}u_{kj} + \sum_{s=1}^{k-1} l_{ks}u_{sj} \qquad (k+1 \leq j \leq n) \qquad (5)$$

$$a_{ik} = l_{ik}u_{kk} + \sum_{s=1}^{k-1} l_{is}u_{sk} \qquad (k+1 \leq i \leq n) \qquad (6)$$

En este proceso, cada paso determina una nueva fila de U y una nueva columna de L. En el paso k, podemos suponer que ya se calcularon las filas 1,2, .., k-1 de U al igual que las columnas 1,2,..,k-1 de L. Haciendo i=j=k en (3) se obtiene:

$$a_{kk} = l_{kk}u_{kk} \sum_{s=1}^{k-1} l_{ks}u_{sk} \qquad (4)$$

Si especificamos un valor para l_{kk} (o para u_{kk}), a partir de la ecuación (4) es posible determinar un valor para el otro término. Conocidas u_{kk} y l_{kk} y a partir de la ecuación (3) se pueden escribir las expresiones para la *k*-ésima fila (*i*=*k*) y para la *k*-ésima columna (*j*=*k*), respectivamente:

$$a_{kj} = l_{kk}u_{kj} + \sum_{s=1}^{k-1} l_{ks}u_{sj} \qquad (k+1 \leq j \leq n) \qquad (5)$$

$$a_{ik} = l_{ik}u_{kk} + \sum_{s=1}^{k-1} l_{is}u_{sk} \qquad (k+1 \leq i \leq n) \qquad (6)$$

Es decir, las ecuaciones (5) y (6) se pueden emplear para encontrar los elementos u_{kj} y l_{ik}. El algoritmo basado en el análisis anterior se denomina **factorización de Doolittle** cuando se toman los términos $l_{ii} = 1$ para $1 \leq i \leq n$ (L triangular inferior unitaria) y **factorización de Crout** cuando se toman los términos $u_{ii}=1$ (U triangular superior unitaria).

Una implementación en pseudocódigo del algoritmo para llevar a cabo la factorización *LU* se muestra

input n,(a_{ij})
for $k = 1, 2, \ldots, n$ **do**
  Especificar un valor para l_{kk} o u_{kk}.
  Calcular el otro término mediante:
  $l_{kk}u_{kk} = a_{kk} - \sum_{s=1}^{k-1} l_{ks}u_{sk}$

  for $j = k+1, k+2, \ldots, n$ **do**
    $u_{kj} \leftarrow \left(a_{kj} - \sum_{s=1}^{k-1} l_{ks}u_{sk}\right) / l_{kk}$
  end

  for $i = k+1, k+2, \ldots, n$ **do**
    $l_{jk} \leftarrow \left(a_{ik} - \sum_{s=1}^{k-1} l_{is}u_{sk}\right) / u_{kk}$
  end
end
output (l_{ij}),(u_{ij})

Es interesante notar que los ciclos que permiten el cómputo de la k-ésima fila de U y de la k-ésima columna de L se pueden llevar a cabo en paralelo, es decir, pueden evaluarse simultáneamente sobre dos procesadores, lo que redunda en un importante ahorro del tiempo de cálculo.

Ejemplo: Encuentre las factorizaciones de Doolittle y Crout de la matriz:

$$A = \begin{pmatrix} 60 & 30 & 20 \\ 30 & 20 & 15 \\ 20 & 15 & 12 \end{pmatrix}$$

La factorización de Doolittle es, a partir del algoritmo:

$$A = \begin{pmatrix} 1 & 0 & 0 \\ \frac{1}{2} & 1 & 0 \\ \frac{1}{3} & 1 & 1 \end{pmatrix} \begin{pmatrix} 60 & 30 & 20 \\ 0 & 5 & 5 \\ 0 & 0 & \frac{1}{3} \end{pmatrix} = LU$$

En vez de calcular la factorización de Crout directamente, la podemos obtener a partir de la factorización de Doolittle que acabamos de ver. Efectivamente, si tenemos en cuenta que la matriz *A* es simétrica, es posible comprobar que se cumple la relación:

$$A = LU = U^T L^T$$

por lo que la factorización de Crout resulta ser:

$$A = \begin{pmatrix} 60 & 0 & 0 \\ 30 & 5 & 0 \\ 20 & 5 & \frac{1}{3} \end{pmatrix} \begin{pmatrix} 1 & \frac{1}{2} & \frac{1}{3} \\ 0 & 1 & 1 \\ 0 & 0 & 1 \end{pmatrix} = U^T L^T$$

Eliminación Gauss Básica

Ilustraremos el método de Gauss aplicando el procedimiento a un sistema de cuatro ecuaciones con cuatro incógnitas:

$$\begin{pmatrix} 6 & -2 & 2 & 4 \\ 12 & -8 & 6 & 10 \\ 3 & -13 & 9 & 3 \\ -6 & 4 & 1 & -18 \end{pmatrix} \begin{pmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \end{pmatrix} = \begin{pmatrix} 12 \\ 34 \\ 27 \\ -38 \end{pmatrix}$$

Pasos:

1) Multiplicamos la primera ecuación 12/6 = 2 y la restamos a la segunda

2) Después multiplicamos la primera ecuación por 3/6 = ½ y la restamos a la

3) Multiplicamos la primera ecuación por -6/6 = -1 y la restamos a la cuarta

El número 6 es el **elemento pivote** de este primer paso y la primera fila, que no sufre modificación alguna, se denomina **fila pivote**. El sistema en estos momentos tiene el siguiente aspecto:

$$\begin{pmatrix} 6 & -2 & 2 & 4 \\ 0 & -4 & 2 & 2 \\ 0 & -12 & 8 & 1 \\ 0 & 2 & 3 & -14 \end{pmatrix} \begin{pmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \end{pmatrix} = \begin{pmatrix} 12 \\ 10 \\ 21 \\ -26 \end{pmatrix}$$

En el siguiente paso del proceso, la segunda fila se emplea como **fila pivote** y -4 como **elemento pivote**. Aplicamos del nuevo el proceso:

1) Multiplicamos la segunda fila por -12/-4 = 3 y la restamos de la tercera

2) Multiplicamos la segunda fila por 2/-4 = -1/2 y la restamos a la cuarta

Los multiplicadores son en esta ocasión 3 y -1/2 y el sistema de ecuaciones se reduce a:

$$\begin{pmatrix} 6 & -2 & 2 & 4 \\ 0 & -4 & 2 & 2 \\ 0 & 0 & 2 & -5 \\ 0 & 0 & 4 & -13 \end{pmatrix} \begin{pmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \end{pmatrix} = \begin{pmatrix} 12 \\ 10 \\ -9 \\ -21 \end{pmatrix}$$

El último paso consiste en multiplicar la tercera ecuación por 4/2 = 2 y restarla a la cuarta. El sistema resultante resulta ser:

$$\begin{pmatrix} 6 & -2 & 2 & 4 \\ 0 & -4 & 2 & 2 \\ 0 & 0 & 2 & -5 \\ 0 & 0 & 0 & -3 \end{pmatrix} \begin{pmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \end{pmatrix} = \begin{pmatrix} 12 \\ 10 \\ -9 \\ -3 \end{pmatrix}$$

El sistema resultante es triangular superior y equivalente al sistema original (las soluciones de ambos sistemas coinciden). Sin embargo, este sistema es fácilmente resoluble aplicando el algoritmo de **sustitución regresiva** explicada. La solución del sistema de ecuaciones resulta ser:

$$x = \begin{pmatrix} 1 \\ -3 \\ -2 \\ 1 \end{pmatrix}$$

Si colocamos los multiplicadores utilizados al transformar el sistema en una matriz triangular inferior unitaria (*L*) ocupando cada uno de ellos la posición del cero que contribuyó a producir, obtenemos la siguiente matriz:

$$L = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 2 & 1 & 0 & 0 \\ \frac{1}{2} & 3 & 1 & 0 \\ -1 & -\frac{1}{2} & 2 & 1 \end{pmatrix}$$

Por otra parte, la matriz triangular superior (*U*) formada por los coeficientes resultantes tras aplicar el algoritmo de Gauss, es:

$$L = \begin{pmatrix} 6 & -2 & 2 & 4 \\ 0 & -4 & 2 & 2 \\ 0 & 0 & 2 & -5 \\ 0 & 0 & 0 & -3 \end{pmatrix}$$

Estas dos matrices nos dan la factorización *LU* de la matriz inicial de coeficientes, *A*, expresada por la ecuación :

$$\begin{pmatrix} 6 & -2 & 2 & 4 \\ 12 & -8 & 6 & 10 \\ 3 & -13 & 9 & 3 \\ -6 & 4 & 1 & -18 \end{pmatrix} = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 2 & 1 & 0 & 0 \\ \frac{1}{2} & 3 & 1 & 0 \\ -1 & -\frac{1}{2} & 2 & 1 \end{pmatrix} \begin{pmatrix} 6 & -2 & 2 & 4 \\ 0 & -4 & 2 & 2 \\ 0 & 0 & 2 & -5 \\ 0 & 0 & 0 & -3 \end{pmatrix}$$

En la figura se muestra un algoritmo en pseudocódigo para llevar a la práctica el proceso básico de eliminación gauss que acabamos de describir. En este algoritmo se supone que todos los elementos pivote son distintos de cero.

```
input n,(a_ij )
for k = 1, 2, ..., n − 1 do
    for i = k + 1, k + 2, ..., n do
        z ← a_ik/a_kk
        a_ik ← 0
        for j = k + 1, k + 2, ..., n do
            a_ij ← a_ij − z a_kj
        end
    end
end
output (a_ij )
```

Método de Gauss-Jordan:

Como hemos visto, el método de Gauss transforma la matriz de coeficientes en una matriz triangular superior. El método de Gauss-Jordan continúa el proceso de transformación hasta obtener una matriz diagonal unitaria (aij=0 para cualquier i ≠ j)

Veamos el método de Gauss-Jordan siguiendo con el ejemplo empleado en el apartado anterior. Aplicando el método de Gauss habíamos llegado a la siguiente ecuación:

$$\begin{pmatrix} 6 & -2 & 2 & 4 \\ 0 & -4 & 2 & 2 \\ 0 & 0 & 2 & -5 \\ 0 & 0 & 0 & -3 \end{pmatrix} \begin{pmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \end{pmatrix} = \begin{pmatrix} 12 \\ 10 \\ -9 \\ -3 \end{pmatrix}$$

Ahora seguiremos un procedimiento similar al empleado en el método de Gauss.

Tomaremos como pivote el elemento a_{44}=-3; multiplicamos la cuarta ecuación por -3/4 y la restamos a la primera:

$$\begin{pmatrix} 6 & -2 & 2 & 0 \\ 0 & -4 & 2 & 2 \\ 0 & 0 & 2 & -5 \\ 0 & 0 & 0 & -3 \end{pmatrix} \begin{pmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \end{pmatrix} = \begin{pmatrix} 8 \\ 10 \\ -9 \\ -3 \end{pmatrix}$$

Realizamos la misma operación con la segunda y tercera fila, obteniendo:

$$\begin{pmatrix} 6 & -2 & 2 & 0 \\ 0 & -4 & 2 & 0 \\ 0 & 0 & 2 & 0 \\ 0 & 0 & 0 & -3 \end{pmatrix} \begin{pmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \end{pmatrix} = \begin{pmatrix} 8 \\ 8 \\ -4 \\ -3 \end{pmatrix}$$

Ahora tomamos como pivote el elemento a_{33}=2, multiplicamos la tercera ecuación por 2/2 = 1 y la restamos a la primera:

$$\begin{pmatrix} 6 & -2 & 0 & 0 \\ 0 & -4 & 2 & 0 \\ 0 & 0 & 2 & 0 \\ 0 & 0 & 0 & -3 \end{pmatrix} \begin{pmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \end{pmatrix} = \begin{pmatrix} 12 \\ 8 \\ -4 \\ -3 \end{pmatrix}$$

Repetimos la operación con la segunda fila:

$$\begin{pmatrix} 6 & -2 & 0 & 0 \\ 0 & -4 & 0 & 0 \\ 0 & 0 & 2 & 0 \\ 0 & 0 & 0 & -3 \end{pmatrix} \begin{pmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \end{pmatrix} = \begin{pmatrix} 12 \\ 12 \\ -4 \\ -3 \end{pmatrix}$$

Finalmente, tomamos como pivote a22=-4, multiplicamos la segunda ecuación por -2/-4 y la sumamos a la primera:

$$\begin{pmatrix} 6 & 0 & 0 & 0 \\ 0 & -4 & 0 & 0 \\ 0 & 0 & 2 & 0 \\ 0 & 0 & 0 & -3 \end{pmatrix} \begin{pmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \end{pmatrix} = \begin{pmatrix} 6 \\ 12 \\ -4 \\ -3 \end{pmatrix}$$

El sistema de ecuaciones anterior es, como hemos visto, *fácil* de resolver. Empleando la ecuación obtenemos las soluciones:

$$x = \begin{pmatrix} b_1/a_{11} \\ b_2/a_{22} \\ b_3/a_{33} \\ \vdots \\ b_n/a_{nn} \end{pmatrix} \longrightarrow x = \begin{pmatrix} 1 \\ -3 \\ -2 \\ 1 \end{pmatrix}$$

Pivoteo

Sin embargo, los algoritmos de Gauss y Gauss-Jordan que acabamos de describir pueden dar lugar a resultados erróneos fácilmente. Por ejemplo, analicemos el siguiente sistema de ecuaciones, en el que Є es un número muy pequeño pero distinto de cero:

$$\begin{pmatrix} \epsilon & 1 \\ 1 & 1 \end{pmatrix} \begin{pmatrix} x_1 \\ x_2 \end{pmatrix} = \begin{pmatrix} 1 \\ 2 \end{pmatrix}$$

Al aplicar el algoritmo gauss se obtiene el siguiente sistema triangular superior:

$$\begin{pmatrix} \epsilon & 1 \\ 0 & 1-\epsilon^{-1} \end{pmatrix} \begin{pmatrix} x_1 \\ x_2 \end{pmatrix} = \begin{pmatrix} 1 \\ 2-\epsilon^{-1} \end{pmatrix}$$

y la solución es:

$$\begin{cases} x_2 = \frac{2-\epsilon^{-1}}{1-\epsilon^{-1}} \\ x_1 = (1-x_2)\epsilon^{-1} \end{cases}$$

En el computador, si ε es suficientemente pequeño, los términos:

$$2 - \epsilon^{-1} \quad = \quad 1 - \epsilon^{-1}$$

se computarán como un mismo número, por lo que: $x_2 \approx 1$ y $x_1 \approx 0$

Sin embargo, la solución correcta es: $\begin{cases} x_1 = \frac{1}{1-\epsilon} \approx 1 \\ x_2 = \frac{1-2\epsilon}{1-\epsilon} \approx 1 \end{cases}$

Tenemos entonces que la solución calculada es exacta para x_2 pero extremadamente inexacta para x_1.

El problema anterior no radica en la *pequeñez* del término a_{ii}, sino en su *pequeñez relativa* respecto de los otros elementos de su fila. La conclusión que podemos extraer es que un buen algoritmo debe incluir el intercambio de ecuaciones cuando las circunstancias así lo exijan.

Un algoritmo que cumple este requisito es el denominado *eliminación gauss con pivoteo de filas escaladas*.

Métodos Iterativos:

El método de Gauss y sus variantes se conocen con el nombre de métodos **directos**: se ejecutan a través de un número finito de pasos y dan lugar a una solución que sería exacta si no fuese por los errores de redondeo.

Por contra, un método **indirecto** da lugar a una sucesión de vectores que idealmente converge a la solución. El cálculo se detiene cuando se cuenta con una solución aproximada con cierto grado de precisión especificado de antemano o después de cierto número de iteraciones. Los métodos indirectos son casi siempre **iterativos**: para obtener la sucesión mencionada se utiliza repetidamente un proceso sencillo.

Conceptos Básicos

En general, en todos los procesos iterativos para resolver el sistema $Ax=b$ se recurre a una cierta matriz Q, llamada matriz *descomposición*, escogida de tal forma que el problema original adopte la forma equivalente:

$$Qx = (Q-A)x+b \qquad (7)$$

La ecuación (7) sugiere un proceso iterativo que se concreta al escribir:

$$Qx^{(k)} = (Q-A)x^{(k-1)} + b \quad (k \geq 1) \qquad (8)$$

El vector inicial $x^{(0)}$ puede ser arbitrario, aunque si se dispone de un buen candidato como solución éste es el que se debe emplear. La aproximación inicial que se adopta, a no ser que se disponga de una mejor, es la idénticamente nula X1=X2=…=Xn=0. A partir de la ecuación (8) se puede calcular una sucesión de vectores $x^{(1)}$, $x^{(2)}$, ... Nuestro objetivo es escoger una matriz Q de manera que:

* se pueda calcular fácilmente la sucesión $[x^{(k)}]$.
* la sucesión $[x^{(k)}]$ converja rápidamente a la solución.

Como en todo método iterativo, deberemos especificar un criterio de convergencia δ y un número máximo de iteraciones *M*, para asegurar que el proceso se detiene si no se alcanza la convergencia. En este caso, puesto que *x* es un vector, emplearemos dos criterios de convergencia que se deberán satisfacer simultáneamente:

1. El módulo del vector diferencia, $\|x^{(k)} - x^{(k-1)}\|$ partido por el módulo del vector *x*, $\|x^{(k)}\|$ deberá ser menor que la convergencia deseada:

$$\mathrm{ABS}\left(\frac{\|x^{(k)} - x^{(k-1)}\|}{\|x^{(k)}\|}\right) \leq \delta$$

2. La diferencia relativa del mayor elemento en valor absoluto del vector $x^{(k)}$, $x_m = \mathrm{Máx}\{x_i\}$ deberá ser n (diez) veces menor que δ :

$$\mathrm{ABS}\left(\frac{x_m^{(k)} - x_m^{(k-1)}}{x_m^{(k)}}\right) \leq \frac{\delta}{10}$$

Método de Richardson

El **método de Richardson** toma como matriz *Q* la matriz identidad (*I*). En este caso la ecuación (8) queda en la forma:

$$Ix^{(k)} = (I-A)x^{(k-1)}+b = x^{(k-1)}+r^{(k-1)} \qquad (9)$$

En donde $r^{(k-1)}$ es el vector residual definido mediante $r^{(k-1)}=b-Ax^{(k-1)}$.
La matriz identidad es aquella matriz diagonal cuyos elementos no nulos son 1, es decir:

$$\begin{cases} a_{ij} = 0 & \text{si } i \neq j \\ a_{ij} = 1 & \text{si } i = j \end{cases}$$

y cumple que $IA = A$, para cualquier valor de A; es decir, es el elemento neutro del producto matricial. De acuerdo con esto, la ecuación (9) se puede escribir como:

$$\boldsymbol{x^{(k)} = x^{(k-1)} - Ax^{(k-1)} + b = x^{(k-1)} + r^{(k-1)}}$$

en donde un elemento cualquiera del vector $r^{(k-1)}$ vendrá dado por la expresión:

$$r_i^{(k-1)} = b_i - \sum_{i=1}^{n} a_{ij} x_j^{(k-1)}$$

se muestra un algoritmo para ejecutar la iteración de Richardson. Este método recibe también el nombre de método de relajación o método de los residuos

input n,$(a_{ij}\ )$,$(b_i\ )$,$(x_i\ )$,M
for $k = 1, 2, \ldots, M$ **do**
  for $i = 1, 2, \ldots, n$ **do**
    $r_i \leftarrow b_i - \sum_{j=1}^{n} a_{ij} xj$
  end
  for $i = 1, 2, \ldots, n$ **do**
    $x_i \leftarrow x_i + r_i$
  end
end
output k ,$(x_j\ )$,$(r_i\ )$

Método de Jacobi

En la **iteración de Jacobi**, se escoge una matriz *Q* que es diagonal y cuyos elementos diagonales son los mismos que los de la matriz *A*. La matriz *Q* toma la forma:

$$Q = \begin{pmatrix} a_{11} & 0 & 0 & \cdots & 0 \\ 0 & a_{22} & 0 & \cdots & 0 \\ 0 & 0 & a_{33} & \cdots & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & 0 & \cdots & a_{nn} \end{pmatrix}$$

y la ecuación general (8) se puede escribir como

$$\boldsymbol{Qx^{(k)} = (Q\text{-}A)x^{(k\text{-}1)} + b} \qquad (10)$$

Si denominamos *R* a la matriz *A*-*Q*:

$$R = \begin{pmatrix} 0 & a_{12} & a_{13} & \cdots & a_{1n} \\ a_{21} & 0 & a_{23} & \cdots & a_{2n} \\ a_{31} & a_{32} & 0 & \cdots & a_{3n} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ a_{n1} & a_{n2} & a_{n3} & \cdots & 0 \end{pmatrix}$$

la ecuación (10) se puede reescribir como:

$$\boldsymbol{Qx^{(k)} = -Rx^{(k-1)} + b}$$

El producto de la matriz *Q* por el vector columna $x^{(k)}$ será un vector columna. De modo análogo, el producto de la matriz *R* por el vector columna $x^{(k-1)}$ será también un vector columna. La expresión anterior, que es una ecuación vectorial, se puede expresar por *n* ecuaciones escalares (una para cada componente del vector). De este modo, podemos escribir, para un elemento *i* cualquiera y teniendo en cuenta que se trata de un producto matriz-vector:

$$\sum_{j=1}^{n} q_{ij} x_j^{(k)} = -\sum_{j=1}^{n} r_{ij} x_j^{(k-1)} + b_i$$

Si tenemos en cuenta que en la matriz *Q* todos los elementos fuera de la diagonal son cero, en el primer miembro el único término no nulo del sumatorio es el que contiene el elemento diagonal q_{ii}, que es precisamente a_{ii}. Más aún, los elementos de la diagonal de *R* son cero, por lo que podemos eliminar el término *i=j* en el sumatorio del segundo miembro. De acuerdo con lo dicho, la expresión anterior se puede reescribir como:

$$a_{ii}x_i^{(k)} = -\sum_{j=1,j\neq i}^{n} a_{ij}x_j^{(k-1)} + b_i$$

de donde despejando $x_i^{(k)}$ obtenemos:

$$x_i^{(k)} = \left(b_i - \sum_{j=1,j\neq i}^{n} a_{ij}x_j^{(k-1)} \right) / a_{ii}$$

que es la expresión que nos proporciona las nuevas componentes del vector $x^{(k)}$ en función de vector anterior $x^{(k\text{-}1)}$ en la iteración de Jacobi. Se muestra algoritmo de Jacobi.

input n,$(a_{ij}\ )$,$(b_i\ )$,$(x_i\ )$,M
for $k = 1, 2, \ldots, M$ **do**
  for $i = 1, 2, \ldots, n$ **do**
    $u_i \leftarrow \left(b_i - \sum_{j=1,j\neq i}^{n} a_{ij}xj \right) / a_{ii}$
  end
  for $i = 1, 2, \ldots, n$ **do**
    $x_i \leftarrow u_i$
  end
end
output k ,$(x_j\ )$

El método de Jacobi se basa en escribir el sistema de ecuaciones en la forma:

$$\begin{cases} x_1 &= (b_1 - a_{21}x_2 - a_{31}x_3 - \cdots - a_{n1}x_n)/a_{11} \\ x_2 &= (b_2 - a_{12}x_1 - a_{32}x_3 - \cdots - a_{n2}x_n)/a_{22} \\ \vdots & \quad \vdots \\ x_n &= (b_n - a_{1n}x_1 - a_{2n}x_2 - \cdots -)/a_{nn} \end{cases} \qquad (11)$$

Partimos de una aproximación inicial para las soluciones al sistema de ecuaciones y sustituimos estos valores en la ecuación (11). De esta forma, se genera una nueva aproximación a la solución del sistema, que en determinadas condiciones, es mejor que la aproximación inicial. Esta nueva aproximación se puede sustituir de nuevo en la parte derecha de la ecuación (11) y así sucesivamente hasta obtener la convergencia.

Método de Gauss-Seidel

La **iteración de Gauss-Seidel** se define al tomar *Q* como la parte triangular inferior de *A* incluyendo los elementos de la diagonal:

$$Q = \begin{pmatrix} a_{11} & 0 & 0 & \cdots & 0 \\ a_{21} & a_{22} & 0 & \cdots & 0 \\ a_{31} & a_{32} & a_{33} & \cdots & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ a_{n1} & a_{n2} & a_{n3} & \cdots & a_{nn} \end{pmatrix}$$

Si, como en el caso anterior, definimos la matriz *R*=*A*-*Q*

$$R = \begin{pmatrix} 0 & a_{12} & a_{13} & \cdots & a_{1n} \\ 0 & 0 & a_{23} & \cdots & a_{2n} \\ 0 & 0 & 0 & \cdots & a_{3n} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & 0 & \cdots & 0 \end{pmatrix}$$

y la ecuación (8) se puede escribir en la forma:

$$\boldsymbol{Qx^{(k)} = -Rx^{(k-1)} + b}$$

Un elemento cualquiera, *i*, del vector $Qx^{(k)}$ vendrá dado por la ecuación:

$$\sum_{j=1}^{n} a_{ij} x_j^{(k)} = -\sum_{j=1}^{n} a_{ij} x_j^{(k-1)} + b_i$$

Si tenemos en cuenta la peculiar forma de las matrices *Q* y *R*, resulta que todos los sumandos para los que *j* > *i* en la parte izquierda son nulos, mientras que en la parte derecha son nulos todos los sumandos para los que i ≤ j Podemos escribir entonces:

$$\sum_{j=1}^{i} a_{ij} x_j^{(k)} = -\sum_{j=i+1}^{n} a_{ij} x_j^{(k-1)} + b_i$$

$$a_{ii} x_i^{(k)} + \sum_{j=1}^{i-1} a_{ij} x_j^{(k)} = -\sum_{j=i+1}^{n} a_{ij} x_j^{(k-1)} + b_i$$

de donde despejando $x_i^{(k)}$, se obtiene:

$$x_i^{(k)} = \left(b_i - \sum_{j=1}^{i-1} a_{ij} x_j^{(k)} - \sum_{j=i+1}^{n} a_{ij} x_j^{(k-1)} \right) / a_{ii}$$

Obsérvese que en el método de Gauss-Seidel los valores actualizados de x_i sustituyen de inmediato a los valores anteriores, mientras que en el método de Jacobi todas las componentes nuevas del vector se calculan antes de llevar a cabo la sustitución. Por contra, en el método de Gauss-Seidel los cálculos deben llevarse a cabo por orden, ya que el nuevo valor x_i depende de los valores actualizados de x_1, x_2, ..., x_{i-1}.

Algoritmo para la iteración de Gauss-Seidel.

input n,$(a_{ij}\ )$,$(b_i\ )$,$(x_i\ )$,M
for $k = 1, 2, \ldots, M$ **do**
  for $i = 1, 2, \ldots, n$ **do**
    $u_i \leftarrow \left(b_i - \sum_{j=1, j\neq i}^{n} a_{ij} x_j\right) / a_{ii}$
  end
  for $i = 1, 2, \ldots, n$ **do**
    $x_i \leftarrow u_i$
  end
end
output k ,$(x_j\ )$

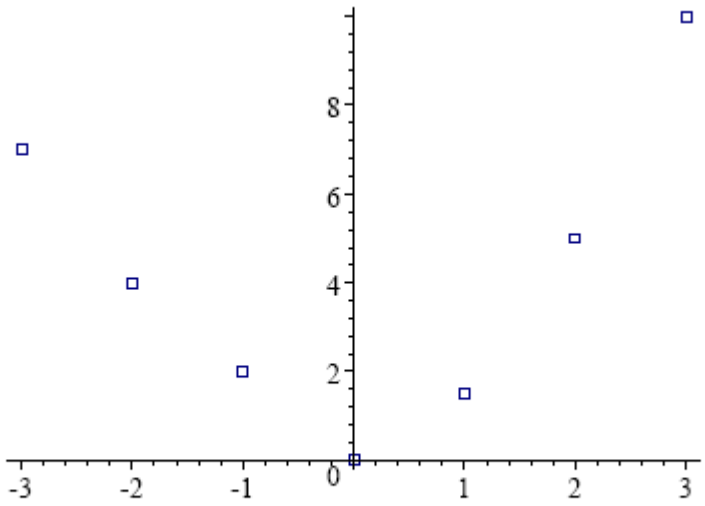
Interpolación

Mínimos Cuadrados

Un gráfico de puntos muestra evidencia de la existencia de una relación de tipo polinomios, para polinomios de grado n del tipo:

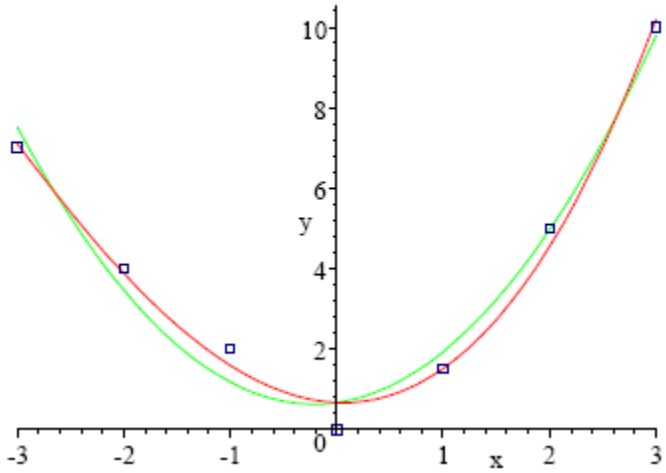
$$Y = A_0 + A_1X + A_2X^2 + \text{-----} + A_nX^n$$

Suponiendo el gráfico y sus datos siguientes:



$$(-3, 7), (-2, 4), (-1, 2), (0, 0), (1, 1.5), (2, 5), (3, 10)$$

El gráfico m muestra un ajuste mediante polinomio de grado (verde) y de grado 3 (rojo).



Para el caso grado n:

$$A_0 * n + A_1 \sum X + A_2 \sum X^2 + \text{-----} + A_n \sum X^n = \sum Y$$

$$A_0 * \sum X + A_1 \sum X^2 + A_2 \sum X^3 + \text{-----} + A_n \sum X^{n+1} = \sum YX$$

*

*

*

*

*

$$A_0 * \sum X^n + A_1 \sum X^{1+n} + A_2 \sum X^{2+n} + \text{-----} + A_n \sum X^{n+n} = \sum YX^n$$

Polinomios de interpolación de Lagrange

Un polinomio de interpolación de Lagrange, *p*, se define en la forma:

$$p(x) = y_0\ell_0(x) + y_1\ell_1(x) + \cdots + y_n\ell_n(x) = \sum_{k=0}^{n} y_k\ell_k(x) \qquad (12)$$

en donde *l0, l1, l2,.., ln* son polinomios que dependen sólo de los nodos tabulados X1,X2,..,Xn, pero no de las ordenadas Y1, Y2, …, Yn. La fórmula general del polinomio *li* es:

$$\ell_i(x) = \prod_{j=0, j\neq i}^{n} \frac{x - x_j}{x_i - x_j} \qquad (13)$$

Polinomios de interpolación de Spline

Una *función spline* está formada por varios polinomios, cada uno definido sobre un sub intervalo, que se unen entre sí obedeciendo a ciertas condiciones de continuidad.

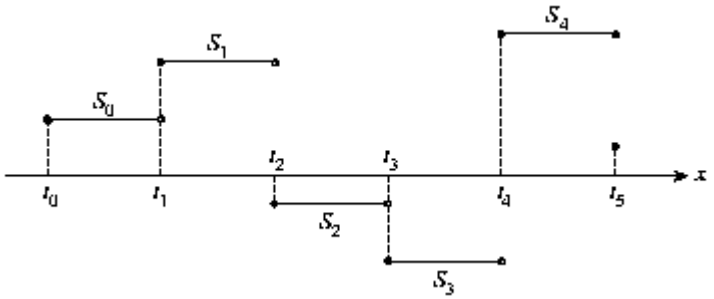
Supongamos que disponemos de $n+1$ puntos, a los que denominaremos *nudos*, tales que $t0 < t1 < \ldots < tn$. Supongamos además que se ha fijado un entero $0 \leq k$. Decimos entonces que una **función spline de grado *k*** con nudos en $t0 < t1 < \ldots < tn$ es una función *S* que satisface las condiciones:

(1) en cada intervalo $(ti-1, ti)$, *S* es un polinomio de grado menor o igual a *k*.

(2) *S* tiene una derivada de orden $(k-1)$ continua en $[t0,tn]$

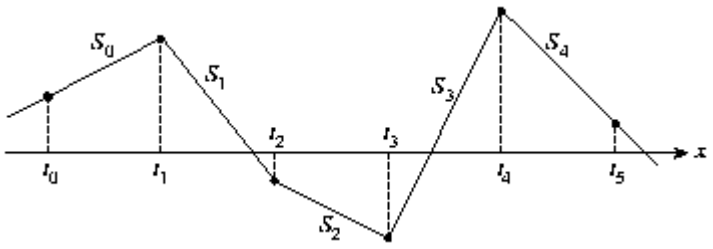
Los splines de grado 0 son funciones constantes por zonas. Una forma explícita de presentar un spline de grado 0 es la siguiente:

$$S(x) = \begin{cases} S_0(x) = c_0 & x \in [t_0, t_1) \\ S_1(x) = c_1 & x \in [t_1, t_2) \\ \vdots & \vdots \\ S_{n-1}(x) = c_{n-1} & x \in [t_{n-1}, t_n) \end{cases}$$



Los intervalos (ti-1, ti) no se intersectan entre sí, por lo que no hay ambigüedad en la definición de la función en los nudos. Un spline de grado 1 se puede definir por:

$$S(x) = \begin{cases} S_0(x) = a_0 x + b_0 & x \in [t_0, t_1) \\ S_1(x) = a_1 x + b_1 & x \in [t_1, t_2) \\ \vdots & \vdots \\ S_{n-1}(x) = a_{n-1} x + b_{n-1} & x \in [t_{n-1}, t_n) \end{cases}$$



Splines Cúbicos

El spline cúbico (k=3) es el spline más empleado, debido a que proporciona un excelente ajuste a los puntos tabulados y su cálculo no es excesivamente complejo. Sobre cada intervalo [t0,t1][t1,t2]….[tn-1,tn] , S está definido por un polinomio cúbico diferente. Sea S_i el polinomio cúbico que representa a S en el intervalo $[t_i,t_{i+1}]$, por tanto:

$$S(x) = \begin{cases} S_0(x) & x \in [t_0, t_1) \\ S_1(x) & x \in [t_1, t_2) \\ \vdots & \vdots \\ S_{n-1}(x) & x \in [t_{n-1}, t_n) \end{cases}$$

Los polinomios S_{i-1} y S_i interpolan el mismo valor en el punto t_i, es decir, se cumple:

$$S_{i-1}(t_i) = y_i = S_i(t_i) \qquad 1 \le i \le n$$

por lo que se garantiza que S es continuo en todo el intervalo. Además, se supone que S' y S'' son continuas, condición que se emplea en la deducción de una expresión para la función del spline cúbico.

Aplicando las condiciones de continuidad del spline *S* y de las derivadas primera *S*' y segunda *S*'', es posible encontrar la expresión analítica del spline. No vamos a obtener esta expresión, ya que su demostración queda fuera del ámbito de estos apuntes. Simplemente diremos que la expresión resultante es:

$$\begin{aligned} S_i(x) &= \frac{z_i}{6h_i}(t_{i+1}-x)^3 + \frac{z_{i+1}}{6h_i}(x-t_i)^3 \\ &\quad + \left(\frac{y_{i+1}}{h_i} + \frac{z_{i+1}h_i}{6}\right)(x-t_i) + \left(\frac{y_i}{h_i} - \frac{z_i h_i}{6}\right)(t_{i+1}-x) \end{aligned}$$

En la expresión anterior, $h_i = t_{i+1} - t_i$ y Z0, Z1 ,…,Zn son incógnitas. Para determinar sus valores, utilizamos las condiciones de continuidad que deben cumplir estas funciones. El resultado (que tampoco vamos a demostrar) es:

$$h_{i-1}z_{i-1} + 2(h_i + h_{i-1})z_i + h_i z_{i+1} = \frac{6}{h_{i-1}}(y_{i+1} - y_i) - \frac{6}{h_{i-1}}(y_i - y_{i-1})$$

La ecuación anterior con i = 1,2, … ,n-1 genera un sistema de *n*-1ecuaciones lineales con *n*+1 incógnitas Z0, Z1, …, Zn. Podemos elegir z_0 y z_1 de forma arbitraria y resolver el sistema de ecuaciones resultante para obtener los valores de Z1, …, Zn-1

Una elección especialmente adecuada es hacer $z_0=z_1=0$. La función spline resultante se denomina *spline cúbico natural* y el sistema de ecuaciones lineal expresado en forma matricial es:

$$\begin{pmatrix} u_1 & h_1 & & & & \\ h_1 & u_2 & h_2 & & & \\ & h_2 & u_3 & h_3 & & \\ & & \ddots & \ddots & \ddots & \\ & & & h_{n-3} & u_{n-2} & h_{n-2} \\ & & & & h_{n-2} & u_{n-1} \end{pmatrix} \begin{pmatrix} z_1 \\ z_2 \\ z_3 \\ \vdots \\ z_{n-2} \\ z_{n-3} \end{pmatrix} = \begin{pmatrix} \nu_1 \\ \nu_2 \\ \nu_3 \\ \vdots \\ \nu_{n-2} \\ \nu_{n-1} \end{pmatrix}$$

En donde:

$$h_i = t_{i+1} - t_i$$

$$u_i = 2(h_i - h_{i-1}) - \frac{h^2_{i-1}}{u_{i-1}}$$

$$b_i = \frac{6}{h_i}(y_{i+1} - y_i)$$

$$v_i = b_i - b_{i-1} - \frac{(h_{i-1})(v_{i-1})}{u_{i-1}}$$

Algoritmo para encontrar los coeficientes z_i de un spline cúbico.

input n,$(t_i\ )$,$(y_i\ )$
for $i = 0, 1, \ldots, n-1$ **do**
    $h_i \leftarrow t_{i+1} - t_i$
    $b_i \leftarrow 6(y_{i+1} - y_i)/h_i$
end

$u_1 \leftarrow 2(h_0 + h_1)$
$\nu_1 \leftarrow b_1 - b_0$
for $i = 2, 3, \ldots, n-1$ **do**
    $u_i \leftarrow 2(h_i + h_{i-1}) - h_{i-1}^2/u_{i-1}$
    $\nu_i \leftarrow b_i - b_{i-1} - h_{i-1}\nu_{i-1}/u_{i-1}$
end

$z_n \leftarrow 0$
for $i = n-1, n-2, \ldots, 1$ **do**
    $z_i \leftarrow (\nu_i - h_i z_{i+1})/u_i$
end
$z_0 \leftarrow 0$
output $(z_i\ )$

Este sistema de ecuaciones, que es tridiagonal, se puede resolver mediante eliminación gaussi sin pivoteo como se muestra en el algoritmo. El código acepta como entrada un conjunto de nodos (t_i) y el conjunto de los valores de la función correspondiente (y_i) y produce un vector con los vectores z_i. Por último, el valor del spline *S* en un punto *x* cualquiera interpolado se puede calcular de forma eficiente empleando la siguiente expresión:

$$S_i(x) = y_i + (x - t_i)\left[C_i + (x - t_i)\left[B_i + (x - t_i)A_i\right]\right]$$

En donde:

Ai = (1/6hi)(Zi+1 – Zi)

Bi = Zi/2

Ci = (-1hi/6)Zi+1 – (hi/3)Zi + (1/hi)(Yi+1 – Yi)

Cálculo Raíces

Muchos problemas de ciencia e ingeniería conducen a determinar las raíces de una ecuación de la forma f(x)=0.

Donde f(x) es una ecuación diferenciable en un intervalo de interés.

Para una ecuación del tipo: $aX^2 + bX + c = 0$, es conocida la solución:

$$X = \frac{-b \quad (b^2 - 4ac)^{(1/2)}}{2a}$$

También hay fórmula para ecuaciones de grado 3 y ecuaciones de grado 4.

Pero para un polinomio de grado 5 o superior u otro tipo de funciones no hay fórmulas

Existen una serie de reglas que pueden ayudar a determinar las raíces de una ecuación:

a. El teorema de Bolzano, que establece que si una función continua, $f(x)$, toma en los extremos del intervalo $[a,b]$ valores de signo opuesto, entonces la función admite, al menos, una raíz en dicho intervalo.

b. En el caso en que $f(x)$ sea una función algebraica (polinómica) de grado n y coeficientes reales, podemos afirmar que tendrá n raíces reales o complejas.

c. La propiedad más importante que verifican las raíces racionales de una ecuación algebraica establece que si p/q es una raíz racional de la ecuación de coeficientes enteros:

$$a_0 + a_1 x + a_2 x^2 + \ldots + a_n x^n = 0 \quad (a_i \in \mathcal{Z})$$

entonces el denominador *q* divide al coeficientes a_n y el numerador *p* divide al término independiente a_0.
Ejemplo: Pretendemos calcular las raíces racionales de la ecuación:
$3x^3 + 3x^2 - x - 1 = 0$

Primero es necesario efectuar un cambio de variable $x = y/3$:

$$3\frac{y^3}{3^3} + 3\frac{y^2}{3^2} - \frac{y}{3} - 1 = 0$$

y después multiplicamos por 3^2: y3 + 3y2 -3y -9 = 0

con lo que los candidatos a raíz del polinomio son:

$$y = 9; \quad y = -9;$$
$$y = 3; \quad y = -3;$$
$$y = 1; \quad y = -1$$

Sustituyendo en la ecuación, obtenemos que la única raíz real es $y = -3$, es decir,

$$x = \frac{-3}{3} = -1$$

(que es además la única raíz racional de la ecuación). Lógicamente, este método es muy poco potente, por lo que sólo nos puede servir a modo de orientación.

La mayoría de los métodos utilizados para el cálculo de las raíces de una ecuación son iterativos y se basan en modelos de aproximaciones sucesivas. Estos métodos trabajan del siguiente modo: a partir de una primera aproximación al valor de la raíz, determinamos una aproximación mejor aplicando una determinada regla de cálculo y así sucesivamente hasta que se determine el valor de la raíz con el grado de aproximación deseado.

Método de la Bisección

Es el método más elemental y antiguo para determinar las raíces de una ecuación. Está basado directamente en el teorema de Bolzano explicado con anterioridad. Consiste en partir de un intervalo $[x_0,x_1]$tal que $f(x_0)f(x_1) < 0$, por lo que sabemos que existe, al menos, una raíz real. A partir de este punto se va reduciendo el intervalo sucesivamente hasta hacerlo tan pequeño como exija la precisión que hayamos decidido emplear.

Tarea: Escriba un Algoritmo

Inicialmente, es necesario suministrar al programa el número máximo de iteraciones MaxIter, la tolerancia δ , que representa las cifras significativas con las que queremos obtener la solución y dos valores de la variable independiente, x_0 y x_1, tales que cumplan la relación $f(x_0)f(x_1) < 0$. Una vez que se comprueba que el intervalo de partida es adecuado, lo dividimos en dos subintervalos tales que $[X_0,(X_0+X_1)/2]$ y $[(X_0+X_1)/2, X_1]$ y determinamos en qué subintervalo se encuentra la raíz (comprobando de nuevo el producto de las funciones).

Repetimos el proceso hasta alcanzar la convergencia (hasta que $\Delta \leq \delta$) o bien hasta que se excede el número de iteraciones permitidas (Iter > MaxIter), en cuyo caso es necesario imprimir un mensaje de error indicando que el método no converge.

Algunas explicaciones adicionales sobre el método:

El punto medio del intervalo se calcula como $Xm = X_0 + (X_1 - X_0)/2$ en lugar de emplear $Xm = (X_1 + X_0)/2$. Se sigue de este modo una estrategia general al efectuar cálculos numéricos que indica que es mejor calcular una cantidad añadiendo un pequeño término de corrección a una aproximación obtenida previamente. Por ejemplo, en un computador de precisión limitada, existen valores de x_0 y x_1 para los cuales x_m calculado mediante $Xm = (X_1 + X_0)/2$ se sale del intervalo $[X_0, X_1]$.

La convergencia Δ se calcula mediante la expresión $\Delta = ABS((X_1 - X_0)/X_1)$ De este modo, el término Δ, representa el número de cifras significativas con las que obtenemos el resultado.

Método de las aproximaciones Sucesivas

Dada la ecuación $f(x) = 0$, el método de las aproximaciones sucesivas reemplaza esta ecuación por una equivalente, $x=g(x)$, definida en la forma $g(x)=f(x)+x$. Para encontrar la solución, partimos de un valor inicial x_0 y calculamos una nueva aproximación $x_1=g(x_0)$. Reemplazamos el nuevo valor obtenido y repetimos el proceso. Esto da lugar a una sucesión de valores {X_0, X_1, …, Xn}, que si converge, tendrá como límite la solución del problema.

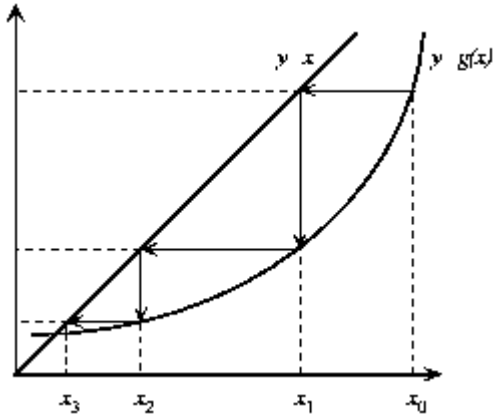


figura del método de aproximaciones sucesivas

En la figura se representa la interpretación geométrica del método. Partimos de un punto inicial x_0 y calculamos $y = g(x_0)$. La intersección de esta solución con la recta $y=x$ nos dará un nuevo valor x_1 más próximo a la solución final.

Sin embargo, el método puede divergir fácilmente. Es fácil comprobar que el método sólo podrá converger si la derivada $g'(x)$ es menor en valor absoluto que la unidad (que es la pendiente de la recta definida por $y=x$). Un ejemplo de este caso se muestra en la figura siguiente.

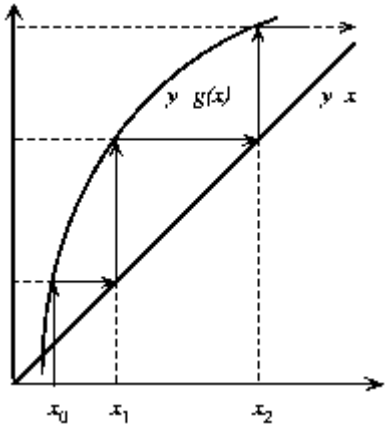


Ilustración gráfica de que el método de las aproximaciones sucesivas diverge si la derivada $g'(x) > 1$

Esta condición, que *a priori* puede considerarse una severa restricción del método, puede obviarse fácilmente. Para ello basta elegir la función $g(x)$ del siguiente modo:

$$g(x) = x + \alpha f(x)$$

de forma que tomando un valor de α adecuado, siempre podemos hacer que $g(x)$ cumpla la condición de la derivada.

Método de Newton

Este método parte de una aproximación inicial x_0 y obtiene una aproximación mejor, x_1, dada por la fórmula:

$$X_1 = X_0 - f(X_0)/f'(X_0) \qquad [14]$$

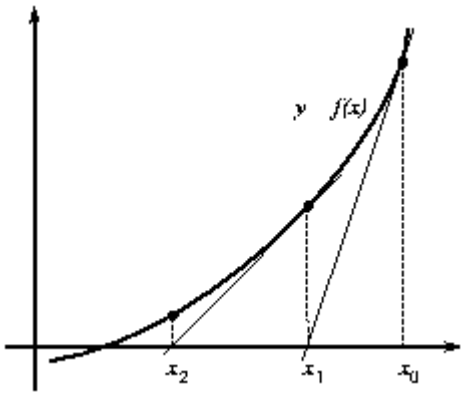
La expresión anterior puede derivarse a partir de un desarrollo en serie de Taylor. Efectivamente, sea *r* un cero de *f* y sea *x* una aproximación a *r* tal que *r*=*x*+*h*. Si *f*'' existe y es continua, por el teorema de Taylor tenemos:

$$0 = f(r) = f(X+h) = f(X) + hf'(X) + O(h^2) \qquad [15]$$

en donde *h*=*r*-*x*. Si *x* está próximo a *r* (es decir *h* es pequeña), es razonable ignorar el término $O(h^2)$:

$$0 = f(X) + hf'(X) \rightarrow h = - f(X)/f'(X) \qquad [16]$$

A partir de la ecuación (16) y teniendo en cuenta que *r*=*x*+*h* es fácil derivar la ecuación (14).



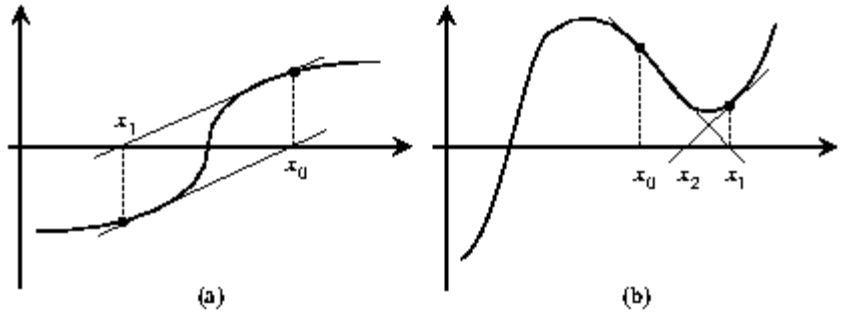
Interpretación geométrica del método de Newton.

El método de Newton tiene una interpretación geométrica sencilla, como se puede apreciar del análisis de la figura. De hecho, el método de Newton consiste en una **linealización** de la función, es decir, *f* se reemplaza por una recta tal que contiene al punto $(x_0,f(x_0))$ y cuya pendiente coincide con la derivada de la función en el punto, $f'(x_0)$. La nueva aproximación a la raíz, x_1, se obtiene de la intersección de la función linear con el eje *X* de ordenadas.

Veamos como podemos obtener la ecuación (14) a partir de lo dicho en el párrafo anterior. La ecuación de la recta que pasa por el punto $(x_0,f(x_0))$ y de pendiente $f'(x_0)$ es:

$$y - f(x_0) = f'(x_0)(x-x_0)$$

de donde, haciendo y=0 y despejando x obtenemos la ecuación de Newton-Raphson (14).



Dos situaciones en las que el método de Newton no funciona adecuadamente: (a) el método no alcanza la convergencia y (b) el método converge hacia un punto que no es un cero de la ecuación.

El método de Newton es muy rápido y eficiente ya que la convergencia es de tipo cuadrático (el número de cifras significativas se duplica en cada iteración). Sin embargo, la convergencia depende en gran medida de la forma que adopta la función en las proximidades del punto de iteración. En la figura se muestran dos situaciones en las que este método no es capaz de alcanzar la convergencia (figura a) o bien converge hacia un punto que no es un cero de la ecuación (figura b).

Método de la Secante

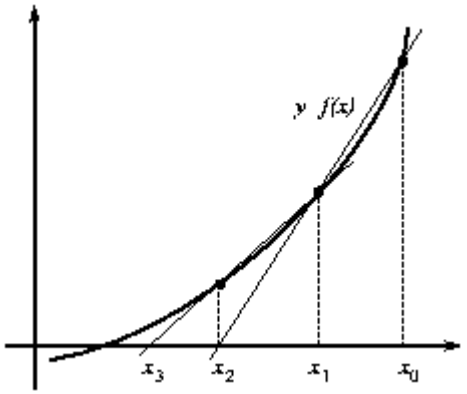
El principal inconveniente del método de Newton estriba en que requiere conocer el valor de la primera derivada de la función en el punto. Sin embargo, la forma funcional de *f*(*x*) dificulta en ocasiones el cálculo de la derivada. En estos casos es más útil emplear el método de la secante.

El método de la secante parte de dos puntos (y no sólo uno como el método de Newton) y estima la tangente (es decir, la pendiente de la recta) por una aproximación de acuerdo con la expresión:

$$f'(x_0) = \frac{f(x_1) - f(x_0)}{x_1 - x_0} \qquad [17]$$

Sustituyendo esta expresión en la ecuación (14) del método de Newton, obtenemos la expresión del método de la secante que nos proporciona el siguiente punto de iteración:

$$x_2 = x_0 - \frac{x_1 - x_0}{f(x_1) - f(x_0)} f(x_0) \qquad [18]$$



Representación geométrica del método de la secante.

En la siguiente iteración, emplearemos los puntos x_1 y $x_2$para estimar un nuevo punto más próximo a la raíz de acuerdo con la ecuación (18). En la figura se representa geométricamente este método.

En general, el método de la secante presenta las mismas ventajas y limitaciones que el método de Newton-Raphson explicado anteriormente.

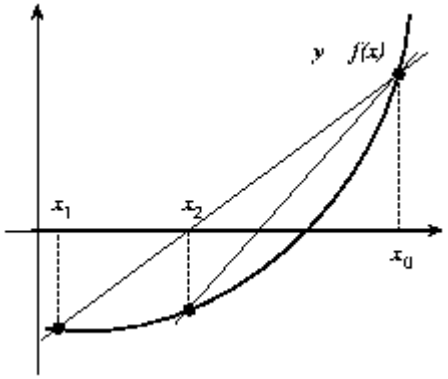
Método de Steffensen

El método de Steffensen presenta una convergencia rápida y no requiere, como en el caso del método de la secante, la evaluación de derivada alguna. Presenta además, la ventaja adicional de que el proceso de iteración sólo necesita un punto inicial. Este método calcula el siguiente punto de iteración a partir de la expresión:

$$x_{n+1} = x_n - \frac{[f(x_n)]^2}{f(x_n + f(x_n)) - f(x_n)} \qquad [19]$$

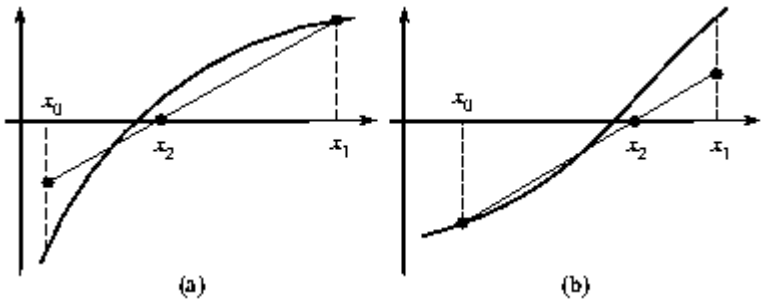
Método de la falsa posición

El método de la falsa posición pretende conjugar la seguridad del método de la bisección con la rapidez del método de la secante. Este método, como en el método de la bisección, parte de dos puntos que rodean a la raíz $f(x) = 0$, es decir, dos puntos x_0 y $x_1$tales que $f(x_0)f(x_1) < 0$. La siguiente aproximación, x_2, se calcula como la intersección con el eje X de la recta que une ambos puntos (empleando la ecuación (18) del método de la secante). La asignación del nuevo intervalo de búsqueda se realiza como en el método de la bisección: entre ambos intervalos, $[x_0,x_2]$ y $[x_2,x_1]$, se toma aquel que cumpla $f(x)f(x_2) < 0$. En la figura se representa geométricamente este método.



Representación geométrica del método de la falsa posición.

La elección *guiada* del intervalo representa una ventaja respecto al método de la secante ya que inhibe la posibilidad de una divergencia del método. Por otra parte y respecto al método de la bisección, mejora notablemente la elección del intervalo (ya que no se limita a partir el intervalo por la mitad).



Modificación del método de la falsa posición propuesta por Hamming. La aproximación a la raíz se toma a partir del punto de intersección con el eje *X* de la recta que une los puntos (x_0,$f(x_0)/2$) y (x_1,$f(x_1)$) si la función es convexa en el intervalo (figura a) o bien a partir de la recta que une los puntos (x_0,$f(x_0)$) y (x_1, $f(x_1)/2$) si la función es cóncava en el intervalo (figura b).

Sin embargo, el método de la falsa posición tiene una convergencia muy lenta hacia la solución. Efectivamente, una vez iniciado el proceso iterativo, uno de los extremos del intervalo tiende a no modificarse (ver figura método). Para obviar este problema, se ha propuesto una modificación del método, denominada método de Hamming. Según este método, la aproximación a una raíz se encuentra a partir de la determinación del punto de intersección con el eje *X* de la recta que une los puntos (x_0,$f(x_0)/2$) y (x_1,$f(x_1)$) si la función es convexa en el intervalo o bien a partir de la recta que une los puntos (x_0,$f(x_0)$) y (x_1, $f(x_1)/2$) si la función es cóncava en el intervalo. En la figura (partes (a) y (b)) se representa gráficamente el método de Hamming.

Como se ha comentado, el método de Hamming requiere determinar la concavidad o convexidad de la función en el intervalo de iteración. Un método relativamente sencillo para determinar la curvatura de la función consiste en evaluar la función en el punto medio del intervalo, $f(x_m)$ (en donde x_m se calcula como en el método de la bisección) y comparar este valor con la media de los valores de la función en los extremos del intervalo,

$$\overline{f} = (f(X_0) + f(X_1))/2$$

Tenemos entonces que:

$$f(x_m) \begin{cases} \leq \overline{f} & \text{si la función es cóncava} \\ \geq \overline{f} & \text{si la función es convexa} \end{cases}$$

Serie de Taylor y Maclaurin

Si f tiene una representación en serie en torno de a, o sea sí.

$$f(x) = \sum_{n=0}^{\infty} Cn(x-a)^n \quad I\, x - a\, I < R$$

Entonces los coeficientes son: $Cn = \frac{f^{(n)}(a)}{n!}$

Sustituyendo Cn en la serie para f, se tiene la serie de Taylor:

$$f(x) = \sum_{n=0}^{\infty} \frac{f^{(n)}(a)}{n!}(x-a)^n = f(a) + \frac{f\prime(a)}{1!}(x-a) + \frac{f\prime\prime(a)}{2!}(x-a)^2 + \frac{f\prime\prime\prime(a)}{3!}(x-a)^3 + \ldots$$

En el caso especial donde a = 0, se tiene la serie de Maclaurin:

$$f(x) = \sum_{n=0}^{\infty} \frac{f^{(n)}(0)}{n!}x^n = f(0) + \frac{f\prime(0)}{1!}x + \frac{f\prime\prime(0)}{2!}x^2 + \frac{f\prime\prime\prime(0)}{3!}x^3 + \ldots$$

Funciones pueden ser representadas por series de potencia en torno a a son llamadas *analíticas en a*. Funciones analíticas son infinitamente diferenciables en a; eso es que tiene derivadas de todo orden en a. Sin embargo, no todas las funciones infinitamente diferenciables son analíticas.

Las sumas parciales de Taylor son:

$$T_n(x) = \sum_{i=0}^{n} \frac{f^{(i)}(a)}{i!}(x-a)^i = f(a) + \frac{f\prime(a)}{1!}(x-a) + \frac{f\prime\prime(a)}{2!}(x-a)^2 + \ldots + \frac{f^{(n)}(a)}{n!}(x-a)^n$$

Tn en un polinomio de grado n llamado polinomio de *Taylor de grado n* de f en a.

Teorema

Si $f(x) = T_n(x) + R_n(x), \quad$ y $\lim_{n\to\infty} R_n(x) = 0$ Para Ix – aI < R

Entonces f es igual a su serie de *Taylor* serie en el intervalo Ix - aI < R; esto es, f es analítica en a.

Teorema (fórmula de Taylor)

Si f tiene n+1 derivadas en el intervalo I que contiene el número a, entonces para x en I hay un número z estrictamente entre x y a tak que el resto puede ser expresado como:

$$R_n(x) = \frac{f^{(n+1)}(z)}{(n+1)!}(x-a)^{n+1}$$

Para el caso especial n=0, se tiene que:

$$f(b) = f(a) + f'(c)(b-a)$$

Es el teorema del valor medio

Series importantes de Macluarin

Serie de Maclaurin	Intervalo de Convergencia
$\frac{1}{1-x} = \sum_{n=0}^{\infty} x^n = 1 + x + x^2 + x^3 + \ldots$	$(-1, 1)$
$e^x = \sum_{n=0}^{\infty} \frac{x^n}{n!} = 1 + \frac{x}{1!} + \frac{x^2}{2!} + \frac{x^3}{3!} + \ldots$	$(-\infty, \infty)$
$\sin x = \sum_{n=0}^{\infty} (-1)^n \frac{x^{2n+1}}{(2n+1)!} = x - \frac{x^3}{3!} + \frac{x^5}{5!} - \frac{x^7}{7!} + \ldots$	$(-\infty, \infty)$
$\cos x = \sum_{n=0}^{\infty} (-1)^n \frac{x^{2n}}{(2n)!} = 1 - \frac{x^2}{2!} + \frac{x^4}{4!} - \frac{x^6}{6!} + \ldots$	$(-\infty, \infty)$
$\tan^{-1} x = \sum_{n=0}^{\infty} (-1)^n \frac{x^{2n+1}}{2n+1} = x - \frac{x^3}{3} + \frac{x^5}{5} - \frac{x^7}{7} + \ldots$	$[-1, 1]$

Ecuaciones Diferenciales Ordinarias

Las Ecuaciones Diferenciales son algunas de las herramientas matemáticas más usadas en el modelamiento de fenómenos físicos. Hay varios métodos para resolver numéricamente el problema de primer orden con condición inicial.

$$Y' = f(x,y)$$
$$Y(x_0) = y_0 \qquad [20]$$

Los métodos que estudiaremos se generalizan fácilmente al caso de sistemas de primer orden, del tipo:

$Y'1 = f1(x,y1,y2, \ldots, yn)$, con condiciones iniciales $Y1(x0) = \eta 1$,
$Y'2 = f2(x,y1,y2, \ldots, yn)$, con condiciones iniciales $Y2(x0) = \eta 2$,

[21]

$Y'n = fn(x,y1,y2, \ldots, yn)$, con condiciones iniciales $Yn(x0) = \eta n$,

Y por lo tanto permiten resolver el problema de orden n, mayor que uno, con condiciones iniciales

$Y^{(n)} = f(x,y,y', \ldots, y^{(n-1)})$, con condiciones iniciales

$Y(x0) = \mu_0$,

$Y`(x0) = \mu_1$,

$Y^{(n-1)}(x0) = \mu_{n-1}$,

Ya que esto se puede reducir al problema anterior mediante el cambio de variables.

$Yk = Y`k-1$ Para todo $k = 2,\ldots,n$, $Y1 = Y$

Con lo que se obtiene el sistema:

$$\begin{aligned} y_1' &= y_2, \\ &\vdots \\ y_{n-1}' &= y_n, \\ y_n' &= f(x, y_1, \ldots, y_{n-1}), \end{aligned} \qquad \text{con condiciones iniciales} \qquad \begin{aligned} y_1(x_0) &= \mu_0, \\ y_2(x_0) &= \mu_1, \\ &\vdots \\ y_n(x_0) &= \mu_{n-1}. \end{aligned}$$

Y por lo tanto permiten resolver el problema de orden n, mayor que uno, con condiciones iniciales

$Y^{(n)} = f(x,y,y', \ldots, y^{(n-1)})$, con condiciones iniciales

$Y(x0) = \mu_0$,

$Y`(x0) = \mu_1$,

$Y^{(n-1)}(x0) = \mu_{n-1}$,

Ya que esto se puede reducir al problema anterior mediante el cambio de variables.

$Yk = Y`k-1$ Para todo $k = 2,\ldots,n$, $Y1 = Y$

Con lo que se obtiene el sistema:

$$\begin{aligned} y_1' &= y_2, \\ &\vdots \\ y_{n-1}' &= y_n, \\ y_n' &= f(x, y_1, \ldots, y_{n-1}), \end{aligned} \qquad \text{con condiciones iniciales} \qquad \begin{aligned} y_1(x_0) &= \mu_0, \\ y_2(x_0) &= \mu_1, \\ &\vdots \\ y_n(x_0) &= \mu_{n-1}. \end{aligned}$$

Sea D un dominio del plano que contenga al punto (X_0,Y_0) y sobre el cual la función f sea continua. Una función y será una solución del problema [20] sobre [a,b] si ara todo $a \leq x \leq b$, $(X,Y(X)) \in D$, existe $Y'(X)$, la derivada tal que $Y'(X) = f(X,Y(X))$, además de satisfacer la condición inicial $Y(X_0) = Y_0$.

Teorema

Si los dos puntos (X,Y_1), $(X,Y_2) \in$ a D entonces la recta vertical que los une también pertenece a D, es decir, $(X, \lambda Y_1 + (1- \lambda)Y_2) \in D$.

Teorema

Sea f una función continua sobre el dominio D y Lipschitz en el segundo argumento, es decir, $\exists\ 0 \leq K$ tal que:

$$|f(X,Y_1) - f(X,Y_2)| \leq K |Y_1 - Y_2| \quad \text{para todo } (X,Y_1), (X,Y_2) \in D.$$

Si (X_0, Y_0) pertenecen al interior del dominio D entonces existe un intervalo $I = (X_0 - \delta, X_0 + \delta)$, sobre el cuál existirá una única solución Y(x) del problema.

Para estudiar la convergencia de los métodos numéricos, se debe imponer la hipótesis más fuerte, es particular que $\frac{\partial f}{\partial y}$. Para que debido al teorema del valor medio, se tenga que f sea de Lipschitz en la segunda variable.

Para estudiar la estabilidad del problema, se considera un problema perturbado adecuado que tenga una solución única sobre un intervalo I. Se considera el problema:

$$y' = f(x, y) + \delta(x),$$
$$y(x_0) = y_0 + \varepsilon,$$

donde δ es una función continua para todo x tal que (x,y) Є D, dado y. Se puede probar que este problema tendrá una única solución, que se denota y(x;δ,ε), sobre un intervalo fijo I = [X_0 - α, X_0 + α], para toda perturbación que satisfaga $|\varepsilon| \leq \varepsilon_0,\ \|\delta\|_\infty \leq \varepsilon_0$ para algún ε_0 suficientemente pequeño. Así:

Teorema

Sean la única solución al problema de perturbación planteado sobre el intervalo I = [X_0 - α, X_0 + α], entonces:

$$\|y(\cdot;\delta,\varepsilon) - y\|_{\infty,I} \leq \frac{1}{1 - K\alpha}(|\varepsilon| + \alpha\|\delta\|_{\infty,I}).$$

Método de Euler

Se reemplaza la derivada por: $\frac{dy(x)}{dx} = \frac{y(x+h) - y(x),}{h}$

siendo h algún número pequeño. Así la ecuación diferencial se transforma en ecuaciones de diferencias:

$$y(x+h) = y(x) + hf(x,y)$$

con $X_{k+1} = X_k + h.$

se tiene que $Y(X_{k+1}) = Y(X_k) + hf(X_k,Y_k)$

Método Predictor Corrector

Los métodos predictor corrector hace uso de una fórmula para una primera aproximación Y_{k+1}, seguida de una fórmula correctora que hace mejoramiento sucesivos. Así por ejemplo una primera aproximación es:

$$y_{k+1}^0 = y_k + hy_k',$$

Que puede ser mejorada con:

$$\begin{aligned} y_{k+1}^1 &= y_k + \frac{1}{2}(y_{k+1}' + y_k') \\ &= y_k + \frac{1}{2}(f(x_{k+1}, y_{k+1}^0) + f(x_k, y_k)). \end{aligned}$$

$y(t_0+y'h)h$

Método Runge – Kutta

Uno de los métodos más simples y que es muy confiable.

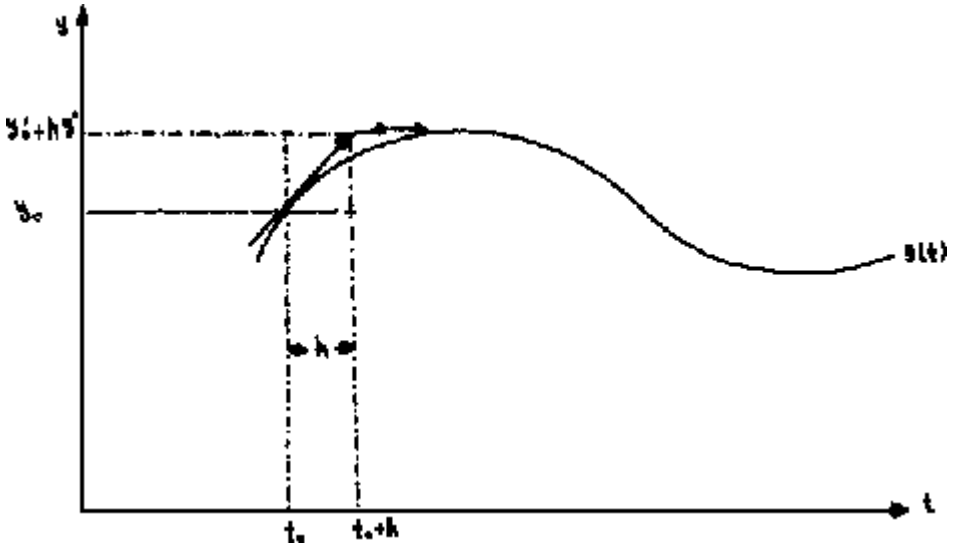
Supóngase una ecuación (trataremos por el momento con una sola variable) de la forma:

$$\frac{dy}{dt} = F \quad \text{donde F es del tipo: } F(y,t) = m(t)y + g(t)$$

La gráfica de la función se representará como en la figura. Las derivadas de dicha función se conocen en todo punto, ya que vienen especificadas por la ecuación diferencial. Si es el incremento de tiempo, entonces, a primera aproximación, se tendrá como valor de la función después de un tiempo , la cantidad:

$$y(t_0+h) = y_0 + y'h$$

Esto se puede representar en la figura.

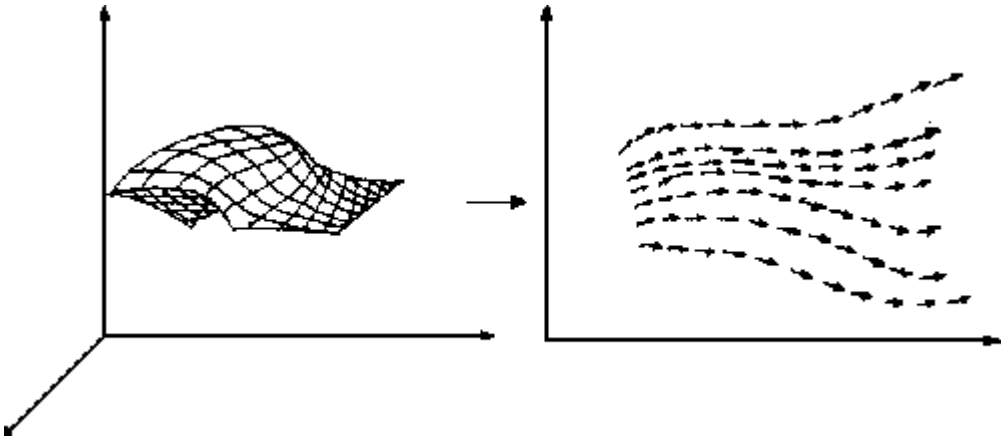


derivada

En el caso no lineal que es el que interesa se escribe: $y(t_0+h) = y(t_0) + hF(y,t)$

aunque eso no es precisamente correcto debido a la no linealidad. Como se ve, este método consiste en tomar una nueva derivada después de cierto tiempo, y así sucesivamente, porque la tangente a la curva está cambiando constantemente su dirección.

Puede entenderse a la función F(y,t) como una superficie en tres dimensiones de la cual nos trasladamos a un campo de tangentes como indica el siguiente dibujo.



En cada punto habrá una flecha indicando en que dirección hay que moverse. Con este procedimiento no se sigue exactamente a la curva pero se puede confiar en que escogiendo suficientemente pequeño habrá un error , de tal manera que el resultado final diferirá del valor real por una cantidad del orden de .

El método que acabamos de exponer se conoce como ``Método de Euler'' y ya está expuesto. La técnica puede mejorarse si en vez de una recta se toma por ejemplo una parábola tangente a cada punto de la curva.

Considere ahora la ecuación y`= y , si se toma y= 1 en t= 0, al desarrollar la función en serie de potencias alrededor de t_0 =0 se tiene:

$$y = y_0 + hy_0' + \frac{h^2}{2!}y_0'' + \frac{h^3}{3!}y_0''' + \cdots = 1 - h + \frac{h^2}{2!} - \frac{h^3}{3!} + \cdots = e^{-h}$$

eso debido a que:

$$\begin{aligned} y_0' &= -1 \\ y_0'' &= (y_0')' = -(y_0)' = -(-1) = 1 \\ y_0''' &= (y_0'')' = -1 \\ \cdots &\quad \cdots \quad \cdots \end{aligned}$$

No hay ningún argumento que impida desarrollar “y” en serie de potencias pero si se toma la función en toda su generalidad se tendrá:

$$\begin{aligned} y' &= F(y,t) \\ y'' &= \frac{d}{dt}F(y,t) = \frac{\partial F}{\partial y}\frac{dy}{dt} + \frac{\partial F}{\partial t} = \frac{\partial F}{\partial y}\cdot F + \frac{\partial F}{\partial t} \\ &= F_y \cdot F + F_t \\ y''' &= \frac{d}{dt}(F_y \cdot F + F_t) \end{aligned}$$

Los subíndices significan derivación parcial con respecto a la variable indicada; entonces, para F’’’ se tendrá:

$$y''' = \frac{d}{dt}F_y \cdot F + F_y \cdot \frac{dF}{dt} + \frac{d}{dt}F_t = \left(\frac{\partial}{\partial y}F'\right) \cdot F + F_y \cdot F' + \frac{\partial}{\partial t}F'$$

Se cambió el orden de la derivación. Sustituyendo F’ resulta lo siguiente:

$$y''' = \frac{\partial}{\partial y}(F_y \cdot F + F_t) \cdot F + F_y \cdot (F_y \cdot F + F_t) + \frac{\partial}{\partial t}(F_y \cdot F + F_t)$$

Resumiendo, se tiene para las tres primeras derivadas:

$$\begin{aligned} y' &= F(y,t) \\ y'' &= F_y \cdot F + F_t \\ y''' &= F_{yy}F^2 + F_y^2 F + F_{yt}F + F_y^2 F + F_y F_t + F_{yt}F + F_y F_t + F_{tt} \\ &= F_{yy}F^2 + 2F_y^2 F + 2F_{yt}F + 2F_y F_t + F_{tt} \end{aligned}$$

Por el mismo procedimiento pueden ser encontradas las derivadas de orden superior. Simbólicamente se resuelve la ecuación diferencial tomando todas las derivadas formando así una serie de Taylor lo que no es práctico porque hay que efectuar una cantidad de cálculos muy grande y tediosa, y eso para cada ecuación diferencial que se presente.

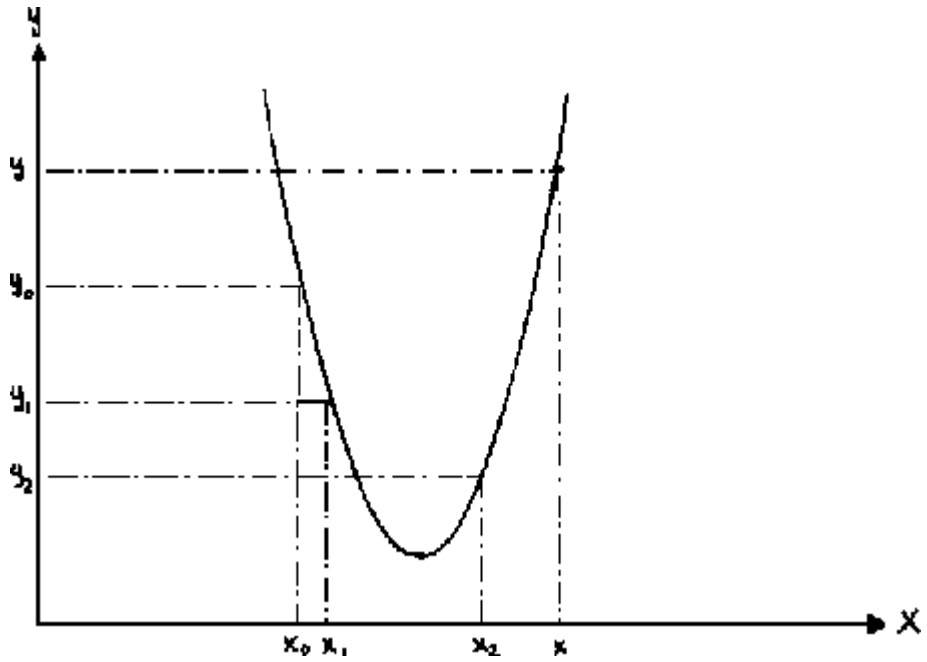
Se desea conocer una técnica más general que permita resolver cualquier ecuación diferencial, sin necesidad de conocer su forma precisa, para obtener las derivadas; el método que emplearemos permite calcularlas de otra manera. La base de este concepto es que existen muchas funciones lineales, por ejemplo, los polinomios forman un espacio vectorial (espacio dual) y por lo tanto puede formarse una base en ese espacio de manera que toda función analítica puede expresarse como una combinación lineal de elementos de dicha base. Los polinomios tiene propiedades muy interesantes, por ejemplo, en un polinomio de segundo grado basta conocer tres valores diferentes de la función para conocer su comportamiento en cualquier otro punto y de esa manera puede encontrarse su derivada como combinación de los tres valores conocidos. Es bien sabido que tres puntos determinan una parábola, de la manera siguiente:

$$y = y_1 \frac{(x - x_2)(x - x_3)}{(x_1 - x_2)(x_1 - x_3)} + y_2 \frac{(x - x_1)(x - x_3)}{(x_2 - x_1)(x_2 - x_3)} + y_3 \frac{(x - x_1)(x - x_2)}{(x_3 - x_1)(x_3 - x_2)}$$

Esta expresión puede también escribirse como un determinante:

$$\begin{vmatrix} 1 & 1 & 1 & 1 \\ x & x_1 & x_2 & x_3 \\ x^2 & x_1^2 & x_2^2 & x_3^2 \\ y & y_1 & y_2 & y_3 \end{vmatrix} = 0$$

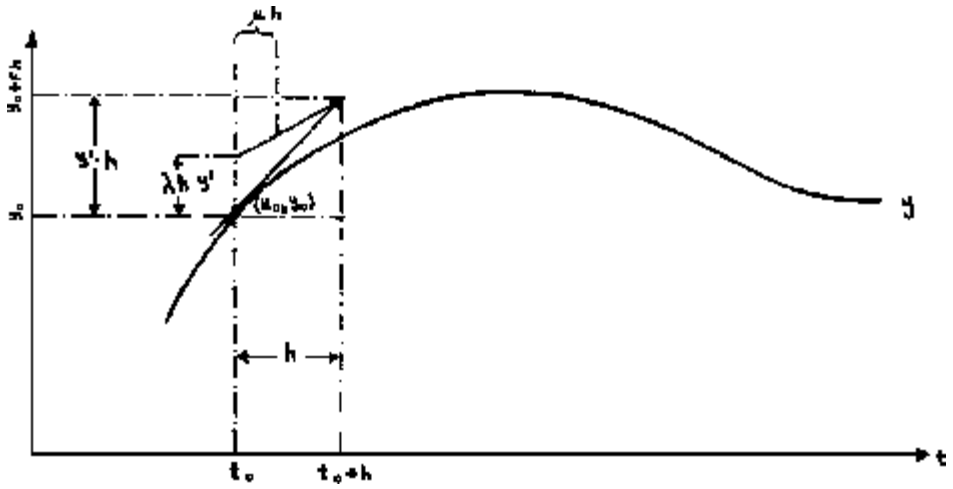
Esta situación se representa en la figura



parábola

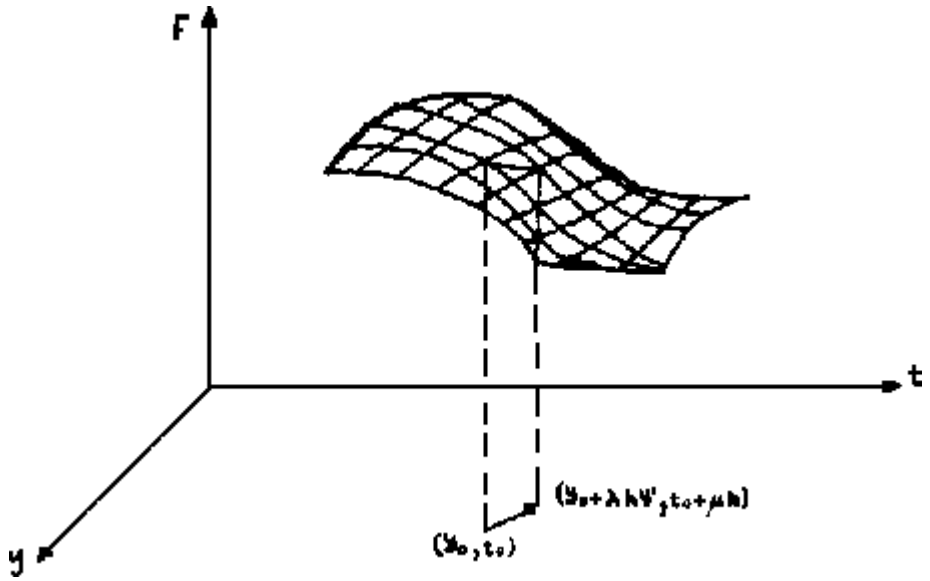
Al hacer operaciones se obtiene un polinomio de segundo grado, así, puede calcularse fácilmente dy/dt porque es una suma de productos y diferencias. Con el objeto de evitar una gran cantidad de pasos algebraicos se procede en una forma equivalente pero más sencilla como veremos en seguida:

Se toma un punto, definido por (yo + λy',to + μh) donde la función expresa la derivada de cualquier curva que pase por ese punto; posiblemente no la curva que nos interesa sino alguna que pase precisamente por ahí con esa derivada, de acuerdo a la figura:



curva

Esperamos obtener las derivadas de F en diversos de tales puntos. Dibujemos esta función en tres dimensiones, representando un cambio como el que acabamos de mencionar, se represente en la siguiente figura.



superficie

Desarróllese la función en serie de potencias alrededor del punto escogido de la manera anterior y utilizando los parámetros indicados, eso es:

$$\begin{aligned} F(y_0+\lambda hF_0, t_0+\mu h) &= F(y_0,t_0)+\lambda hF\frac{\partial F}{\partial y}+\mu h\frac{\partial F}{\partial t}+ \\ &+\tfrac{1}{2}\left(\lambda^2F^2h^2\frac{\partial^2F}{\partial y^2}+2\lambda\mu h^2\frac{\partial^2F}{\partial y\partial t}+\mu^2h^2\frac{\partial^2F}{\partial t^2}+\cdots\right) \end{aligned} \qquad [22]$$

La pendiente en este punto, entonces, queda expresada en términos de la pendiente en el punto original, teniendo en cuenta que F como función de dos variables puede desarrollarse en serie de potencias, como acaba de hacerse. Esta ecuación contiene las derivadas deseadas; la idea consiste en invertirla. Esto se hace por un procedimiento más sencillo, por ejemplo, escribiendo y como:

$$y = \quad y_0+hy'+\frac{h^2}{2}y''+\cdots \qquad [23]$$

Se considerará la hipótesis de suponer para y un desarrollo de la forma:

$$y = \quad y_0+\alpha_0\kappa_0+\alpha_1\kappa_1+\cdots \qquad [24]$$

donde αo y α1 son coeficientes, mientras que k0 y k1 se definen como:

$$\begin{aligned} \kappa_0 &= hF(y_0,t_0)=hF \\ \kappa_1 &= hF(y_0+\lambda\kappa_0, t_0+\mu h) \end{aligned}$$

se tienen dos valores de ***F*** uno en cada punto; puede identificarse a *k0* como el término h, y' que se introdujo en ($y(t_0+h) = y_0 + y'h$). Al expresar y en término de esas dos cantidades se evita el problema de trabajar con las derivadas; el procedimiento es tan legítimo como lo es el obtener la función, la derivada y la segunda derivada usando tres puntos de la curva. Que dicho procedimiento sea o no correcto depende de qué tanto coinciden las series [23] y [24]

Entonces, en lugar de tener que invertir el desarrollo [22] se supone que puede expresarse en la forma que hemos indicado y se verá que error resulta al hacer esa consideración. El error es, naturalmente, la diferencia entre las expresiones [23] y [24].

Al substituir K0 y K1 en la segunda de estas series se tiene:

$$y = y_0 + \alpha_0 F h + \alpha_1 h(F + \lambda h F F_y + \mu h F_t) + o h^3 + \cdots \qquad [25].$$

haciendo también una substitución en [23] y usando (22] resulta

$$y = y_0 + hF + \frac{h^2}{2}(FF_y + F_t) + o h^3 + \cdots \qquad [26].$$

Por comparación de [25] y [26] se observa que son muy próximas una de otra si los coeficientes de las mismas potencias de h son aproximadamente iguales, o sea, que

$$\begin{aligned} Fh\alpha_0 + Fh\alpha_1 &\approx hF \\ h^2(\lambda\alpha_1 FF_y + \alpha_1\mu F_t) &\approx (FF_y + F_t)h^2 \end{aligned}$$

Si la aproximación no representa a, por lo menos y, la diferencia entre aquella y el caso real tiende a cero a tercer orden en h. Como deseamos que el método tenga validez general, debemos pedir que se cumplan las relaciones

$$\alpha_0 + \alpha_1 = 1 \quad , \quad \lambda\alpha_1 = \frac{1}{2} \quad , \quad \mu\alpha_1 = \frac{1}{2}$$

Si esto se cumple, podemos confiar en nuestro método. Como tenemos cuatro incógnitas y sólo tres ecuaciones, el sistema esta indeterminado, pero si se toma α1 = c por ejemplo, entonces todas las soluciones quedan en términos del parámetro y se tendrá por consiguiente :

$$\begin{aligned} \alpha_1 &= c & \lambda &= \frac{1}{2c} \\ \alpha_0 &= 1 - c & \mu &= \frac{1}{2c} \end{aligned}$$

un valor muy favorecido para c es ½ porque de esa manera se tiene

$$\alpha_1 = \frac{1}{2} \qquad \lambda = 1$$
$$\alpha_0 = \frac{1}{2} \qquad \mu = 1$$

La selección de c de ninguna manera aumenta el orden de aproximación (igualar términos en
h^3) sino que simplifica el cómputo si se toma un valor adecuado. La técnica que acabamos de exponer es conocida como ``Método de Runge-Kutta'' y puede extenderse para incluir aproximaciones de orden mayor en las potencias de h. Escribiendo la fórmula de Runge-Kutta para cualquier punto se tiene:

$$y_{n+1} = y_n + \alpha_0 \kappa_0 + \alpha_1 \kappa_1$$

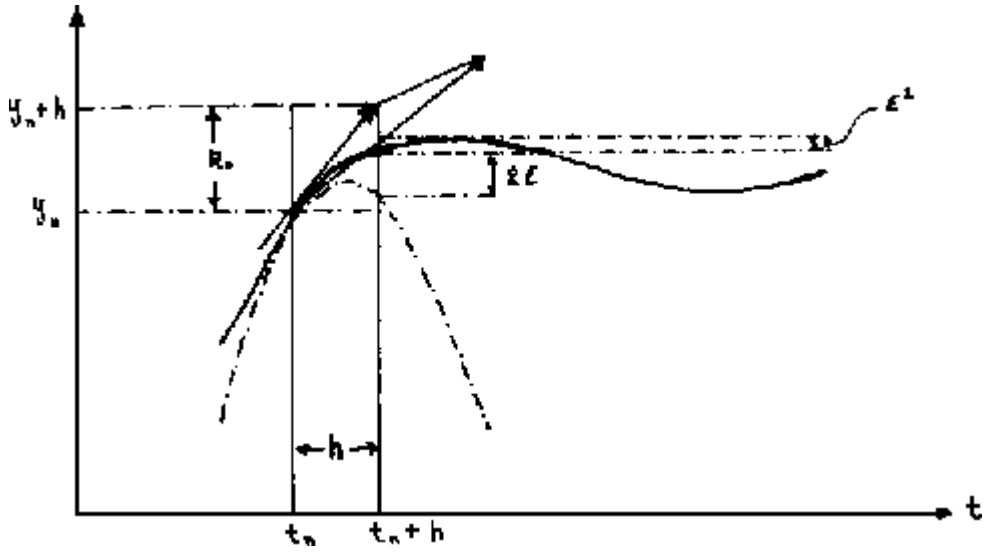
donde:

$$\kappa_0 = hF(y_n, t_n)$$
$$\kappa_1 = hF(y_n + \lambda\kappa_0, t_n + \mu h)$$

Al usar el valor convenido anteriormente para c, resulta

$$\begin{aligned} y_{n+1} &= y_n + \tfrac{1}{2}\kappa_0 + \tfrac{1}{2}\kappa_1 \\ \kappa_0 &= hF(y_n, t_n) \\ \kappa_1 &= hF(y_n + \kappa_0, t_n + h) \end{aligned} \qquad [27]$$

Gráficamente, el proceso puede extenderse de la manera siguiente: Primero hay que formar k0 que es el incremento vertical de la figura; con k1 se realiza el segundo paso de Euler en el punto $(y_n + \kappa, t_n + h)$ .La fórmula [27] dice que hay que sumar los dos incrementos y el error obtenido es del orden de ε^2cuando hay un error de 2ε al ajustar nuestro elemento de arco con la parábola.



Cuando se hacen los cálculos a tercer orden el error irá como ε^3 y el punto resultante se toma como el peso de tres incrementos. Existen varios métodos de Runge-Kutta y su elaboración es más o menos la misma. Hay otros métodos y todos ellos suponen que son conocidas las derivadas. La ventaja de los métodos de Runge-Kutta está en que no hay que calcular derivadas, además, es posible modificar el intervalo (se aumenta o se reduce el incremento) de acuerdo con la variación de la tangente. El inconveniente que representan estos métodos es el de que hay que realizar un gran número de cálculos para obtener *F* y cada paso está basado en los anteriores. En general, supóngase que es un punto sobre la curva y que puede escribirse en términos de uno de sus valores anteriores más múltiplos de *F* en diferentes puntos. Se expresa y en serie de potencias y se comparan los dos desarrollos. En otras palabras, suponemos que:

$$y = y_0 + \alpha F(y_1, t_1) + \beta F(y_2, t_2) + \gamma F(y_3, t_3) + \cdots$$

Aquí, la aproximación (el orden) depende del número de puntos que se consideren, uno para la primera aproximación, dos para la segunda, etc. Además, se supone, como hemos mencionado, que:

$$\text{y} = y_0 + hy_0' \frac{h^2}{2!} y_0'' + \frac{h^3}{3!} y_0''' + \cdots \qquad [28]$$

El plan es el mismo que en caso anterior; hay que expresar y(t) en términos de (y0,t0 y elaborar ***F*** en varios puntos para obtener una serie; los valores de ***F*** se obtienen con ayuda de la ecuación diferencial. Hecho todo lo anterior, la tarea consiste en seleccionar los coeficientes de tal manera que haya una correspondencia, válida hasta la potencia deseada.

Presentamos ahora varios resultados sin entrar en más detalles, ya que el procedimiento es el mismo que se discutió en páginas anteriores. En [28] se escribe la fórmula hasta tener órden, lo que significa que el error es a cuarto orden en h. En ese caso se tiene:

$$\begin{aligned} y_{n+1} &= y_n + \tfrac{1}{6}(\kappa_0 + 4\kappa_1 + \kappa_2) \\ \kappa_0 &= hF(y_n, t_n) \\ \kappa_1 &= hF(y_n + \tfrac{1}{5}\kappa_0, t_n + \tfrac{1}{3}h) \\ \kappa_2 &= hF(y_n + \tfrac{3}{2}\kappa_0, t_n + \tfrac{3}{2}h) \end{aligned} \qquad [29]$$

Esta fórmula se atribuye a Kutta. Hay otra que se atribuye a Heun y es la siguiente:

$$\begin{aligned} y_{n+1} &= y_n + \tfrac{1}{4}(\kappa_0 + 3\kappa_2) \\ \kappa_0 &= hF(y_n, t_n) \\ \kappa_1 &= hF(y_n + \tfrac{1}{5}\kappa_0, t_n + \tfrac{3}{2}h) \\ \kappa_2 &= hF(y_n + \tfrac{3}{2}\kappa_1, t_n + \tfrac{3}{2}h) \end{aligned} \qquad [30]$$

Nótese que en el incremento de la función no aparece k1. Este sólo se usa como un paso intermedio. Con un método de Runge-Kutta a cuarto orden se obtiene:

$$y_{n+1} = y_n + \frac{1}{6}(\kappa_0 + 2\kappa_1 + 2\kappa_2 + \kappa_3) + o(h^4) \qquad [31]$$

$$\begin{aligned} \kappa_0 &= hF(y_n, t_n) \\ \kappa_1 &= hF(y_n + \tfrac{1}{2}\kappa_0, t_n + \tfrac{1}{2}h) \\ \kappa_2 &= hF(y_n + \tfrac{1}{2}\kappa_1, t_n + \tfrac{1}{2}h) \\ \kappa_2 &= hF(y_n + \kappa_2, t_n + h) \end{aligned} \qquad [32]$$