

Tema 1: Análisis del error.

A.- Estudio y tipología del error :

Si tenemos un problema p (problema real), con el fin de estudiarlo, modularemos dicho problema mediante una aproximación \tilde{p} , a costa de un *error de modulación* E_m en el paso de la realidad a la aproximación. Si la solución a \tilde{p} diverge demasiado de la solución a p , el planteamiento o el algoritmo de resolución no son válidos para el problema real en concreto.

Error	depende de
E_{inp} Error de entrada (input) \Rightarrow	del condicionamiento del problema (sin solución)
E_{int} Error intermedio \Rightarrow	de la estabilidad del algoritmo (en la máquina, ...)
E_{alg} Error del algoritmo \Rightarrow	se da al ejecutarse el algoritmo en sí
E_{red} Error de redondeo y aritmética de la computadora	

Def. 1: **Error absoluto:** se mide con la relación: $E_{abs}: |p - \tilde{p}|$

Def. 2: **Error relativo:** es el resultado del cociente: $E_{rel}: \frac{|p - \tilde{p}|}{|p|}$

Def. 3: **Error de cancelación:** E_{can} : Se da en operaciones donde se restan 2 números muy próximos. El resultado es la anulación de algunos dígitos que podrían ser significativos.

Def. 4: **Aproximación de un número:** \tilde{p} se aproxima a p con t dígitos significativos si el error relativo responde a: $E_{rel} < 5 * 10^{-t}$

Def. 5: **Error normalizado:** El error está normalizado si $|a| \leq 10^{-1}$

Def. 6: **Tolerancia:** es el error máximo que deseamos permitir en un cálculo.

B.- Representación numérica:

En un ordenador, un número tiene una representación especial denominada número máquina, que se compone de una mantisa o fracción y un exponente o característica.

Nota 1: Si el resultado de una operación es menor que el menor número que puede representar la máquina, el compilador retorna “*Underflow*”; pero si es mayor que el mayor que pueda memorizar, devuelve “*Overflow*”. En ambos casos se detienen los cálculos del compilador.

Def. 7: Notación en punto flotante: Un número real x se escribe en notación punto flotante de esta manera:

$$x = \pm a \cdot 10^b: \text{ mantisa} \cdot \text{base}^{\text{exponente}}$$

Por ejemplo, para escribir un número en la representación punto flotante de un compilador concreto, escribiremos:

$$1 \text{ bit [signo]} + 7 \text{ bits [exponente]} \text{ (en base 16)} + 24 \text{ bits [mantisa]}$$

Def. 8: Número normalizado: un número a está normalizado si el primer dígito decimal es distinto de cero, de manera que:

$$0.1 \leq |a| \leq 1$$

Por ejemplo: Si $x = 0.028 \cdot 10^{28}$ entonces se convierte en $x = 0.285 \cdot 10^{27}$

Nota 2: El ordenador usa base 2 (binaria) en los cálculos: $m \cdot 2^e$

Def. 9: FUNCIÓN REDONDEO $Rd(x)$: UN NÚMERO SE REDONDEA O APROXIMA A t CIFRAS CON LA FUNCIÓN Rd DEFINIDA A CONTINUACIÓN:

$$Rd : R \xrightarrow{t} A$$

$$x \quad rd(x) = \tilde{x}, \quad t \text{ dígitos}$$

donde:

$$\tilde{x} = a' \cdot 10^b \quad y \quad a' = 0.\alpha_1\alpha_2\alpha_3\dots\alpha_k \quad y \quad k = \begin{cases} t & \leftrightarrow 0 \leq \alpha_{t+1} \leq 4 \\ t+1 & \leftrightarrow 5 \leq \alpha_{t+1} \leq 9 \end{cases}$$

Def. 10: Cifras decimales exactas: Ganar n cifras decimales (exactas) entre 2 etapas es reducir el error 10^{-n} entre una etapa y la siguiente.

Def. 11: Error absoluto y relativo producido al redondear:

$$E_{abs} = |rd(x) - x| \leq 5 \cdot 10^{-(t+1)} \cdot 10^b \quad \text{más exacto que } E_{abs}$$

$$E_{rel} = \frac{E_{abs}}{|x|} \leq \frac{5 \cdot 10^{-(t+1)}}{|a|} \leq \frac{5 \cdot 10^{-(t+1)}}{10^{-1}} = 5 \cdot 10^{-t} \quad \text{indica la magnitud del error}$$

Def. 12: Error de precisión de la máquina EPS: de *epsilon* ε : $eps = 5 \cdot 10^{-t}$.

De forma que si el error E tiende a *cero*:

$$E \rightarrow 0, \quad E_{abs} \leq eps, \quad \Rightarrow \quad |E| \leq eps,$$

y así

$$\tilde{x} = x + \varepsilon \cdot x = x \cdot (1 + \varepsilon)$$

Def. 13: Factor de corrección: $\tilde{x} = x \cdot (1 + \varepsilon)$ donde $eps \equiv \varepsilon$

C.- Ejemplos:

Ejemplo 1: Error de cancelación: diferencia de números muy próximos.

La forma normal, aunque incorrecta, de hacer la operación es:

$$\sqrt{9876} - \sqrt{9875} = 0'000503142 \cdot 10 = 5'03142 \overbrace{0000}^{\text{error}} \cdot 10^{-3}$$

La forma correcta de hacerlo es multiplicar y dividir por el conjugado.

$$\sqrt{9876} - \sqrt{9875} = \frac{1}{\sqrt{9876} + \sqrt{9875}} = 5'031418679 \cdot 10^{-3}$$

Ejemplo 2: Error de cancelación: Cálculo de raíces de ecuaciones de segundo grado:

Por ejemplo, para $x^2 - 1634x + 2 = 0$ la solución que genera un valor incorrecto es:

$$\left\{ \begin{array}{l} x = 817 \pm \sqrt{667487} \\ x_1 = 817 + 816'9987760 = 1'633998776 \cdot 10^{-3} \\ x_2 = 817 - 816'9987760 = 1'224 \underbrace{000000}_{\text{ERROR}} \cdot 10^{-3} \end{array} \right.$$

Y la solución que genera los valores correctos es:

$$x_1 * x_2 = 2 \Rightarrow x_2 = \frac{2}{x_1} = 1'223991125 \cdot 10^{-3}$$

Ejemplo 3: Error de cancelación: Suma de números muy pequeños:

Por ejemplo, en la serie aritmética S_n siguiente.

$$(a) \quad S_n = 1 + \sum_{k=1}^n \frac{1}{k^2 + k}, \text{ que es equivalent e a (b)}$$

$$(b) \quad S_n = 1 + \sum_{k=1}^n \left(\frac{1}{k} - \frac{1}{k+1} \right) = 2 - \frac{1}{n+1}$$

Con una aritmética exacta, tendríamos que obtener

$$S_9 = 1'9$$

$$S_{99} = 1'99$$

$$S_{999} = 1'999$$

$$S_{9999} = 1'9999, \dots \dots \dots$$

Pero en realidad obtenemos cálculos erróneos:

$$S_9 = 1'9$$

$$S_{99} = 1'990000003$$

$$S_{999} = 1'999000003$$

$$S_{9999} = 1'9998999972, \dots \dots \dots$$

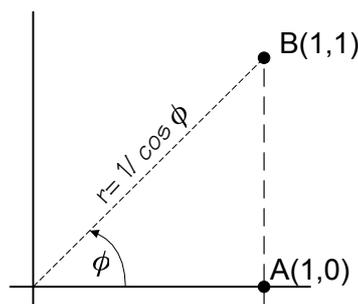
Mas, si ordenamos los términos de forma decreciente, y rellenamos con 0 hasta obtener 2^k términos, como sigue ahora, obtenemos los resultados esperados:

$$(c) \quad S_n = \frac{1}{(n+1)n} + \frac{1}{(n-1)n} + \dots + \frac{1}{2*1} + 1$$

Ejemplo 4: Usando el método anterior, **calcular la distancia entre A y B**, sabiendo que:

$$\varphi_k = \pi k / 4n \quad \text{y} \quad r_k = 1 / \cos \varphi_k \quad / \quad k = 1 \dots n$$

<u>N</u>	<u>Método tradicional</u>	<u>Método correcto</u>
2	1'000000001	1'000000001
4	0'999999996	1'000000001
8	1'000000002	1'000000000
16	1'000000079	1
32	1'000000312	1
64	0'999999333	1
128	0'999996236	1'000000006
256	0'999986218	1'000000005
512	0'999985107	1'000000006
1024	0'999817631	1'000000006



Ejemplo 5: Cálculo de π

La longitud de un polígono regular de n -lados, inscrito en una circunferencia de radio $r=1$, es igual a: $\ell = 2*n*\text{sen}(\pi/n)$. Por tanto, tenemos que :

$$f(n) = n * \text{sen} \frac{\pi}{4} \xrightarrow{n \rightarrow \infty} \pi$$

Tomando la sucesión $n = 2^k$, obtenemos:

$$y^k = 2^k \text{sen}(2^{-k} \pi) \xrightarrow{k \rightarrow \infty} \pi$$

Usando $\text{sen} \frac{\alpha}{2} = \sqrt{\frac{1 - \cos \alpha}{2}}$ se llega a

$$\begin{cases} y_1 = 2 \\ y_{k+1} = 2^{k+1} \sqrt{\frac{1 - \sqrt{1 - (2^{-k} y_k)^2}}{2}} \end{cases}$$

Y usando $1 - x = \frac{1 - x^2}{1 + x}$ y con $x = \sqrt{1 - (2^{-k} y_k)^2}$ tendremos esto:

$$\begin{cases} y_1 = 2 \\ y_{k+1} = y_k \sqrt{\frac{2}{1 + \sqrt{1 - (2^{-k} y_k)^2}}} \end{cases}$$

Tema 2: Ecuaciones no lineales

A.- Introducción:

Buscamos la solución p a una función continua $f(x)$ donde $f(p)=0$. Podemos intuir los ceros del polinomio gráficamente, para detectar el intervalo donde está la solución, pero lo solucionaremos aproximando $f(x)$ a un polinomio $P_n(x)$, con $2k$ raíces complejas y $n-2k$ reales.

Veamos algunos teoremas útiles sobre las raíces de una función:

Teorema 1: Teorema de Bolzano sobre el cambio de signo de una función:

$$f(a) \cdot f(b) < 0 \leftrightarrow \exists p \in (a, b) \mapsto f(p) = 0 / f \in \text{Cont}(a, b)$$

Teorema 2: Teorema de Rolle sobre la primera derivada nula de una función:

$$f'(c) = 0 \leftrightarrow f(a) = f(b) = 0 \mapsto c \in (a, b)$$

Nota 3: Las raíces complejas siempre van a pares. Esto significa que, para un polinomio de orden n , siendo n impar, existe al menos 1 raíz de carácter real.

Pero, ¿qué ocurre cuando hay raíces múltiples en ecuaciones polinómicas (algebraicas)?.

Los métodos numéricos de resolución de ecuaciones presentan dificultades ante raíces múltiples, ya que se basan en la existencia de un único cambio de signo en la función en un intervalo. Para ecuaciones polinómicas se busca una ecuación con iguales raíces, pero simples.

Por el **Teorema fundamental del Álgebra**, sabemos que un polinomio de grado n tiene n raíces, y puede ser factorizado, y también agruparse según la multiplicidad m_i de sus raíces:

$$P_n(x) = a_0(x - x_1)(x - x_2)\dots(x - x_n) \quad \text{con } x_1, x_2, \dots, x_k \text{ raíces}$$

$$P_n(x) = a_0(x - x_1)^{m_1}(x - x_2)^{m_2}\dots(x - x_k)^{m_k}, \quad \text{con } m_1 + m_2 + \dots + m_k = n$$

Derivando la expresión anterior, y operando, obtendremos un polinomio $Q(x)$ cuyas raíces son las mismas que las de la función anterior, pero con multiplicidad simple.

$$P'(x) = na_0(x - x_1)^{m_1-1}(x - x_2)^{m_2-1}\dots(x - x_k)^{m_k-1}Q_{k-1}(x), \quad \text{con } Q_{k-1}(x_i) \neq 0 / i = 1, 2, \dots, k$$

$$D(x) = \text{med}[P(x), P'(x)] = (x - x_1)^{m_1-1}(x - x_2)^{m_2-1}\dots(x - x_k)^{m_k-1}$$

$$Q(x) = \frac{P(x)}{D(x)} = a_0(x - x_1)(x - x_2)\dots(x - x_k)$$

B.- Métodos iniciales:

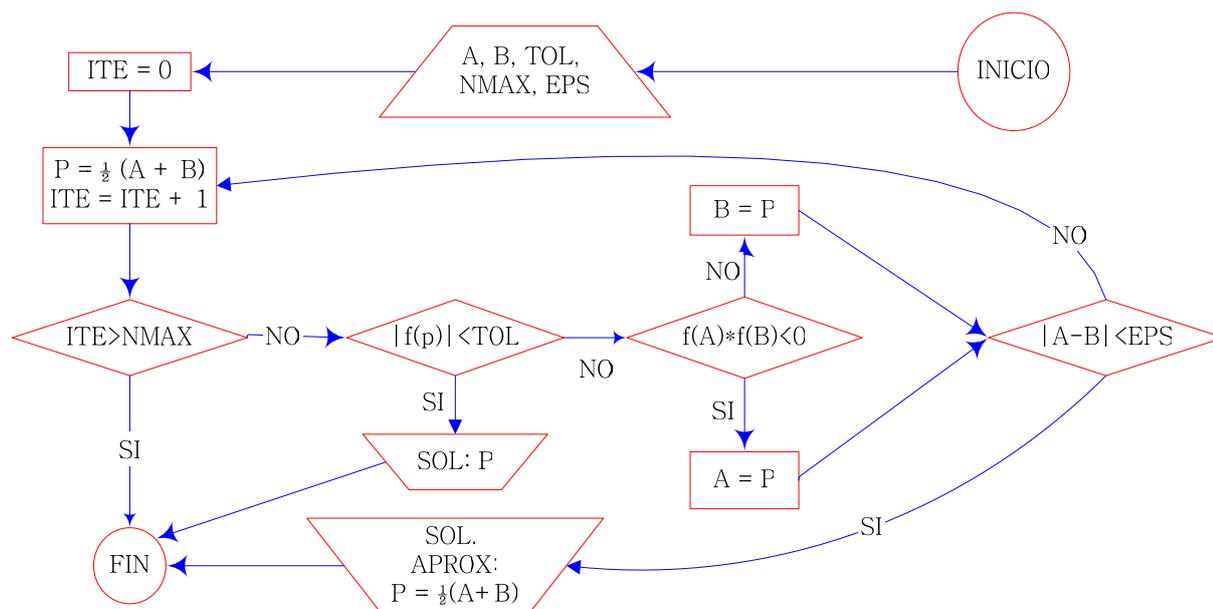
Una vez tenemos la función objeto de estudio en una forma con raíces simples, habrá que separar tales raíces en intervalos suficientemente delimitados.

1.- Método de Bisección:

Este método lineal y de convergencia segura, aplica directamente el teorema de *Bolzano*, dividiendo sucesivamente la función en hemisferios, e itera con el intervalo que contiene la raíz. Da apoyo a otros métodos, proporcionando un intervalo de entrada aproximado.

Algoritmo 1: BISECCION

Calcula la raíz p , o su aproximación \tilde{p} , de una función $f(x)$ en $x \in (a,b)$, en ITE iteraciones, tal que $ITE \leq NMAX$, y donde $NMAX$ es el máximo de iteraciones deseadas; con una tolerancia TOL , y sabiendo que el factor de corrección de la máquina es EPS .



Nota 4: Es preferible calcular el punto medio de $[a,b]$ como $m = a + \frac{b-a}{2}$, para que m no se salga del intervalo en valores muy pequeños de a y b .

Uniéndolo n puntos de un conjunto de valores proporcionado, podemos construir una gráfica lineal, aunque no derivable, que se aproxime a la solución del problema. Incluso, si llegamos a aproximar por parábolas, no existe una segunda derivada. Es suficiente con aproximar por “esplines cúbicos”, porque el ojo humano solo reconoce hasta la 3ª derivada.

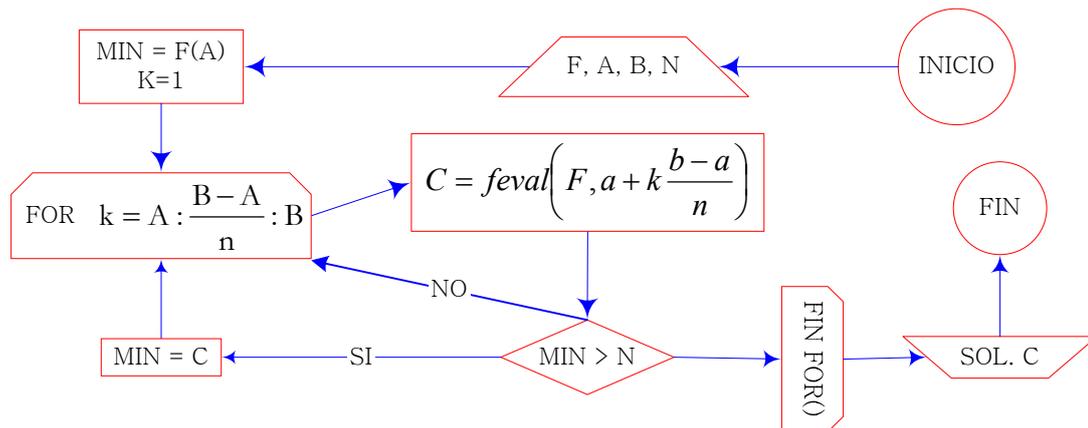
(El ajuste de curvas se ampliará en el tema 6)

2.- Cálculo de las cotas de una función:

Podemos obtener los valores discretos de x (la máquina no logra valores exactos) que ofrecen las cotas de una función. Luego calculamos la altura de la función, su mínimo o máximo.

Algoritmo 2: CÁLCULO DEL MÍNIMO DE UNA FUNCIÓN

Calcula el mínimo de una función en el intervalo (a,b) y con un mayado de N puntos.



3.- Método del Punto Fijo:

Def. 14: **Función contractiva:** un función es contractiva si verifica que:

$$|f(x_1) - f(x_2)| < |x_1 - x_2| \quad / \quad \forall x_1, x_2 \in \mathcal{R}$$

Teorema 3: Una **función derivable** es **contractiva** si $|f'(x)| \leq q < 1$, $/ \quad \forall x \in \mathcal{R}$

Para resolver $f(x)$, se escribe de la forma $x = \varphi(x)$, con $\varphi(x)$ función contractiva. Se construye luego la sucesión $x_{n+1} = \varphi(x_n)$ a partir del valor inicial x_0 , cuya convergencia se garantiza en el teorema del punto fijo.

Los ceros de $f(x)$ son los puntos fijos, en adelante PF , de $\varphi(x)$, dado que $\varphi(x)$ y se cruzan en $[p, f(p)]$, con $f(p) = 0$. Puede escribirse $\varphi(x)$ como $\varphi(x) = x - f(x)$.

Teorema 4: Teorema del Punto Fijo: La función contractiva $\varphi(x)$ es óptima, es decir, tiene un PF p_0 que converge a p , si cumple estas 2 condiciones:

- $\{ g: [a,b] \in (a,b), g \in \mathcal{D}(a,b) \} \wedge \{ |g'(x)| \leq q < 1 / \forall x \in [a,b] \} \Rightarrow$
- $\Rightarrow \exists$ un único punto fijo p_0 en (a,b)
- \Rightarrow Este algoritmo converge a $p / \forall p_0 \in (a,b)$.
- \Rightarrow Se cumple que: $k = \max(|g'(x)|)$, entonces: $|p_n - p| \leq \frac{k^n}{1-k} |p_1 - p_0|$

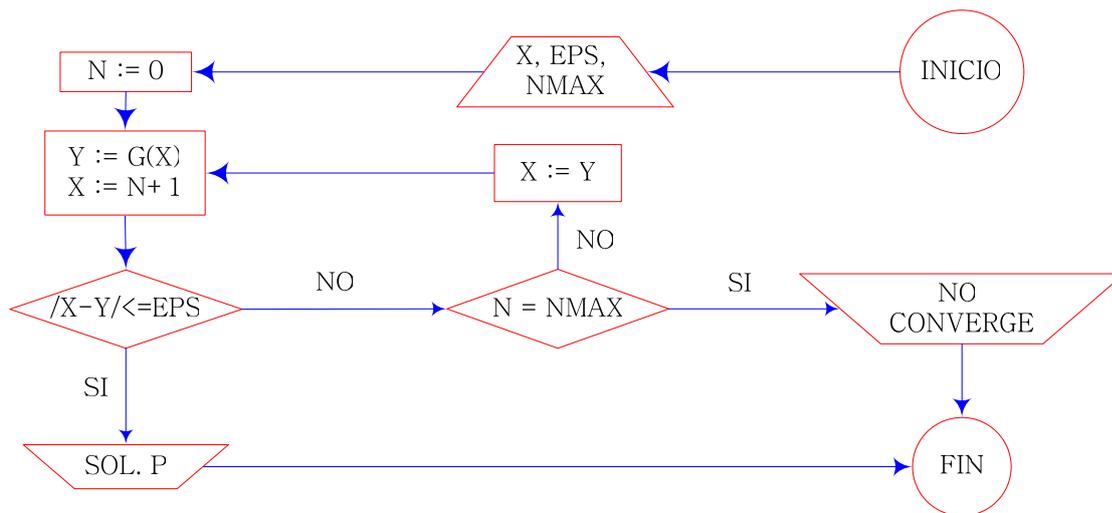
Nota 5: El punto fijo p_0 debe ser próximo a p para que el algoritmo converja.

Teorema 5: Teorema local del Punto Fijo: el *algoritmo del punto fijo* da una serie de PF's p_n que convergen a p (orden de convergencia $\alpha=m$) si:

$$p=g(p) \wedge g'(p)=0=g''(p)=\dots=g^{(m-1)}(p) \wedge g^{(m)}(p)\neq 0 \Rightarrow$$

Algoritmo 3: PUNTO FIJO

Calcula el punto fijo de una función en el intervalo (a,b) , donde Nmax es el número máximo de iteraciones y X es la condición inicial.



Ejemplo 6: Calcular la raíz cuadrada de 3. (Ver anexo de ejercicios)

Se puede hacer, por ejemplo, buscando la raíz positiva de la ecuación $f(x)=0$ con $f(x)=x^2-3$.

- ◆ Por el método de Bisección:

Dado que $f(1)=-2<0$ y $f(2)=1$, el teorema de Bolzano nos garantiza una raíz, que, además no es única, ya que $f'(x)=2x$ no se anula en $[1,2]$. Para obtener la raíz con k cifras exactas, o lo que es lo mismo, con un error $E<10^{-k}$, detendremos el proceso en la iteración n -ésima:

$$\frac{2-1}{2^{n+1}} < 10^{-k} \Rightarrow 2^{n+1} > 10^k$$

Por ejemplo, para $k=14$, nos detendremos en la media m_{46} : $\frac{2-1}{2^{n+1}} < 10^{-14} \Rightarrow 2^{n+1} > 10^{14} \Rightarrow n \geq 46$

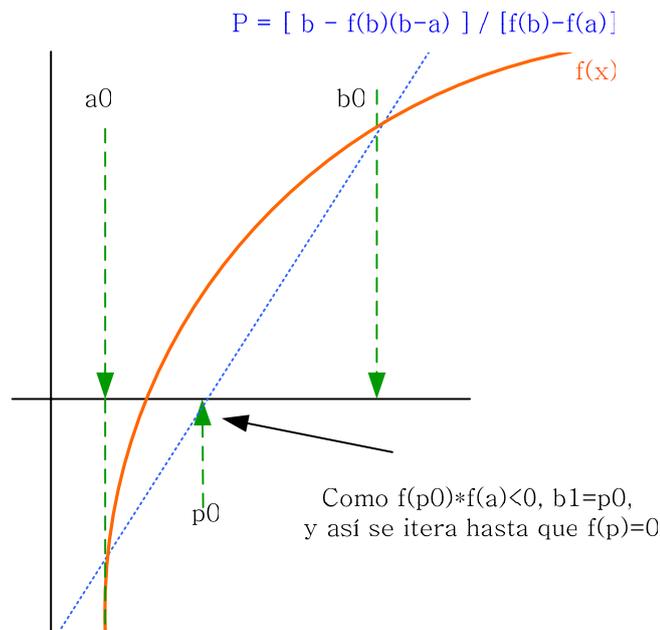
- ◆ Por el método del Punto Fijo:

El error viene dado por $\varepsilon_n < \frac{|f(x_n)|}{\min_{x \in [1,2]} |f'(x_n)|}$,

por lo que pararemos en la iteración 26 si deseamos una tolerancia de 10^{-14} : $\varepsilon_{26} < \frac{|x_{26}^2 - 3|}{2} < 10^{-14}$

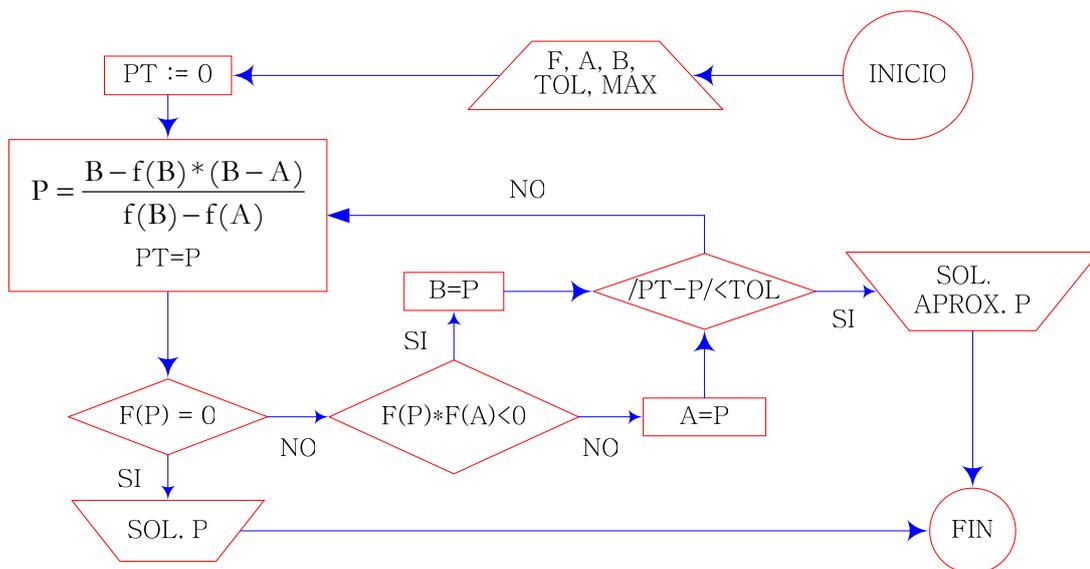
4.- Método de la Regula Falsi (posición falsa)

Intermedio entre *Bisección* y *Secante*, se basa en la recta secante $[(a, f(a)), (b, f(b))]$, siendo a y b extremos del intervalo con la raíz de f . Se calcula el punto p , donde la secante corta al eje OX , de modo que, si $f(p) \cdot f(a) < 0$, definimos $a_{i+1} = a_i$ y $b_{i+1} = p_i$; En caso contrario, $a_{i+1} = p_i$ y $b_{i+1} = b_i$.



Algoritmo 4: POSICIÓN FALSA O “REGULA FALSI”

Calcula la raíz de una función en un intervalo (a,b) . Los datos necesarios son únicamente la función y el intervalo donde se presume está la raíz.



C.- Métodos de mayor convergencia:

1.- Método de Newton-Raphson-Fourier:

Conocido como método de Newton-Raphson o simplemente de Newton, lo diseñó Raphson y Fourier lo rediseñó. Por su convergencia cuadrática, es muy usado actualmente.

$$f(x) = 0 \equiv x = x - \frac{f(x)}{f'(x)} \equiv g(x) \quad \text{de donde:} \quad g'(x) = 1 - \frac{f'(x)f'(x) - f(x)f''(x)}{[f'(x)]^2}$$

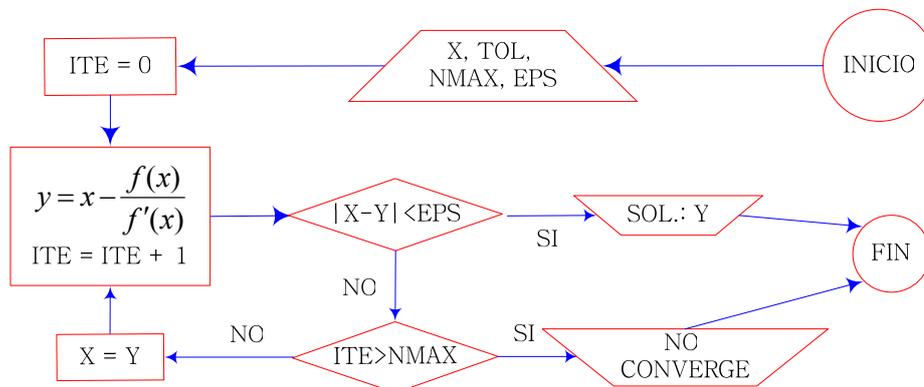
$$g'(p) = 1 - \frac{[f'(p)]^2 - f(x) \cdot 0}{[f'(p)]^2} = 1 - \frac{[f'(p)]^2 - 0}{[f'(p)]^2} = 1 - 1 = 0$$

Nota 6: Si intentáramos calcular la función $g(x)$ del algoritmo del punto fijo con el método de Newton, obtendríamos un método que es, al menos, de orden cuadrático: $\alpha \geq 2$

Nota 7: Halley diseñó un método de convergencia cúbica mucho mejor, pero no se usa porque la tecnología actual no lo soporta y los resultados son similares.

Algoritmo 5: NEWTON – RAPHSON - FOURIER

Los datos necesarios son la función, su derivada y el intervalo donde se presume la raíz.



Resulta ineficiente sin una buena aproximación en un intervalo suficientemente pequeño y derivable. Además, sólo es aplicable si existe la 2ª derivada para cálculos posteriores.

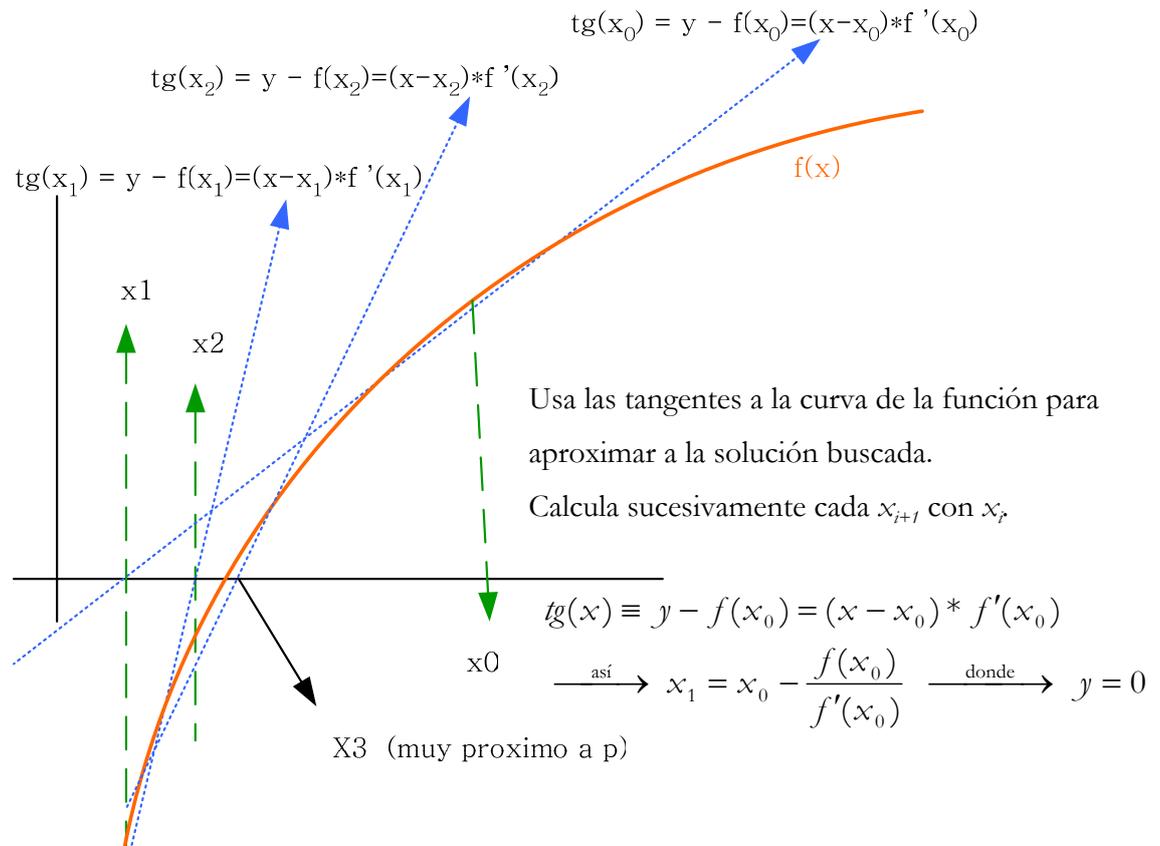
Sería un mal criterio de paro el elegido $(|x_n - y| \leq \epsilon)$ ya que los x_i son próximos (si las tangentes son paralelas, diverge). A veces es mejor usar uno de estos criterios o incluso ambos:

$$\frac{|x_{n+1} - x_n|}{|x_n|} < \epsilon \quad \text{o bien} \quad |f(x_n)| < \epsilon$$

Nota 8: La convergencia se ralentiza con raíces múltiples. Para acelerarla aplicamos esta otra fórmula, siendo k el grado de la primera derivada que no se anula en la raíz (su multiplicidad)

$$f(x) = 0 \equiv x = x - k \frac{f(x)}{f'(x)} \equiv g(x)$$

Veamos la base gráfica del método:



Deberemos tener en cuenta las siguientes observaciones:

- ◆ Si en las proximidades de la raíz existe un punto de inflexión, el método diverge.
- ◆ Cerca de máximos o mínimos locales oscila, persistiendo o encontrando pendientes casi nulas, donde la solución ofrecida se aleja del área de interés donde está la solución.
- ◆ Un valor inicial cercano a una raíz puede converger a otra raíz muy distante como consecuencia de pendientes cercanas a cero. Una pendiente nula provoca una división por cero, porque una tangente horizontal nunca corta al eje de abscisas.

a.- Elección del punto inicial de iteración p_0

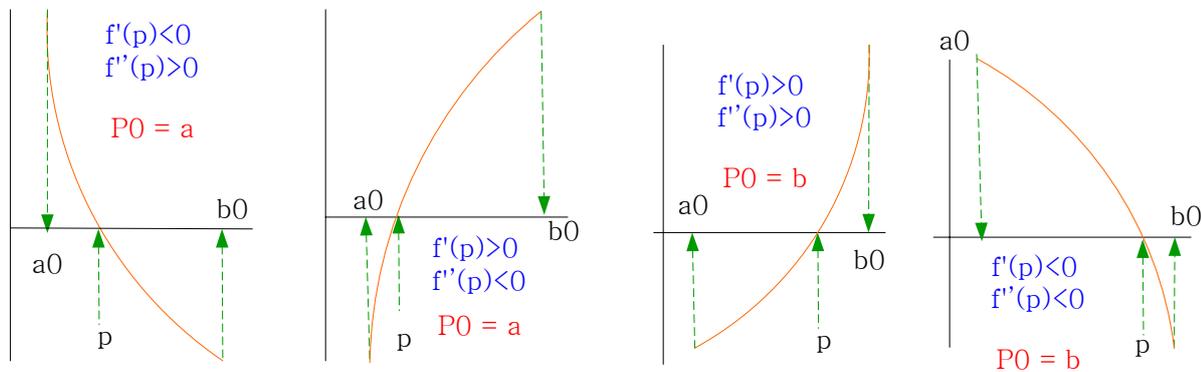
Si el método puede divergir para algún p_0 , debemos hacer un análisis gráfico, y lanzar el método de bisección sobre un intervalo apropiado para obtener un p_0 que se presuma bueno.

No debe elegirse nunca p_0 con $f'(p_0)=0$, porque la tangente es una asíntota en $tg(p_0)=0$. Debe elegirse el extremo del intervalo cuyo signo coincida con el de $f''(p)$, es decir:

$$p_0 = \begin{cases} a \leftarrow f(a) * f''(a) > 0 \\ b \leftarrow f(b) * f''(b) > 0 \end{cases}$$

Nota 9: Este resultado se conoce como *Regla de Fourier*.

Teorema 6: Regla de Fourier: sea $f(x)$ una función continua y 2 veces derivable en $[a,b]$. Si $\text{sig}(f(a)) \neq \text{sig}(f(b))$, y f' y f'' no anulables en $[a,b]$, existe una única raíz $f(x)=0$ en $[a,b]$. Y además, se puede garantizar la convergencia del método de Newton tomando como valor inicial p_0 el extremo del intervalo en el que la función y su segunda derivada tienen el mismo signo.



Ejemplo 7: Resolver esta ecuación: $y = x^5 + x^3 + x - 1 \equiv f(x) = 0$

$$f'(x) = 5x^4 + 3x^2 + 1 > 0 \quad \forall x \Rightarrow f(x) \text{ creciente en } (0,1).$$

Tenemos que: $(x, f(x)) = \{ (0, -1), (1, 2), (0.5, 0.328125), (0.25, -0.73339) \}$.

$$x_i = x_{i-1} - \frac{\overbrace{f(x_{i-1})}^{\text{tangente}}}{f'(x_{i-1})} \longrightarrow \begin{cases} x_0 = 1/3 \\ x_1 = 1/3 - \frac{(1/3)^5 + (1/3)^3 + 1/3 - 1}{5 * (1/3)^4 + 3 * (1/3)^2 + 1} \\ x_2 = x_1 - \frac{x_1^5 + x_1^3 + x_1 - 1}{5 * x_1^4 + 3 * x_1^2 + 1} \end{cases}$$

b.- Convergencia y error:

Podemos definir una sucesión $\{p_n\}$ que converge a p , así:

$$p_n \rightarrow p \Leftrightarrow \lim_{n \rightarrow \infty} \frac{|p_{n+1} - p|}{|p_n - p|^\alpha} = \lambda$$

donde α es el orden de convergencia (rapidez del método) y λ el error asintótico / $\alpha > 0 \wedge \lambda > 0$.

$$+ \text{velocidad} \Downarrow \left\{ \begin{array}{l} a_1 = \left\{ \frac{1}{n} \right\} \rightarrow 0, \alpha = 1, \lambda = 1 \\ a_2 = \left\{ \frac{1}{n^2} \right\} \rightarrow 0, \alpha = 1, \lambda = 1 \\ a_3 = \left\{ \frac{1}{2^n} \right\} \rightarrow 0, \alpha = 1, \lambda = 1 \\ a_4 = \left\{ \frac{1}{2^{2^n}} \right\} \rightarrow 0, \alpha = 2, \lambda = 1 \\ a_5 = \left\{ \frac{1}{3^n} \right\} \rightarrow 0, \alpha = 1, \lambda = 1/3 \end{array} \right.$$

Nota 10: El valor a_4 equivale a la velocidad de convergencia del método de Newton. Es de convergencia cuadrática, dado que el error en una iteración es el cuadrado del de la anterior.

$$\left. \begin{array}{l} E_{n+1} = 2^{-2^{n+1}} \\ E_n = 2^{-2^n} \end{array} \right\} \Rightarrow E_{n+1} = E_n^2 \qquad \left. \begin{array}{l} E_{n+1} = |p_{n+1} - p| \\ E_n = |p_n - p| \end{array} \right\} \Rightarrow E_{n+1} \approx \lambda * E_n^\alpha$$

2.- Método de la Secante:

Es un método de orden $\varphi = \frac{1+\sqrt{5}}{2} = 1.6180339$, valor bautizado como *Porción Áurea*.

Nota 11: La *Porción Áurea* es también conocida como el *número de oro de la Ciencia* (ver anotación en tema misceláneo y anexos).

Aunque es de menor orden que el de Newton, en realidad converge mucho más rápido (en la mitad de iteraciones que Newton). Cuenta además con la notable ventaja de que nos permite operar sin conocer la derivada de la función en ningún punto.

El *método de la secante* tiene menor coste computacional que el de Newton y en realidad equivaldría a un orden de convergencia $\alpha=2.6$, porque sólo necesita la función, y no su función derivada.

Por lo demás, este método es idéntico al de Newton usado en la resolución de ecuaciones no lineales.

Tiene su base en el hecho de que la derivada de una función en un punto es próxima al cociente incremental (la secante de una función).

Si sabemos que la derivada es:

$$f'(x_n) = \frac{f(x_n) - f(x_{n-1})}{x_n - x_{n-1}},$$

entonces podemos evitar tratar con la derivada si escribimos p_{n+1} como:

$$p_{n+1} = p_n - \underbrace{\frac{f(p_n)(p_n - p_{n-1})}{f(p_n) - f(p_{n-1})}}_{\text{secante a 1 curva en 1 punto}}$$

3.- Método “Leap-frog”:

Novísimo, sencillo y pseudo-innovador, apareció en un simple artículo en la revista SIAM (*Society and industry applied Mathematic*: “Matemática aplicada para la sociedad y la industria”) en el año 2002.

Este método es una combinación entre los de Newton y Secante anteriores, a los que se añade un ingenioso cálculo que permite “saltarse” soluciones intermedias, cual si se tratase del salto de una ágil rana, para acercarse mucho más rápidamente a una solución.

Su orden de convergencia es cúbico ($\alpha=3$), pero en contra tiene un elevado coste computacional que debe sopesarse en problemas de cierta complejidad.

Este es el planteamiento general:

Se trata de obtener un dato de entrada adicional \tilde{p} para cada iteración. Este dato se logra mediante un la aplicación de Newton a un punto inicial p_0 . Una vez tengamos estas aproximaciones p_0 y \tilde{p} , se utiliza de nuevo el algoritmo de la secante para obtener el próximo punto solución p_1 .

$$\begin{aligned} p_0 \\ \tilde{p} &= p_0 - \frac{f(p_0)}{f'(p_0)} \\ p_1 &= \tilde{p} - \frac{f(\tilde{p})(\tilde{p} - p_0)}{f(\tilde{p}) - f(p_0)} = \dots \text{operando} \dots = p_0 + \frac{f^2(p_0)}{f'(p_0)[f(\tilde{p}) - f(p_0)]} \\ p_1 &= p_0 + \frac{f^2(p_0)}{f'(p_0) \left[f\left(p_0 - \frac{f(p_0)}{f'(p_0)}\right) - f(p_0) \right]} \end{aligned}$$

4.- Método de Horner (valor de un polinomio en punto)

Es un algoritmo recursivo que puede implementarse en varios pasos. Fácil de implementar y poco costoso computacionalmente, será el usado para calcular $f(x=p)$ y $f'(x=p)$.

Este método calcula fácilmente el valor de un polinomio en un punto, basándose en el de Ruffini para calcular raíces de polinomios. Tiene también fundamento en el teorema del resto.

Teorema 7: Teorema del resto: el valor numérico de un polinomio $P(x)$ para $x=x_0$, viene dado por el resto de la división de $P(x)$ entre $(x - x_0)$.

$$P(x) = (x - x_0) \cdot Q(x) + R$$

De donde se deduce que el polinomio se anula para $x=x_0$:

$$P(x_0) = R \quad / \quad x=x_0$$

Nota 12: Un polinomio es un vector, y no una ecuación. Tiene la forma:

$$P_n(x): A = [a_n, a_{n-1}, a_{n-2}, \dots, a_1, a_0] \equiv a_n x^n + a_{n-1} x^{n-1} + \dots + a_2 x^2 + a_1 x + a_0$$

El algoritmo de Horner original, o sea, de 1 paso, calcula el valor de un polinomio $P_n(x)$ en un único punto $x=p$.

Pero también podemos seguir iterando recursivamente, durante una serie de i pasos, y lograremos la siguiente sucesión de valores:

$$\text{Horner}_i(P_n(x=p)) = \left\{ P_n(p), \frac{P_n'(p)}{1!}, \frac{P_n''(p)}{2!}, \dots, \frac{P_n^{(i)}(p)}{i!} \right\}$$

Esta sucesión nos permite calcular también fácilmente el valor de las sucesivas derivadas de un polinomio, si se equipara a una función, en un punto p .

Si tenemos en cuenta que el método de Ruffini calcula esto:

$$a_n x^n + a_{n-1} x^{n-1} + \dots + a_2 x^2 + a_1 x + a_0 = (b_n x^{n-1} + b_{n-1} x^{n-2} + \dots + b_2 x + b_1)(x - x_0) + b_0,$$

calcularemos entonces las raíces del polinomio dado y redefiniremos los términos de la solución, de esta manera:

	a_n	a_{n-1}	...	a_2	a_1	a_0
X_0		$x_0 b_n$...	$x_0 b_3$	$x_0 b_2$	$x_0 b_1$
	a_n	$a_{n-1} + x_0 b_n$...	$a_2 + x_0 b_3$	$a_1 + x_0 b_2$	$a_0 + x_0 b_1$
Redefinimos	b_n	b_{n-1}	...	b_2	b_1	$b_0 = P(x_0)$

Ejemplo 8: Calculemos los valores pertinentes de $P_3(2) = x^3 - 2x^2 + 2x - 1$

	1	-2	0	2	-1
2		2	0	0	4
	1	0	0	2	$3 = P_n(2)$
2		2	4	8	
	1	2	4	$10 = P_n'(2)$	
2		2	8		
	1	4	$12 = [P_n''(2)]/2! \Rightarrow P_n''(2) = 12 \cdot 2!$		
2		2			
	1	$6 = [P_n'''(2)]/3! \Rightarrow P_n'''(2) = 6 \cdot 3!$			

Veamos ahora el algoritmo con una corrección útil.

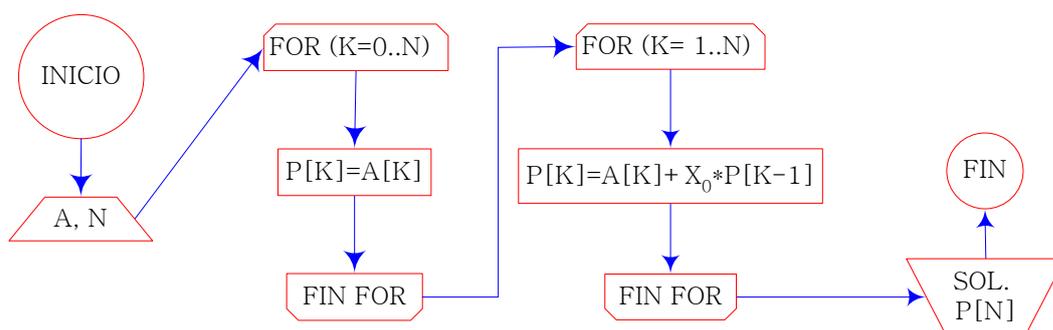
Algoritmo 6: HORNER DE 1 PASO

Calcula el valor de un polinomio $P_n(x)$ en un punto $x=p$ por el método de Ruffini.

El polinomio debe expresarse como un vector de coeficientes, y no como una ecuación; es decir, de este modo:

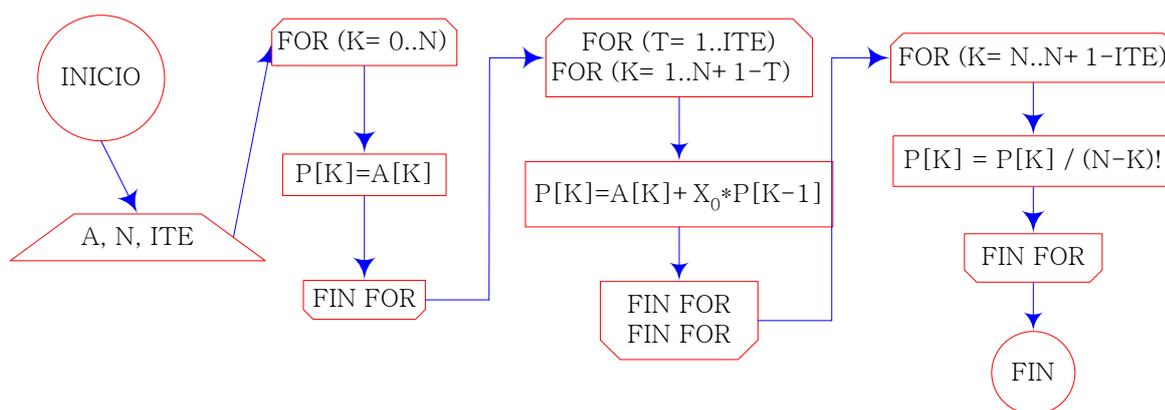
$$P_n(x): A=[a_n, a_{n-1}, a_{n-2}, \dots, a_1, a_0]$$

$$(a_n x^n + a_{n-1} x^{n-1} + \dots + a_2 x^2 + a_1 x + a_0)$$

**Algoritmo 7:** HORNER DE n PASOS

Calcula, dado un polinomio $P_n(x)$, la sucesión de valores siguiente en un punto $x=p$:

$$\text{Horner}_i(P_n(x=p)) = \{P_n(p), P_n'(p), P_n''(p), P_n'''(p), \dots, P_n^{ite}(p)\}$$



Nota 13: Los algoritmos que quedan aquí reflejados ya presentan la corrección pertinente a la división de factoriales que incomodan las soluciones que son ofrecidas por el método de Ruffini. Es conveniente no echar al olvido dicha modificación a la hora de implementar el método informáticamente.

Tema 3: Sistemas de ecuaciones lineales

A.- Álgebra matricial para sistemas lineales:

1.- Planteamiento de sistemas de ecuaciones lineales:

Según su tamaño, los sistemas se dividen en *pequeños*, con menos de 300 ecuaciones, que usan métodos directos que resuelven en pocos pasos; y *grandes*, con métodos iterados que crean una sucesión de vectores convergente a la solución, evitando efectos del error de redondeo.

Según su estructura, se dividen en *sistemas llenos*, con matriz no dispersa; y *sistemas dispersos*, con una matriz dispersa (*'sparse'*), por ejemplo, matrices tridiagonales y triangulares.

Def. 15: **Compatibilidad de un sistema de ecuaciones:** un sistema de ecuaciones es *incompatible* si no tiene solución, *compatible determinado* si posee una solución única y *compatible indeterminado* si posee varias soluciones.

Def. 16: **Sistema de ecuaciones homogéneo:** aquel cuyos términos independientes son todos nulos.

Def. 17: **Sistemas equivalentes:** son los sistemas que comparten las mismas soluciones.

Def. 18: **Matriz opuesta:** la matriz $-A$ es la opuesta de A si tiene todos sus elementos opuestos en el cuerpo de los números reales.

Def. 19: **Matriz traspuesta:** la matriz A^t o A^p es la traspuesta de la matriz cuadrada A si se obtiene intercambiando en A las filas por las columnas.

Def. 20: **Matriz dispersa o esparcida:** la que tiene alto porcentaje de elementos nulos.

Def. 21: **Matriz regular:** aquella cuya inversa existe.

Def. 22: **Matriz singular:** aquella que no tiene inversa.

Def. 23: **Matriz ortogonal:** aquella que tiene inversa y traspuesta coincidentes.

Def. 24: **Transformaciones elementales por filas** (análogamente por columnas):

- ◆ F_{ij} : intercambiar las filas i -ésima y j -ésima
- ◆ $F_{i(\lambda)}$: multiplicar la fila i -ésima por un número real no nulo λ
- ◆ $F_{ij(\lambda)}$: sumar a la fila i -ésima la fila j -ésima multiplicada por un número real no nulo λ

Teorema 8: Al aplicar una transformación elemental por filas en la matriz ampliada de un sistema de ecuaciones, obtenemos la matriz ampliada de un sistema equivalente al dado.

Tratamos de resolver un sistema de ecuaciones planteado de la forma $Ax=b$, tal que:

$$\begin{pmatrix} a_{11} & a_{12} & \dots & a_{1n} \\ a_{21} & a_{22} & \dots & a_{2n} \\ \dots & \dots & \dots & \dots \\ a_{n1} & a_{n2} & \dots & a_{nn} \end{pmatrix} \begin{bmatrix} x_1 \\ x_2 \\ \dots \\ x_n \end{bmatrix} = \begin{pmatrix} b_1 \\ b_2 \\ \dots \\ b_n \end{pmatrix}, \text{ donde: } \begin{cases} a_{11}x_1 + a_{12}x_2 + \dots + a_{1n}x_n = b_1 \\ a_{21}x_1 + a_{22}x_2 + \dots + a_{2n}x_n = b_2 \\ \dots \dots \dots \dots = \dots \\ a_{n1}x_1 + a_{n2}x_2 + \dots + a_{nn}x_n = b_n \end{cases}$$

Pueden tener 0, 1 ó infinitas soluciones según su linealidad y determinismo. Algunos sistemas incluso alcanzan solución manualmente con pocas ecuaciones.

Nota 14: Si $|A| \neq 0$ entonces el sistema tiene solución única.

Nota 15: El sistema matricial $Ax=b$, se resuelve en MATLAB con la orden

$$\gg x=A \setminus b.$$

Pero en otros lenguajes no se usarán métodos directos de resolución, porque requieren cálculos de orden $2n^3/3$. En estos casos utilizaremos métodos iterativos de convergencia.

Los métodos iterativos buscan aproximaciones progresivas con diferentes estrategias de cálculo, aunque no todos convergen siempre con todos los sistemas que se planteen.

2.- Normas:

Para medir y comparar vectores se usan normas, sobre todo las normas $/1$, $/2$ y $/\infty$.

Def. 25: Longitud o norma de un vector $x = (x_1, x_2, \dots, x_n)^t \in \mathbb{R}^n$:

$$\left\{ \begin{array}{l} \text{norma 1:} \\ \text{norma 2 o norma euclídea:} \\ \text{norma infinito } (\infty): \end{array} \right. \quad \begin{cases} \|x\|_1 = \sum_{i=1}^n |x_i| \\ \|x\|_2 = \sqrt{\sum_{i=1}^n |x_i|^2} \\ \|x\|_\infty = \max_{i=1..n} |x_i| \end{cases}$$

Def. 26: Norma de Frobenius de una matriz: se define de manera análoga a la norma euclídea de vector, si consideramos una matriz como un vector de $m \times n$ coordenadas:

$$\|A\|_F = \sqrt{\sum_{i,j} |a_{ij}|^2} = \sqrt{\text{tr} A^* A}$$

Def. 27: Proximidad: 2 vectores son próximos si la norma de su diferencia es pequeña, o sea, si los separa una longitud mínima:

$$x, y \in \mathbb{R}^n, x \approx y \Leftrightarrow \|x - y\|_i < \text{TOLERANCIA.}$$

Ejemplo 9: Para $x=(1,-2,-1)^t$, entonces: $\|x\|_1 = 4$; $\|x\|_2 = \sqrt{6}$; $\|x\|_\infty = 2$

Def. 28: Convergencia en espacios normados: una sucesión de vectores $\{v_i\}$ de un espacio vectorial normado $(V, \|\cdot\|)$ converge a un vector v si: $\lim_{k \rightarrow \infty} \|v_k - v\| = 0$

3.- Caracterización de una matriz:

Def. 29: **Matriz diagonalizable:** una matriz cuadrada A es diagonalizable si existe una matriz diagonal D semejante a ella. Es decir, con la matriz regular P denominada *matriz de paso*, tal que:

$$D = P^{-1}AP$$

Def. 30: **Par autovalor-autovector de una matriz:** Sean $A_{n \times n}$ y X_n , podemos definir un par *autovalor-autovector* (λ, X) de A , (también conocidos como *valor propio* o *característico* y *vector propio* o *característico* respectivamente) $\Leftrightarrow A * X = \lambda * X$, o lo que es lo mismo:

$$(A - \lambda I)X = 0 \quad / \quad X \neq 0,$$

siendo:

$$A = \begin{pmatrix} a_{11} & a_{12} & \dots & a_{1n} \\ a_{21} & a_{22} & \dots & a_{2n} \\ \dots & \dots & \dots & \dots \\ a_{n1} & a_{n2} & \dots & a_{nn} \end{pmatrix}; \quad X = \begin{bmatrix} x_1 \\ x_2 \\ \dots \\ x_n \end{bmatrix}$$

Def. 31: **Radio espectral de una matriz:** el máximo de sus autovalores en valor absoluto:

$$\rho(A) = \max\{|\lambda| \mid \lambda \in C_A \text{ autovalor de } A\}$$

Geoméricamente, el radio espectral representa el radio del círculo mínimo que contiene a todos los autovalores de la matriz

Ejemplo 10: Para $A = (1, 2, 1-i, 1+i)$, el radio espectral es $\rho(A) = \max\{1, 2, \sqrt{2}, \sqrt{2}\} = 2$

Def. 32: **Polinomio característico de una matriz cuadrada A :** polinomio definido por:

$$\rho(A) = |A - \lambda I|$$

Teorema 9: Los **autovalores** de una matriz son los ceros de la función que se construye con su *polinomio característico*.

Teorema 10: Matriz diagonalizable: Una matriz cuadrada de orden n es diagonalizable si y sólo si pueden encontrarse n autovectores linealmente independientes.

Def. 33: **Norma natural de una matriz cuadrada $A_{n \times n}$:** $\|A\| = \max_{\|x\|=1} \|Ax\|$

Teorema 11: Si $A_{n \times n} = (a_{ij})$, entonces:

$$\|A\|_{\infty} = \max_{1 \leq i \leq n} \sum_{j=1}^n |a_{ij}|$$

Teorema 12: Para una matriz cuadrada A , tenemos que:

$$\sqrt{\rho(A^t A)} = \|A\|_2 \quad \text{y} \quad \rho(A) \leq \|A\| \quad / \quad \forall \quad \| \cdot \| \text{ norma natural.}$$

Def. 34: Matriz cuadrada A convergente: una matriz cuadrada es convergente si se cumple que

$$\lim_{k \rightarrow \infty} (A^k)_{ij} = 0 \quad / i = 1..n ; j = 1..n .$$

Esto equivale a decir que su polinomio característico es

$$\rho(A) < 1 \text{ o bien que } \lim_{n \rightarrow \infty} A^n x = 0 \quad / \forall x$$

Teorema 13: Propiedades de la norma de un vector:

$\forall \alpha \in \mathbb{R} \wedge \forall x, y \in \mathbb{R}^n$ (también se cumplen para matrices de orden $n \times n$)

1. $\|x\| \geq 0$, tal que $\|x\| = 0 \Leftrightarrow x = (0, 0, \dots, 0)^t \equiv 0$
2. $\|\alpha x\| = |\alpha| \cdot \|x\|$
3. $\|x+y\| \leq \|x\| + \|y\|$
4. $\|x\|_\infty \leq \|x\|_2 \leq n^{1/2} \cdot \|x\|_\infty$
5. $\|x \cdot y\| \leq \|x\| \cdot \|y\|$
6. Para λ compleja, tenemos que $\lambda = \alpha + \beta i$, $|\lambda| = \sqrt{\alpha^2 + \beta^2}$

B.- Planteamiento de los algoritmos iterativos:

Genéricamente, los métodos iterados son *métodos de descomposición*, dado que descomponen la matriz de coeficientes. Para obtener un buen algoritmo, debemos lograr transformar el sistema de ecuaciones original en otro con el que poder iterar más rápidamente en el ordenador.

Existen diversas maneras de llevar a cabo esta transformación, de manera que esto es lo que diferencia a los distintos algoritmos iterativos, aunque todos tengan el mismo sistema de ecuaciones de origen y confluyan a un sistema de ecuaciones equivalente al inicial.

$$Ax = b \rightarrow \dots \rightarrow X = Tx + c$$

La base de los algoritmos está en el *teorema del Punto Fijo* contemplado en el tema anterior, que consistía en expresar la ecuación de la forma $x = \varphi(x)$ exigiendo que $\varphi'(x) < q < 1$ para cualquier valor x del intervalo donde se busca la solución.

En el caso de sistemas de ecuaciones lineales, de la forma $Ax = b$, trataremos de buscar métodos iterativos de forma análoga a como se hace con las ecuaciones, es decir, transformando el sistema en otro equivalente de la forma $x = F(x)$, donde $F(x) = Mx + N$, en la mayoría de los casos, se trabaja con *factorización LU* de la matriz A .

Evidentemente, la matriz M tendrá algunas condiciones según los conceptos de normas vectoriales y matriciales, como se exige para avalar la convergencia que $\varphi'(x) < q < 1$ en ecuaciones.

Def. 35: **Sistemas de ecuaciones lineal en su componente k -ésima:** el que verifica:

$$f(x_1, \dots, x_{k-1}, \alpha * x_k^1 + \beta * x_k^2, x_{k+1}, \dots, x_m) = \\ = \alpha * f(x_1, \dots, x_{k-1}, x_k^1, x_{k+1}, \dots, x_m) + \beta * f(x_1, \dots, x_{k-1}, x_k^2, x_{k+1}, \dots, x_m)$$

Def. 36: **Sistema de ecuaciones lineal:** el que es lineal en todas sus componentes. Un sistema lineal puede dividirse de modo que: $M=A + iB$.

Def. 37: **Sistema lineal bien condicionado:** aquel en el que los errores cometidos en los datos A y b producen un error del mismo orden en la solución x ; es decir, que un sistema $Ax=b$ está bien condicionado si la solución exacta apenas diverge de la solución aproximada.

Def. 38: **Número de condición de una matriz A :** $\kappa(A)$: para una matriz regular A , permite conocer la existencia de un buen condicionamiento del sistema $Ax=b$. Siendo κ una cota de la norma de la matriz inversa, y las normas multiplicativas, se define como

$$\kappa(A) = \|A\| \cdot \|A^{-1}\| < \kappa \|A\|$$

Teorema 14: **Algoritmo iterativos para la aproximación a una solución de sistemas de ecuaciones lineales:** $X^{(0)} \in \mathcal{R}^n$, dato inicial:

$$X^{(k)} = TX^{(k-1)} + c,$$

Este algoritmo converge a la raíz del sistema (0) si:

$$\rho(T) < 1 \quad \forall X^{(0)} \in \mathcal{R}^n$$

Def. 39: **Matriz de Hodamard o de diagonal dominante:** una matriz cuadrada A de orden n , $A = (a_{ij})_{\substack{i=1,2,\dots,n \\ j=1,2,\dots,n}}$ es:

- ◆ de diagonal dominante por filas si: $|a_{ii}| > \sum_{k=1, k \neq i}^n |a_{ik}|$, $i = 1, 2, \dots, n$
- ◆ de diagonal dominante por columnas si: $|a_{ii}| > \sum_{k=1, k \neq i}^n |a_{ki}|$, $i = 1, 2, \dots, n$

Teorema 15: Toda matriz diagonal dominante es regular, y admite factorización LU.

Def. 40: **Matriz estrictamente diagonal dominante por filas:** $\Leftrightarrow |a_{ii}| > \sum_{j \neq i} |a_{ij}|$, $\forall i$.

Ejemplo 11: A es de diagonal dominante por filas, pero no por columnas:

$$A = \begin{pmatrix} 3 & 1 & 1 \\ 0 & 2 & 1 \\ 2 & -1 & 5 \end{pmatrix}$$

C.- Métodos iterados:

1.- Método de Jacobi:

Jacobi descompuso la matriz de coeficientes en 3 componentes diferenciadas, que son su diagonal principal [$D \equiv \text{diagonal}$], y las partes triangulares superior [$U \equiv \text{upper}$] e inferior [$L \equiv \text{lower}$].

$$A = D + L + U.$$

Tras varias operaciones, el método resulta:

$$Dx = (L+U)x + b \Rightarrow x = D^{-1}(L+U)x + D^{-1}b.$$

Esto da lugar a la forma matricial siguiente:

$$x^{(k)} = D^{-1}(L+U)x^{(k-1)} + D^{-1}b \quad / k = 1, 2, \dots$$

y, finalmente, la matriz de Jacobi J queda:

$$J = D^{-1}(L+U) = D^{-1}(D-A) = I - D^{-1}A$$

Es decir, que una iteración se calcula:

$$x^{(k)} = Jx^{(k-1)} + D^{-1}b \quad / k = 1, 2, \dots$$

Teorema 16: El método de Jacobi converge si la matriz A es de diagonal dominante.

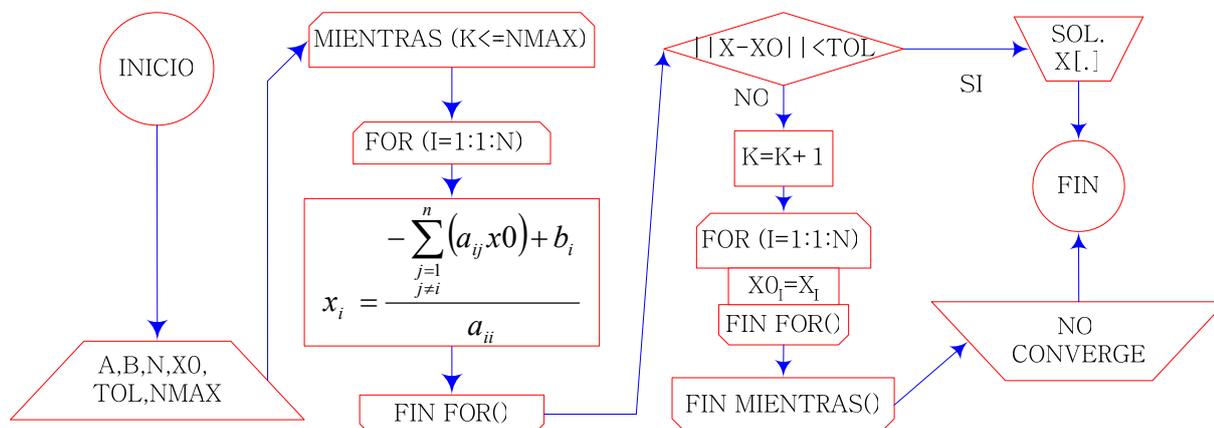
Siempre que $a_{ii} \neq 0$, este método resuelve la i -ésima ecuación de $Ax=b$ para x_i en la forma:

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

y genera cada $x_i^{(k)}$ de la componente $x^{(k-1)}$ para $k \geq 1$, con

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

Algoritmo 8: JACOBI ITERATIVO:



2.- Método de Gauss-Seidel:

Con una descomposición matricial similar a la anterior, este método obtiene esto:

$$x^{(k)} = (D-L)^{-1}Ux^{(k-1)} + (D-L)^{-1}b \quad / k = 1, 2, \dots$$

y define la matriz Gauss-Seidel como

$$L_1 = (D-L)^{-1}U = (A+U)^{-1}(A+U-A) = I - (A+U)^{-1}A = I - (D-L)^{-1}A$$

Es decir, que cada iteración se calcula con:

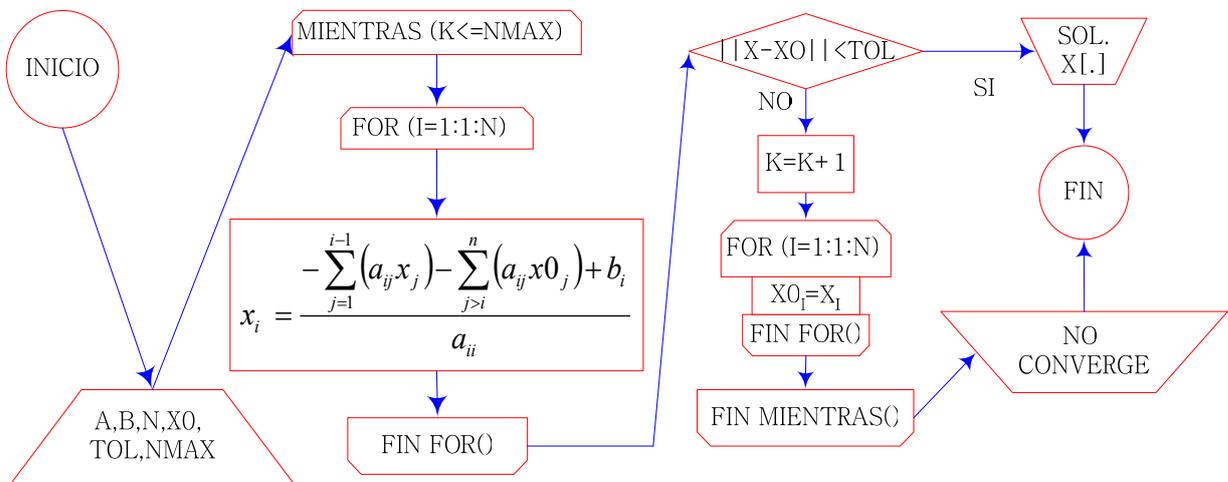
$$x^{(k)} = L_1 x^{(k-1)} + D^{-1}b \quad / k = 1, 2, \dots$$

Teorema 17: El método Gauss-Seidel converge si A es de diagonal estrictamente dominante

Algoritmo 9: GAUSS-SEIDEL ITERATIVO:

Este método resuelve, con $a_{ii} \neq 0$, la i -ésima ecuación de $Ax=b$ para x_i en la forma:

$$x_i^{(k)} = \frac{b_i - \sum_{j=1}^{i-1} (a_{ij} x_j^{(k)}) - \sum_{j>i}^n (a_{ij} x_j^{(k-1)})}{a_{ii}} \quad / i = 1, \dots, n$$



Teorema 18: La sucesión $\{x^{(k)}\}_{k=0..∞}$ en $X^{(k)} = TX^{(k-1)} + c$ converge al vector $x \in \mathcal{R}^n$, $/ \forall x^{(0)} \in \mathcal{R}^n$, si $\|T\| < 1$ para toda norma matricial natural.

Además se cumplen estas cotas de error:

$$\|x - x^{(k)}\| \leq \|T\|^k \|x^{(0)} - x\|$$

y

$$\|x - x^{(k)}\| \leq \frac{\|T\|^k}{1 - \|T\|} \|x^{(1)} - x^{(0)}\|$$

Teorema 19: Teorema de Stein-Rosenberg: si $a_{ij} \leq 0 \ / i \neq j$, y $a_{ii} >= 1 \ / i = 1, \dots, n$, entonces se satisface una y solamente una de las siguientes afirmaciones:

- ◆ $0 < \rho(T_{GS}) < \rho(T_J) < 1$
- ◆ $1 < \rho(T_J) < \rho(T_{GS})$
- ◆ $\rho(T_J) = \rho(T_{GS}) = 0$
- ◆ $\rho(T_J) = \rho(T_{GS}) = 1$

Nota 16: Debe escogerse el método, sea Jacobi o Gauss-Seidel, con menor $\rho(T)$, ya que la relación de la rapidez de convergencia con el radio espectral sería:

$$\|x^{(k)} - x\| \approx \rho(T)^k \|x^{(0)} - x\|$$

3.- Método generalizado de Jacobi-Gauss-Seidel:

Si unimos la base teórica de los dos métodos anteriores en una sola fórmula, tendríamos una generalización que nos proporcionaría mejores resultados en cualquier caso. Se toman simultáneamente $x^{(0)}$ y $x^{(1)}$, pero con distintos pesos

$$[\omega]: \quad 1 - \omega(x_1^{(k-1)}, \dots) \quad \text{y} \quad \omega(x_1^{(k)}, \dots)$$

$$x^{(k)} = (D - \omega L)^{-1}((1 - \omega)L + U)x^{(k-1)} + (D - \omega L)^{-1}b \quad / k = 1, 2, \dots$$

4.- Métodos SOR (Successive Over-Relaxation):

Los métodos de *sobre-relajación sucesiva* (S.O.R.) realizan una descomposición muy especial. El sistema $Ax=b$ se perfila ahora así:

$$\omega Ax = \omega b, \ / \omega \neq 0, \text{ con } 0 < \omega < 2$$

Def. 41: Vector residual r de una aproximación \tilde{x} al sistema $Ax=b$:

$$r = b - A\tilde{x} \ / \ \tilde{x} \in \mathbb{R}^n$$

Teorema 20: Es condición de convergencia que : $\omega \in (0, 2)$.

$$(\omega D - \omega L - \omega U)x = \omega b \Rightarrow \dots \Rightarrow (D - \omega L)x = ((1 - \omega)L + \omega U)x + \omega b \quad / k = 1, 2, \dots$$

O sea, multiplican a ambos lados de la ecuación ω , de forma que al final les queda esto:

$$x^{(k)} = (D - \omega L)^{-1}((1 - \omega)L + \omega U)x^{(k-1)} + \omega(D - \omega L)^{-1}b \quad / k = 1, 2, \dots$$

Así, la matriz del método de SOR, queda: $L_\omega = (D - \omega L)^{-1}((1 - \omega)L + \omega U)$

- ◆ Si $\omega = 1$, este método se transforma en el de Gauss-Seidel.
- ◆ Si $\omega > 1$, ahora es de sobre-relajación.
- ◆ Si $\omega < 1$, entonces es de sub-relajación.

Def. 42: **Matriz tri-diagonal simétrica:** aquella donde $a_{i+1,i} = a_{i,i+1}$ y el resto de a_{ij} nulos.

Nota 17: Para recorrer una matriz simétrica sólo se recorre la diagonal y una parte triangular. Se usan 2 bucles *for()*. Por ejemplo, $D+L$: *for i, for j / j=1..i*. Si la matriz fuese tri-diagonal simétrica, sólo se necesita un bucle *for()* que recorra a_{ii} y $a_{i,i+1}$.

Def. 43: **Matriz singular:** es aquella que tiene diagonal principal no nula. Las matrices no singulares, se puede alterar permutando filas libremente.

Nota 18: La ordenación de la diagonal de la matriz de coeficientes se hará de modo decreciente por filas en valor absoluto, tal que en la 1ª fila esté el máximo de la 1ª columna, y así los a_{ii} decrecerán sucesivamente. No hay que olvidar permutar el vector de la variable independiente b .

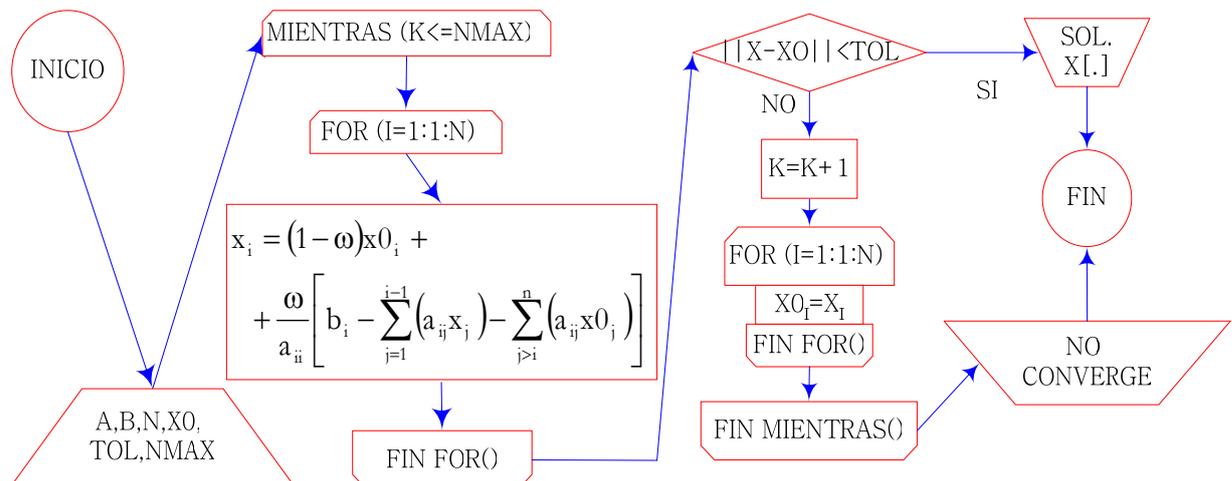
Teorema 21: Si A es de diagonal dominante, el método es convergente para $\omega \in (0, 1)$.

Teorema 22: Si A es simétrica y definida positiva, el método converge para $\omega \in (0, 2)$.

Algoritmo 10: MÉTODOS ITERATIVOS DE SOBRE-RELAJACIÓN SUCESIVA:

Siempre que $a_{ii} \neq 0$, este método resuelve la i -ésima ecuación de $Ax=b$ para x_i en la forma:

$$x_i^{(k)} = (1 - \omega)x_i^{(k-1)} + \frac{\omega}{a_{ii}} \left[b_i - \sum_{j=1}^{i-1} (a_{ij}x_j^{(k)}) - \sum_{j>i} (a_{ij}x_j^{(k-1)}) \right] \quad / i = 1, \dots, n$$



Tema 4: Sistemas de ecuaciones no lineales

La solución a un sistema de ecuaciones no lineal tampoco es fácil, incluso computacionalmente. Habrá que estudiar distintas maneras, modos iterativos y echar mano de otros conocimientos alternativos, como por ejemplo, ecuaciones importantes de la Física, parte fundamental de la ciencia Matemática.

A.- Método de Newton global:

Esta vez, Newton plantea un método iterativo para la resolución de problemas donde deban tenerse en cuenta importantes sistemas de ecuaciones no lineales. Estos sistemas deben poder escribirse en una forma vectorial $f(\mathbf{X})$:

$$\{ f(xy)=0, \quad g(xy)=0 \} \Rightarrow h(x,y)=f^2(xy)+g^2(xy) \geq 0 \quad / \forall x,y$$

$$h(\mathbf{X}) = \sum f_i^2(x_1, \dots, x_n) = F_1^2 + F_2^2 + \dots + F_n^2 \quad \text{para } n \text{ ecuaciones } \mathbf{X} = (x_1, \dots, x_n)$$

Newton, haciendo uso ahora de su faceta de estudioso de la Física, calcula la velocidad de la función con v , de esta forma:

$$\vec{v} = -DF(x_0) / F(x_0)$$

Podemos definir también h como $h = x - \tilde{x}$, y haciendo uso del desarrollo de Taylor, obtenemos que:

$$0 = f(x) = f(\tilde{x} + h) \approx f(\tilde{x}) + h \cdot f'(\tilde{x}),$$

de donde

$$h = -f(\tilde{x}) / f'(\tilde{x})$$

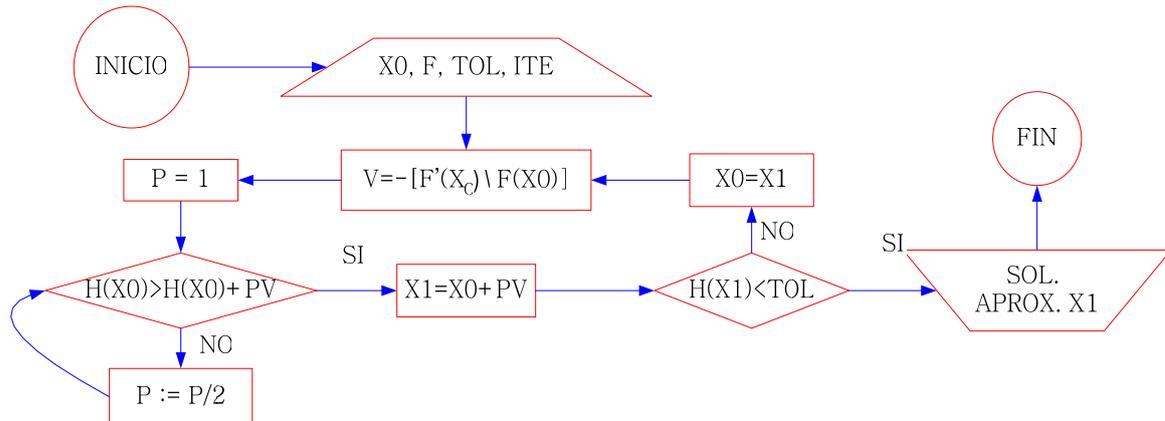
y por tanto

$$x \approx \tilde{x} - \frac{f(\tilde{x})}{f'(\tilde{x})}$$

Y, partiendo de $x_0 = \tilde{x}$, la aproximación que se usaría como valor x_{n+1} para iterar es:

$$x_{n+1} \approx x_n - \frac{f(x_n)}{f'(x_n)}$$

Elijamos la fórmula que sea, el método basado en el de Newton-Raphson-Fourier, que diseñó Newton ahora para sistemas de ecuaciones no lineales, calcula en cada iteración la solución a una ecuación. Se trata de valerse de la posibilidad de iterar sobre sí mismo para hallar sucesivamente las soluciones a todas las ecuaciones.

Algoritmo 11: NEWTON GLOBAL

Nota 19: Método de aproximación: Podríamos calcular el mínimo de una función altura “ $h(x)$ ” como: $h(X) = f^2(X) + g^2(X)$. Esta función debe representarse gráficamente con un malleado suficientemente fino para poder visualizar el mínimo de h . En realidad se necesitan métodos mucho más rápidos que este para trabajar eficazmente en problemas de gran complejidad.

B.- Método del gradiente o máximo descenso:

El gradiente es un vector que da la dirección del máximo ascenso, o descenso si es negativo, de una función en el espacio vectorial.

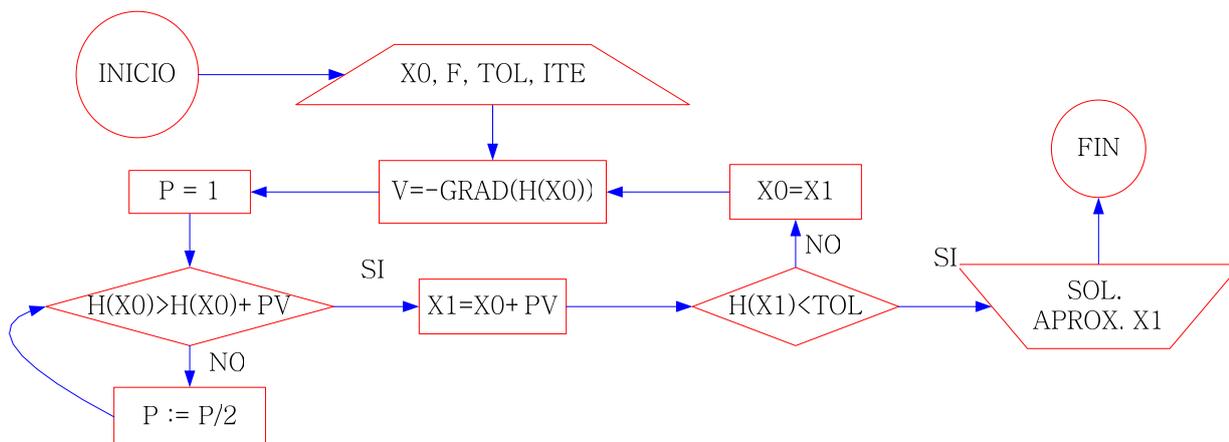
Con un algoritmo similar al de Newton para sistemas no lineales, plantea un cálculo basado en la *ecuación de la energía de Lyapunov*. El cero de esta función vectorial que contiene el sistema de ecuaciones no lineales implícito, está en un mínimo (o máximo, si se trata del gradiente positivo) de dicha ecuación de la energía. Ésta viene siempre dada en función de derivadas parciales:

$$\vec{v} = -\nabla h(x_0) = \frac{-\partial h(x_0)}{\partial x_0}$$

Dado que no llega a superar al método de Newton en velocidad, este método sólo se usa a fin de lograr dar una aproximación bastante fiable a un método de orden superior. La diferencia principal con el de Newton estriba en la necesidad de cálculos mucho más complejos.

Cabe advertir también que resulta inherentemente indispensable un ajustado análisis gráfico para acotar suficientemente el intervalo que contiene la raíz buscada. De no realizar este procedimiento, los cálculos pueden centrarse en algún mínimo o máximo local; es decir, que el método puede llegar a divergir bastante de la solución.

Algoritmo 12: GRADIENTE O MÁXIMO DESCENSO



Por ejemplo, este es un sistema de ecuaciones no lineal bastante sencillo:

$$\{ x^2 + 2xy - y^2 \operatorname{sen} x - 7 = 0 ; \quad x - 2y + x^2 y^2 - 4y^2 = 0 \}$$

Para sistemas no lineales es obligado acercarse iterativamente a una solución aproximada.

$$\left\{ \begin{array}{l} f_1(x_1, x_2, \dots, x_n) = 0 \\ f_2(x_1, x_2, \dots, x_n) = 0 \\ \dots \\ f_n(x_1, x_2, \dots, x_n) = 0 \end{array} \right\} \Rightarrow \left\{ \begin{array}{l} x_1 = g_1(x_1, x_2, \dots, x_n) \\ x_2 = g_2(x_1, x_2, \dots, x_n) \\ \dots \\ x_n = g_n(x_1, x_2, \dots, x_n) \end{array} \right\}$$

C.- Método iterativo del Punto Fijo:

Teorema 23: Sea A un conjunto cerrado en \mathbb{R}^n y $G: A \rightarrow A$ una función continua donde

$$\|G(x) - G(y)\| < L \|x - y\| \quad \forall \exists L \in [0, 1)$$

entonces G tienen un único punto fijo $p \in A$, que verifica:

$$(p \equiv PF) \quad \begin{cases} p_0 \in A \\ p_{k+1} = G(p_k) \quad / \quad k \geq 0 \end{cases}$$

Dicho punto fijo converge a p , y además se cumple que:

$$\|p_k - p\| \leq \frac{L}{1-L} \|p_k - p_{k-1}\| \leq \frac{L^k}{1-L} \|p_0 - p_1\| \quad / \quad k \geq 1$$

Nota 20: La constante L se llama constante de *Lipschitz* para G .

Algoritmo 13: PUNTO FIJO ITERATIVO:

$$X^{(0)} = (x_1^{(0)}, x_2^{(0)}, \dots, x_n^{(0)}) \text{ datos}$$

$$X_i^{(k+1)} = g_i(x_i^{(k)}, x_i^{(k)}, \dots, x_i^{(k)}) \quad / k \geq 0$$

D.- Método de Seidel:

Similar al anterior, usa los nodos del mismo orden. Es escalar pero converge mejor.

$$X^{(0)} = (x_1^{(0)}, x_2^{(0)}, \dots, x_n^{(0)}) \text{ datos}$$

$$X_1^{(k+1)} = g_1(x_1^{(k)}, x_2^{(k)}, \dots, x_n^{(k)})$$

$$X_2^{(k+1)} = g_2(x_1^{(k+1)}, x_2^{(k)}, \dots, x_n^{(k)})$$

$$X_3^{(k+1)} = g_3(x_1^{(k+1)}, x_2^{(k+1)}, x_3^{(k)}, \dots, x_n^{(k)})$$

$$\dots$$

$$X_n^{(k+1)} = g_n(x_1^{(k+1)}, x_2^{(k+1)}, x_3^{(k+1)}, \dots, x_{n-1}^{(k+1)}, x_n^{(k)}) \quad / k \geq 0$$

Ejemplo 12: Resuelve el siguiente sistema de ecuaciones con el método del punto fijo:

$$\begin{cases} 3x_1 - \cos(x_2 x_3) - \frac{1}{2} = 0 \\ x_1^2 - 81(x_2 + 0.1)^2 + \operatorname{sen}(x_3) + 1.06 = 0 \\ e^{-x_1 x_2} + 20x_3 + \frac{10\pi - 3}{3} = 0 \end{cases}$$

Tomamos:

$$G(X) = (g_1(x_1, x_2, x_3), g_2(x_1, x_2, x_3), g_3(x_1, x_2, x_3)), \text{ con}$$

$$g: [-1, 1]^3 \rightarrow [-1, 1]^3 \text{ y } |\partial g_i(x) / \partial x_i| \leq 0.281$$

Por tanto se cumple el teorema del punto fijo para

$$A = [-1, 1]^3 \text{ y } L = 0.843$$

Una función G posible sería la siguiente: (despejando $x_i = g(x) \dots$)

$$\begin{cases} x_1^{(k)} = \frac{1}{3} \left(\cos(x_2^{(k-1)} x_3^{(k-1)}) + \frac{1}{2} \right) \\ x_2^{(k)} = \frac{1}{9} \sqrt{(x_1^{(k-1)})^2 + \operatorname{sen}(x_3^{(k-1)}) + 1.06} - 0.1 \\ x_3^{(k)} = \frac{-1}{20} \left(e^{-x_1^{(k-1)} x_2^{(k-1)}} - \frac{10\pi - 3}{3} \right) \end{cases}$$

Y este algoritmo converge a la solución para cualquier

$$x^{(0)} \in [-1, 1]^3$$

E.- Método iterativo de Newton:

Def. 44: **Matriz jacobiana (de 3 funciones):** Sean $f_1(x, y, z)$, $f_2(x, y, z)$ y $f_3(x, y, z)$, tres funciones de tres variables independientes $\{x, y, z\}$, entonces su matriz jacobiana $J(x, y, z)$ es:

$$J(x, y, z) = \begin{pmatrix} \frac{\partial f_1}{\partial x} & \frac{\partial f_1}{\partial y} & \frac{\partial f_1}{\partial z} \\ \frac{\partial f_2}{\partial x} & \frac{\partial f_2}{\partial y} & \frac{\partial f_2}{\partial z} \\ \frac{\partial f_3}{\partial x} & \frac{\partial f_3}{\partial y} & \frac{\partial f_3}{\partial z} \end{pmatrix}$$

Nota 21: Por razones de espacio, construiremos el método de Newton-Raphson en el caso bidimensional. Consideramos el sistema de vectores $\{u, v\}$, con $u=f_1(x, y)$ y $v=f_2(x, y)$ con derivadas parciales continuas.

Si usamos la matriz jacobiana, podemos escribir una relación de la forma:

$$\begin{pmatrix} u - u_0 \\ v - v_0 \end{pmatrix} \approx \begin{pmatrix} \frac{\partial f_1(x_0, y_0)}{\partial x} & \frac{\partial f_1(x_0, y_0)}{\partial y} \\ \frac{\partial f_2(x_0, y_0)}{\partial x} & \frac{\partial f_2(x_0, y_0)}{\partial y} \end{pmatrix} * \begin{pmatrix} x - x_0 \\ y - y_0 \end{pmatrix}$$

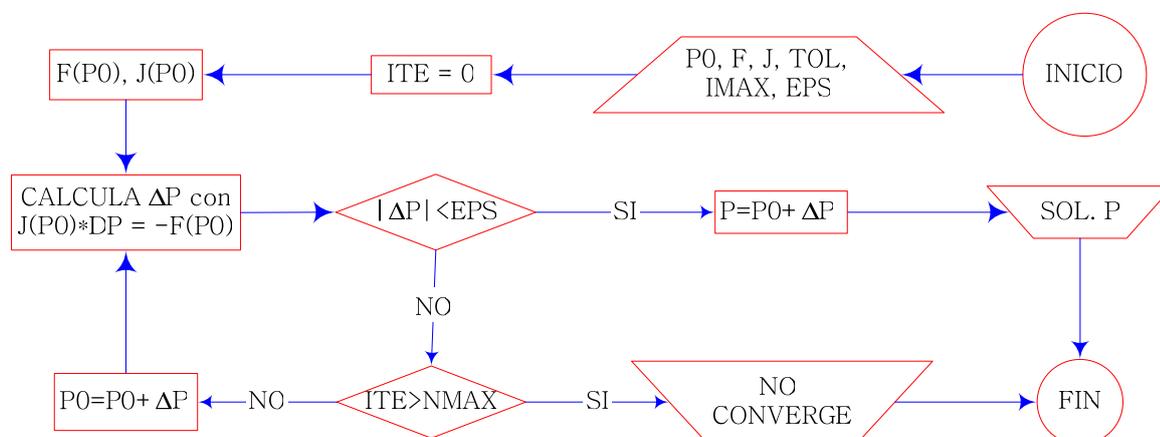
Si escribimos el sistema anterior como función vectorial, de la forma $V=F(X)$, nos queda:

$$\Delta F \approx J(x_0, y_0) \Delta X.$$

Y entonces podemos, tras varias operaciones, construir la siguiente iteración del método:

$$P_1 = P_0 + \Delta P = P_0 - J(p_0, q_0)^{-1} F(p_0, q_0)$$

Algoritmo 14: NEWTON PARA SISTEMAS DE ECUACIONES NO LINEALES



Tema 5: Interpolación polinomial

El proceso de ajustar una función a un conjunto de puntos dado se llama *interpolación*. En el caso de que queramos conocer el comportamiento de la función fuera del intervalo al que pertenece el conjunto soporte, el proceso se llama *extrapolación*; es decir, conocer el valor de la función, su tendencia, en algún punto fuera del intervalo conocido.

Def. 45: **Conjunto soporte:** conjunto de nodos o puntos $\{x_0, x_1, \dots, x_n\}$ sobre los que se evalúa una función $y=f(x)$. Tal conjunto es regular si sus nodos están equiespaciados.

Def. 46: **Interpolación:** si aproximamos un conjunto de puntos (x_i, y_i) por los que pasa una función $y=f(x)$ en un *conjunto soporte* $\{x_i\}$, por una función $g(x)$ que se ajuste a este conjunto, de modo que $g(x_i)=y_i$, se dice que la función $g(x)$ interpola a $f(x)$ en dicho conjunto soporte. Esta función $g(x)$ debe ser un polinomio, un cociente de polinomios, una función trigonométrica, o alguna expresión similar.

A.- Polinomio de Interpolación:

La interpolación puede hacerse con una función polinómica, de la forma $f(x) \equiv P_{n-1}(x)$, donde $f(x)$ se aproxima por un polinomio en x denominado “*polinomio de interpolación*”, que siempre existe y es único.

Teorema 24: Existencia y unicidad del polinomio de interpolación: dado el conjunto de números reales $\{x_0, \dots, x_n\} / x_0 < \dots < x_n$, y los $n+1$ números reales cualesquiera $\{y_0, \dots, y_n\}$, existe un único polinomio $P_n(x)$, tal que $P_n(x_i)=y_i$ para $i=0,1,\dots,n$.

Conocido el polinomio, calculamos las coordenadas deseadas con el método de Horner. Así, el cálculo del polinomio de interpolación se haría:

1) El vector Y de $n+1$ puntos se aproximaría:

$$[y_0, y_1, \dots, y_n] \equiv P_n(x) \equiv a_n x^n + a_{n-1} x^{n-1} + \dots + a_1 x + a_0$$

$$P_n(x): A = [a_n, a_{n-1}, a_{n-2}, \dots, a_1, a_0]$$

$$\begin{cases} a_n x_0^n + a_{n-1} x_0^{n-1} + \dots + a_2 x_0^2 + a_1 x_0 + a_0 = y_0 \\ a_n x_1^n + a_{n-1} x_1^{n-1} + \dots + a_2 x_1^2 + a_1 x_1 + a_0 = y_1 \\ \dots \dots \dots \\ a_n x_n^n + a_{n-1} x_n^{n-1} + \dots + a_2 x_n^2 + a_1 x_n + a_0 = y_n \end{cases}$$

2) Al invertir el orden $a_i x_i^j$, es un sistema matricial con los nodos x_i y las incógnitas a_i .

Podemos construir ahora la siguiente matriz de Vandermonde, para resolver como un sistema de ecuaciones lineales visto en el tema anterior.

$$\begin{pmatrix} x_0^n & x_0^{n-1} & \dots & x_0 & 1 \\ x_1^n & x_1^{n-1} & \dots & x_1 & 1 \\ \dots & \dots & \dots & \dots & \dots \\ x_{n-1}^n & x_{n-1}^{n-1} & \dots & x_{n-1} & 1 \\ x_n^n & x_n^{n-1} & \dots & x_n & 1 \end{pmatrix} \begin{pmatrix} a_0 \\ a_1 \\ \dots \\ a_{n-1} \\ a_n \end{pmatrix} = \begin{pmatrix} y_0 \\ y_1 \\ \dots \\ y_{n-1} \\ y_n \end{pmatrix}$$

Nota 22: Al introducir la matriz en un sistema informático, sólo se introduce la penúltima columna, ya que las otras se construyen iterativamente por potenciación de ésta en modo escalar.

Pueden escogerse métodos de interpolación polinomial más acertados que éste, como pudieran ser los métodos del *Polinomio interpolante de Lagrange* o las fórmulas de interpolación que definiera Newton, que incluyen a los métodos de *diferencias divididas* y de *diferencias finitas*.

B.- Polinomio de Interpolación de Lagrange:

Este es un método mejor que el anterior. Usa la fórmula introducida por Lagrange para el cálculo de polinomios a partir de un número de nodos conocidos.

Construimos polinomios que se anulen en todos los puntos conocidos (nodos) menos en el nodo x_i . Si los sumamos obtendremos un conjunto de polinomios que construyan un polinomio de interpolación en otros nodos no conocidos. El resultado es el polinomio interpolador de Lagrange que se muestra a continuación:

$$y = P_n(x) = L_n(x) = y_0 L_{n,0}(x) + y_1 L_{n,1}(x) + \dots + y_n L_{n,n}(x)$$

Con lo que el polinomio interpolador de Lagrange queda:

$$L_n(x) = y_0 \frac{(x-x_1)(x-x_2)\dots(x-x_n)}{(x_0-x_1)(x_0-x_2)\dots(x_0-x_n)} + y_1 \frac{(x-x_1)(x-x_2)\dots(x-x_n)}{(x_1-x_0)(x_1-x_2)\dots(x_1-x_n)} + \dots$$

$$\dots + y_n \frac{(x-x_0)(x-x_1)\dots(x-x_{n-1})}{(x_n-x_0)(x_n-x_1)\dots(x_n-x_{n-1})}$$

Es decir:

$$L_n(x) = y_0 \frac{\prod_{\substack{j=1..n \\ j \neq 0}} (x-x_j)}{\prod_{\substack{j=1..n \\ j \neq 0}} (x_0-x_j)}$$

El error cometido en la aproximación de una función por el polinomio de Lagrange es:

$$E_N(X) = f(X) - L_N(X) = \frac{(x-x_0)(x-x_1)\dots(x-x_N)f^{(N+1)}(c)}{(N-1)!} \quad / c = c(X), \quad c \in [a, b]$$

Ejemplo 13: Para el conjunto de nodos siguiente, el polinomio de interpolación es:

x	y
-1	4
1	4
2	7

$$L_2(x) = 4 \frac{(x-1)(x-2)}{(-1-1)(-1-2)} + 4 \frac{(x+1)(x-2)}{(1+1)(1-2)} + 7 \frac{(x+1)(x-1)}{(2+1)(2-1)} = x^2 + 3$$

Nota 23: Para multiplicar 2 polinomios, podemos implementar una rutina que lo haga correctamente. Si tenemos los polinomios con sus coeficientes:

$$P_n(p_0, p_1, \dots, p_n) * R_m(r_0, r_1, \dots, r_n) = Q_{m+n}(x)$$

$q_0 =$	$q_1 =$	$q_2 =$...	$q_m =$	$q_{m+1} =$	$q_{m+2} =$	$q_{m+3} =$	$q_{m+4} =$...
$p_0 r_0$	$p_0 r_1$	$p_0 r_2$...	$p_0 r_m$...
	$+p_1 r_0$	$+p_1 r_1$...	$+p_1 r_{m-1}$	$p_1 r_m$...
		$+p_2 r_0$...	$+p_2 r_{m-2}$	$+p_2 r_{m-1}$	$p_2 r_m$...
			...	$+p_3 r_{m-3}$	$+p_3 r_{m-2}$	$+p_3 r_{m-1}$	$p_3 r_m$...
					$+p_4 r_{m-3}$	$+p_4 r_{m-2}$	$+p_4 r_{m-1}$	$p_4 r_m$...

Una expresión algorítmica correcta y eficiente sería:

```
w=1; s=0; for (i=0..n) w=w(x-x(i)); // inicialización
for (i=0..n) { P=y(i);
               for (j=0..n) and (j≠i) P=P / (x(i)-x(j));
               S=S+P*w / (x-x(i));
            };
S=L_n(x) (* salida *)
```

C.- Interpolación de Newton:

1.- Método de las Diferencias Divididas:

Newton ideó una fórmula para calcular un polinomio interpolador que se aproxime a una serie de nodos de una función f desconocida. Este polinomio interpolador de Newton puede escribirse como:

$$P_n(x) = a_0 + a_1(x-x_0) + a_2(x-x_0)(x-x_1) + a_3(x-x_0)(x-x_1)(x-x_2) + \dots + a_n(x-x_0)(x-x_1) \dots (x-x_{n-1})$$

Así, el interpolar la función en el término x_1 , podemos escribir:

$$f(x_0) + a_1(x-x_0) = P_n(x_1) = f(x_1)$$

y despejamos a_1 como:

$$a_1 = \frac{f(x_1) - f(x_0)}{x_1 - x_0} \equiv f[x_1]$$

Los coeficientes de este polinomio se calculan como diferencias divididas.

Def. 47: **Notación de diferencia dividida:** la diferencia dividida cero de la función f con respecto a x_i , se denota por $f[x_i]$ y es simplemente la evaluación de f en x_i :

$$f[x_i] = f(x_i)$$

Def. 48: **Primera diferencia dividida** de f con respecto a x_i y x_{i+1} : $f[x_i, x_{i+1}]$, donde:

$$f[x_i, x_{i+1}] = \frac{f[x_{i+1}] - f[x_i]}{x_{i+1} - x_i}$$

Def. 49: La **k-ésima diferencia dividida** de f con respecto a $x_i, x_{i+1}, \dots, x_{i+k}$ es:

$$f[x_i, x_{i+1}, \dots, x_{i+k-1}, x_{i+k}] = \frac{f[x_{i+1}, x_{i+2}, \dots, x_{i+k}] - f[x_i, x_{i+1}, \dots, x_{i+k-1}]}{x_{i+k} - x_i}$$

Visto esto, el polinomio de interpolación anterior puede escribirse como:

$$P_n(X) = f[x_0] + f[x_0, x_1](x - x_0) + f[x_0, x_1, x_2](x - x_0)(x - x_1) + \dots + f[x_0, x_1, \dots, x_n](x - x_0)(x - x_1) \dots (x - x_{n-1})$$

Def. 50: **Fórmula de diferencia dividida interpolante de Newton:**

$$P_n(X) = f[x_0] + \sum_{k=1}^n f[x_0, x_1, \dots, x_k](x - x_0)(x - x_1) \dots (x - x_{k-1})$$

Iterando se construye una tabla con las sucesivas diferencias divididas de una función f .

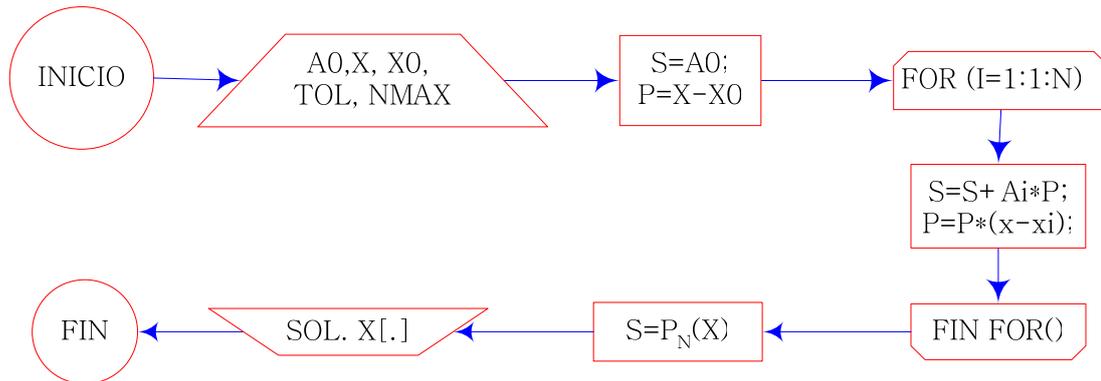
x	$f(x)$	PRIMERAS DIFERENCIAS DIVIDIDAS	SEGUNDAS DIFERENCIAS DIVIDIDAS	TERCERAS DIFERENCIAS DIVIDIDAS	...	N-ESIMAS DIFERENCIAS DIVIDIDAS	N+1-ESIMAS Y POSTERIORES DIF. DIVIDIDAS
x_0	$f[x_0]$						
x_1	$f[x_1]$	$f[x_0, x_1]$					
x_2	$f[x_2]$	$f[x_1, x_2]$	$f[x_0, x_1, x_2]$				
x_3	$f[x_3]$	$f[x_2, x_3]$	$f[x_1, x_2, x_3]$	$f[x_0, x_1, x_2, x_3]$			
...		
x_N	$f[x_N]$	1	
x_{N+1}	$f[x_{N+1}]$	1	0

Nota 24: Obligatoriamente deben reordenarse los nodos según la cercanía al punto en el cual se desea calcular el valor en $f(X)$, ya que los nodos más cercanos a X contribuyen más en la elaboración final de $f[X]$, aportando mucha más información que los nodos más lejanos a X .

Nota 25: El término correspondiente al error para el polinomio integrador de Newton es igual que el expuesto en el polinomio integrador de Lagrange.

El criterio de paro en el algoritmo de Lagrange es: $P_{i+1}(\tilde{x}) - P_i(\tilde{x}) < TOL$. Si no se cumple este criterio, son necesarios k puntos más para tener $P_{i+k}(\tilde{x})$ puntos.

A continuación se presenta la forma computacionalmente más correcta de este algoritmo.



Las diferencias divididas que sirven al cálculo del polinomio se calculan por filas, controlando el error a cada iteración, usando la recursividad para evitar redundancias.

Nota 26: Sin control del error, calcularemos el polinomio interpolante con todos los nodos.

2.- Método de las Diferencias finitas:

Este método es similar al anterior. Calcula el polinomio a partir de las *diferencias finitas* de una función en un conjunto soporte regular, o sea, con nodos equiespaciados: $x_{i+1} - x_i = h$.

Def. 51: Notación de diferencia finita k-ésima de la función f con respecto a x_i :

$$\Delta^k y_i = \Delta(\Delta^{k-1} y_i), \quad \text{donde } \Delta y_i = y_{i+1} - y_i$$

Redefiniendo las diferencias divididas, el polinomio de interpolación anterior se escribe:

$$f[x_0, x_1, \dots, x_k] = \frac{\Delta^k f(x_0)}{h^k k!} \quad \text{para } t = \frac{x - x_0}{h}$$

$$P_n(x) = f(x_0) + \frac{\Delta f(x_0)}{1!} t + \frac{\Delta^2 f(x_0)}{2!} t(t-1) + \dots + \frac{\Delta^n f(x_0)}{n!} t(t-1)\dots(t-(n-1))$$

Def. 52: Fórmula de diferencia finita interpolante de Newton:

$$P_n(X) = f(x_0) + \binom{t}{1} \Delta f(x_0) + \binom{t}{2} \Delta^2 f(x_0) + \dots + \binom{t}{n} \Delta^n f(x_0) \equiv \sum_{i=0}^n \binom{t}{i} \Delta^i f(x_0)$$

con $\binom{t}{k} = \frac{t(t-1)\dots(t-(k-1))}{k!}$

Def. 53: **Fenómeno de Runge:** circunstancia que se da si el conjunto soporte es extenso, ya que el incremento del número de operaciones acumula errores y se pierde precisión en los extremos. Deben añadirse nodos poco a poco, 2, 4, 8, 16, etcétera, e ir comprobando la bondad del ajuste del polinomio, empezando por un conjunto inicial mínimo. Cuando al añadir más nodos no varíe considerablemente la solución, habremos encontrado la exactitud máxima.

Teorema 25: Dados $n+1$ nodos de una función f desconocida, y sean $L_k(x)$ sus polinomios de Lagrange para el soporte dado y $k=0, 1, \dots, n$, se cumple que el polinomio

$$P_{2n+1}(x) = \sum_{k=0}^n [a_k + b_k(x - x_k)] L_k^2(x) \quad / \quad a_k = f(x_k), \quad b_k = f'(x_k) - 2f(x_k)L_k'(x_k)$$

verifica que:

$$P_{2n+1}(x_k) = f(x_k), \quad y \quad P'_{2n+1}(x_k) = f'(x_k)$$

D.- Nodos de Chébyshv:

Chébyshv estudió el óptimo espaciado de nodos en el cálculo de raíces. Con su método, se eligen los nodos óptimos para minimizar el error en la interpolación. Se reduce la complejidad del cálculo al minimizar el grado del polinomio interpolante, y con mínima pérdida de precisión.

Para $f(x) \in (0,1)$, construimos $P_n(x)$ con los nodos $-1 \leq x_0 \leq \dots \leq x_n \leq 1$, que satisface:

$$f(x) = P_n(x) + E_n(x),$$

donde la cota de error es: $E_n(x) = Q(x) \frac{f^{(n+1)}(\xi)}{(n+1)!}$,

y el factor $Q(x)$ es: $Q(x) = (x - x_0) \dots (x - x_n)$

1.- Polinomios de Chébyshv y propiedades:

Para minimizar el término $Q(x)$, se usan los llamados *polinomios de Chébyshv*, que cumplen una serie de propiedades. Los 8 primeros de esos polinomios son:

$$T_0(x) = 1$$

$$T_1(x) = x$$

$$T_2(x) = 2x^2 - 1$$

$$T_3(x) = 4x^3 - 3x$$

$$T_4(x) = 8x^4 - 8x^2 + 1$$

$$T_5(x) = 16x^5 - 20x^3 + 5x$$

$$T_6(x) = 32x^6 - 48x^4 + 18x^2 - 1$$

$$T_7(x) = 64x^7 - 112x^5 + 56x^3 - 7x$$

PROPIEDAD 1. Relación de recurrencia:

$$T_k(x) = 2xT_{k-1}(x) - T_{k-2}(x) \quad / \quad k = 2, 3, \dots$$

PROPIEDAD 2. Coeficiente líder:

$$\text{En } T_N(x), \text{ el coeficiente de } x^N \text{ es } 2^{N-1} \quad / \quad N \geq 1$$

PROPIEDAD 3. Simetría:

$$\text{Si } N \text{ par} \rightarrow T_N(-x) = T_N(x)$$

$$\text{Si } N \text{ impar} \rightarrow T_N(-x) = -T_N(x)$$

PROPIEDAD 4. Representación trigonométrica en [-1,1]:

$$T_N(x) = \cos(N \cdot \arccos(x)) \quad / -1 \leq x \leq 1$$

PROPIEDAD 5. Ceros simples en [-1,1]:

$$T_N(x) \text{ tiene } N \text{ ceros } x_k \text{ en } [-1,1], \text{ con: } x_k = \cos\left(\frac{(2k+1)\pi}{2N}\right) \quad / k = 0, 1, \dots, N-1$$

PROPIEDAD 6. Valores extremos:

$$|T_N(x)| \leq 1 \quad / -1 \leq x \leq 1$$

Chébyshev descubrió que, para minimizar el factor $\max\{|Q(x)|\}$, los nodos x_0, \dots, x_N deben ser escogidos de manera que se satisfaga lo siguiente:

$$Q(x) = \frac{T_{N+1}(x)}{2^N}$$

2.- Nodos de Chébyshev en otros intervalos:

Si debemos obtener una aproximación $P_n(x)$ a $f(x)$ en $[a,b]$ distinto de $[-1,1]$, hacemos un cambio de variable para formular el problema en $[-1,1]$, así:

$$x = \frac{b-a}{2}t + \frac{b+a}{2} \quad \text{o bien}$$

$$t = 2\frac{x-a}{b-a} - 1, \quad \text{con } a \leq x \leq b \text{ y } -1 \leq t \leq 1$$

Así, los nodos de Chébyshev correspondientes a $T_{n+1}(t)$ en $[-1,1]$ son:

$$t_k = \cos\left(\frac{2k+1}{2N+2}\pi\right) \quad / k = 0, 1, \dots, N$$

De manera que los nodos de interpolación de Chébyshev en $[a,b]$ son:

$$x_k = \frac{b-a}{2}t_k + \frac{b+a}{2} \quad / k = 0, 1, \dots, N$$

E.- Método de Extrapolación de Richardson:

Richardson ideó un método de extrapolación muy útil también en problemas ajenos a la extrapolación. Construye aproximaciones mejoradas a partir de términos previamente calculados, reduciendo el término de error, expresado en forma de potencias pares de $O(b^{2k})$.

(Se verá ampliamente en el tema siguiente)

Teorema 26: Método de aproximación de Richardson: Supongamos una aproximación de orden $O(b^{2k})$ a $f'(x_0)$ como $D_{k-1}(h)$, que verifique:

$$f'(x_0) = D_{k-1}(h) + c_1 h^{2k} + c_2 h^{2k+2} + \dots$$

con lo cual:

$$f'(x_0) = D_{k-1}(2h) + 4^k c_1 h^{2k} + 4^{k+1} c_2 h^{2k+2} + \dots$$

y entonces podemos construir la siguiente aproximación mejorada:

$$f'(x_0) = D_k(b) + O(b^{2k+2}) = \frac{4^k D_{k-1}(b) - D_{k-1}(2b)}{4^k - 1} + O(b^{2k+2})$$

Con las aproximaciones de Richardson podemos construir una tabla (por filas) que nos aporte aproximaciones y un criterio de paro eficaz. Es decir, cuando una aproximación tenga un error aceptable, se paran las iteraciones y se acepta ese valor como bueno.

Para el valor $h = \frac{b-a}{n}$, tendremos estas fórmulas según el error E :

	$E \in O(b^2)$	$\tilde{E} \in O(b^4)$	$\tilde{\tilde{E}} \in O(b^6)$	$\tilde{\tilde{\tilde{E}}} \in O(b^8)$...
h	$T(h)$...
h/2	$T\left(\frac{h}{2}\right)$	$\tilde{T}(h) = \frac{4T\left(\frac{h}{2}\right) - T(h)}{3}$...
h/4	$T\left(\frac{h}{4}\right)$	$\tilde{T}\left(\frac{h}{2}\right) = \frac{4T\left(\frac{h}{4}\right) - T\left(\frac{h}{2}\right)}{3}$	$\tilde{\tilde{T}}(h) = \frac{16\tilde{T}\left(\frac{h}{2}\right) - \tilde{T}(h)}{15}$...
h/8	$T\left(\frac{h}{8}\right)$	$\tilde{T}\left(\frac{h}{8}\right) = \frac{4T\left(\frac{h}{8}\right) - T\left(\frac{h}{4}\right)}{3}$	$\tilde{\tilde{T}}\left(\frac{h}{2}\right) = \frac{16\tilde{T}\left(\frac{h}{4}\right) - \tilde{T}\left(\frac{h}{2}\right)}{15}$	$\tilde{\tilde{\tilde{T}}}(h) = \frac{64\tilde{\tilde{T}}\left(\frac{h}{2}\right) - \tilde{\tilde{T}}(h)}{63}$...
...

El criterio de paro es:

$$|T^{*(k)}(w) - T^{*(k-1)}(w)| < \text{TOL} \rightarrow \text{conseguida exactitud deseada.}$$

Si no se logra satisfacer este criterio, se construye otra fila, iterando de nuevo con los términos anteriores.

Tema 6: Ajuste de curvas

Las aplicaciones de las técnicas numéricas a la vida real consisten, a menudo, en ajustar una curva a una serie de datos experimentales, que generalmente son de dudosa precisión.

A.- Mínimos cuadrados:

Normalmente, se dispone de una clase de fórmulas previamente establecidas con una serie de parámetros llamados coeficientes que hay que ajustar según los datos del modelo. El método de los mínimos cuadrados *minimiza* los *errores cuadráticos* de los cálculos.

1.- Tipos de errores:

Def. 54: **Desviación o residuo:** Es el error en este tipo de aproximaciones. Puede ser elevado, y se define como: $e_k = f(x_k) - y_k$, $1 \leq k \leq n$

Podemos medir la distancia entre la curva $y=f(x)$ y los datos, usando varias normas:

Def. 55: **Error máximo:** determina si el error en un punto es de gran dimensión.

Def. 56: **Error medio:** es la media de los valores absolutos de los errores en n puntos.

Def. 57: **Error cuadrático medio:** considera la naturaleza aleatoria del error.

$$\text{Error máximo: } E_{\infty}(f) = \max \{ |f(x_k) - y_k| : 1 \leq k \leq N \},$$

$$\text{Error medio: } E_1(f) = \frac{1}{N} \sum_{k=1}^N |f(x_k) - y_k|,$$

$$\text{Error cuadrático: } E_2(f) = \sqrt{\frac{1}{N} \sum_{k=1}^N |f(x_k) - y_k|^2}$$

2.- Rectas de regresión en mínimos cuadrados:

La recta de regresión pasa *cerca*, y no *sobre*, de cada uno de los n puntos dados como aproximación lineal de la forma:

$$y = f(x) = Ax + B$$

Def. 58: **Recta óptima o de regresión en el sentido de los mínimos cuadrados:** la que minimiza el error $E_2(f)$. O sea si minimiza el valor $N(E_2(f))^2$, valor que equivale a la suma de los cuadrados de las distancias verticales desde los puntos hasta la recta.

$$N(E_2(f))^2 = \sum_{k=1}^N (Ax_k + B - y_k)^2$$

Teorema 27: Para el conjunto de los N nodos conocidos (x_k, y_k) , los coeficientes A y B de una recta de regresión óptima en mínimos cuadrados $y = Ax + B$, son la solución del siguiente sistema lineal, conocido como **ecuaciones normales de Gauss**:

$$\begin{cases} \sum_{k=1}^N x_k y_k = A \left(\sum_{k=1}^N x_k^2 \right) + B \left(\sum_{k=1}^N x_k \right) \\ \sum_{k=1}^N y_k = A \left(\sum_{k=1}^N x_k \right) + BN \end{cases}$$

3.- Ajuste exponencial:

A veces, el ajuste de una situación se modela con una función potencial tipo $f(x) = Ax^M$.

Teorema 28: Ajuste exponencial: para el conjunto de los N nodos conocidos (x_k, y_k) , el coeficiente único A de la curva exponencial óptima en mínimos cuadrados $y^{\text{oo}} = Ax^M$, viene dado por la expresión:

$$A = \frac{\sum_{k=1}^N x_k^M y_k}{\sum_{k=1}^N x_k^{2M}}$$

4.- Ajuste de curvas por linealización de los datos:

Si vamos a ajustar una curva exponencial de la forma $y = Ce^{Ax}$, debemos tomar logaritmos, y también hacer un cambio de variable y otro de constante, para lograr una relación lineal.

$$\ln(y) = Ax + \ln(C)$$

de donde

$$\begin{cases} Y = \ln(y) \\ X = x \\ B = \ln(C) \end{cases} \Rightarrow Y = AX + B$$

Una vez planteadas y resueltas las ecuaciones normales de Gauss, se hallará el parámetro C de la relación:

$$C = e^B$$

y luego se procede a calcular la recta de regresión para los puntos $\{(X_k, Y_k)\}$.

B.- Interpolación polinomial a trozos:

1.- Cerchas o ‘splines’:

Una solución mejor que la interpolación polinomial es enlazar las gráficas de polinomios de grado bajo $S_k(x)$ interpolados entre 2 extremos consecutivos $(k, k+1)$. Se enlazarán en cada par $(x_{k+1/2}, y_{k+1/2})$ las curvas $y=S_k(x)$ e $y=S_{k+1}(x)$, construidas sobre los intervalos $[x_k, x_{k+1}]$ y $[x_{k+1}, x_{k+2}]$.

Nota 27: El ajuste de curvas por trozos se suele llamar *ajuste de trazador*.

Def. 59: **Cercha o curva polinomial a trozos** (en ingles, *spline*):

$$S(x) \equiv \{S_k(x)\}, \quad \text{con } S_k(x) = y_k + d_k(x-x_k). \quad \text{con } x_0 < x_1 < \dots < x_{N-1} < x_N, \quad \text{con } x \in [x_k, x_{k+1}]$$

Esta técnica debe extenderse a polinomios de grado mayor, porque la cercha lineal, generalmente discontinua en los nodos pares, y la cuadrática no tienen derivada segunda. Aún así, no es aconsejable usar polinomios de grado superior a 3, dada la complejidad de los cálculos.

Def. 60: **Cercha cuadrática:** con un número impar $2M-1$ de nodos del conjunto soporte $\{x_0, x_1, \dots, x_{2M}\}$, calcula un polinomio cuadrático en cada subintervalo $[x_{2k}, x_{2k+2}]$ para $k=0, \dots, M-1$.

2.- Cerchas cúbicas:

Si usamos polinomios cúbicos a trozos, garantizamos la derivabilidad, hasta la segunda derivada, en el intervalo de estudio.

Def. 61: **Cercha cúbica interpoladora:** es la definida por:

$$S(x) \equiv \{S_k(x)\}_N, \quad \text{con } x \in [x_k, x_{k+1}], \quad \text{y } a=x_0 < x_1 < \dots < x_{N-1} < x_N=b.$$

Los N polinomios $S_k(x)$ se escriben como coeficientes $S_{k,0}$, $S_{k,1}$, $S_{k,2}$ y $S_{k,3}$, y las relaciones:

$$1) \quad S(x) = S_k(x) = S_{k,0} + S_{k,1}(x-x_k) + S_{k,2}(x-x_k)^2 + S_{k,3}(x-x_k)^3 \quad / \quad x \in [x_k, x_{k+1}], \quad k=0 \dots N-1$$

Indica que $S(x)$ es un polinomio cúbico a trozos

$$2) \quad S(x_k) = y_k \quad / \quad k=0 \dots N$$

Significa que $S(x)$ interpola los datos

$$3) \quad S_k(x_k) = S_{k+1}(x_{k+1}) \quad / \quad k=0 \dots N-2$$

$$4) \quad S'_k(x_k) = S'_{k+1}(x_{k+1}) \quad / \quad k=0 \dots N-2$$

3) y 4) Significan que $S(x)$ es una función derivable y con derivada continua.

$$5) \quad S''_k(x_k) = S''_{k+1}(x_{k+1}) \quad / \quad k=0 \dots N-2$$

Indica que la derivada segunda de $S(x)$ también existe y es continua.

a.- Construcción de una cercha cúbica interpoladora:

Cada S_k tiene 4 coeficientes ($S_{k,0}, S_{k,1}, S_{k,2}, S_{k,3}$) desconocidos, y debemos tener un total de $4N$ condiciones que los identifiquen. La relación 2) proporciona $N+1$ condiciones, y 3), 4) y 5) otras $N-1$ cada una, que suman en conjunto $4N-2$ condiciones.

Los 2 grados de libertad restantes se llaman restricciones de extremos, porque involucran los valores de $S'(x)$ y $S''(x)$.

Como $S(x)$ es un polinomio cúbico a trozos, $S''(x)$ es lineal a trozos en $[x_0, x_N]$. por tanto, la fórmula de interpolación lineal de Lagrange lo representa así:

$$S''(x) = S''_k(x) = S''(x_k) \frac{x - x_{k+1}}{x_k - x_{k+1}} + S''(x_{k+1}) \frac{x - x_k}{x_{k+1} - x_k}, \quad x_k \leq x \leq x_{k+1}$$

Usando la notación: $m_i = S''(x_i)$ y $h_k = x_{k+1} - x_k$ tenemos que:

$$S''(x) = \frac{m_k}{h_k}(x_{k+1} - x) + \frac{m_{k+1}}{h_k}(x - x_k), \quad x_k \leq x \leq x_{k+1}, \quad k = 0, \dots, N-1$$

y operando repetidas veces, obtenemos lo siguiente para $k=1 \dots N-1$:

$$h_{k-1}m_{k-1} + 2(h_{k-1} + h_k)m_k + h_k m_{k+1} = u_k \quad \text{donde } u_k = 6(d_k - d_{k-1}) \quad \text{con } d_k = \frac{y_{k+1} - y_k}{h_k}$$

b.- Tipos de cerchas cúbicas interpoladoras:

La siguiente tabla muestra las restricciones en los extremos para cerchas cúbicas:

ESTRATEGIA	ECUACIONES PARA m_0 Y m_N
Cercha cúbica sujeta: <i>cercha sujeta</i> Especifica $S'(x_0)$ y $S'(x_N)$ (opción óptima si se conocen derivadas)	(1) $m_0 = \frac{3}{h_0}(d_0 - S'(x_0)) - \frac{m_1}{2}$ $m_N = \frac{3}{h_{N-1}}(S'(x_N) - d_{N-1}) - \frac{m_{N-1}}{2}$
Cercha cúbica natural: (curva relajada) <i>cercha natural</i>	(2) $m_0 = 0 \quad \text{y} \quad m_N = 0$
Extrapolar $S''(x)$ a los extremos <i>cercha extrapolada</i>	(3) $m_0 = m_1 - \frac{h_0(m_2 - m_1)}{h_1}$ $m_N = m_{N-1} - \frac{h_{N-1}(m_{N-1} - m_{N-2})}{h_{N-2}}$
$S''(x)$ constante cerca de los extremos <i>cercha parabólica</i>	(4) $m_0 = m_1 \quad \text{y} \quad m_N = m_{N-1}$
Especifica $S''(x)$ en cada extremo <i>cercha de curvatura</i>	(5) $m_0 = S''(x_0) \quad \text{y} \quad m_N = S''(x_N)$

donde los coeficientes a_j y b_j del desarrollo en serie de Fourier salen de estas fórmulas de Euler:

$$\begin{cases} a_j = \frac{1}{\pi} \int_{-\pi}^{\pi} f(x) \cos(jx) dx & / j = 0, 1, \dots \\ b_j = \frac{1}{\pi} \int_{-\pi}^{\pi} f(x) \operatorname{sen}(jx) dx & / j = 1, 2, \dots \end{cases}$$

Obtenemos a_0 por sustitución en la fórmula de Euler para a_j , j por su valor en $j=0$.

Teorema 29: Desarrollo de una serie de Fourier: si la función $f(x)$ anteriormente definida tiene derivada $f'(x)$ continua a trozos en $[-\pi, \pi]$, y tiene derivadas laterales por la izquierda y por la derecha en cada punto de dicho intervalo, entonces, la serie de Fourier de desarrollo de la función $f(x)$ en el intervalo $[-\pi, \pi]$, $S(x)$, converge para $\forall x \in [-\pi, \pi]$ y se verifica además que:

$$S(x) = f(x) \quad / \quad \forall x \in [-\pi, \pi] \text{ donde } f \text{ es continua}$$

Además, si $x=a$ es un punto de discontinuidad de f , entonces se verifica también que:

$$S(a) = \frac{f(a^-) + f(a^+)}{2}$$

Teorema 30: Series de Fourier de desarrollo de cosenos: si la función $f(x)$ es par, es decir, si $f(-x) = f(x) \quad / \quad \forall x$, y tiene un periodo 2π , y además, tanto dicha función f como su función derivada f' son ambas continuas a trozos en x , entonces la serie de Fourier de la función $f(x)$, $S(x)$, sólo tiene los términos de los cosenos:

$$S(x) = \frac{a_0}{2} + \sum_{j=1}^{\infty} (a_j \cos(jx)),$$

donde

$$a_j = \frac{1}{\pi} \int_{-\pi}^{\pi} f(x) \cos(jx) dx \quad / \quad j = 0, 1, \dots$$

Teorema 31: Series de Fourier de desarrollo de senos: si f es impar, o lo que es lo mismo, si $f(-x) = -f(x) \quad / \quad \forall x$, y tiene un periodo 2π , y tanto f como f' son continuas a trozos en x , entonces la serie de Fourier de f , $S(x)$, sólo tiene los términos de los senos:

$$S(x) = \frac{a_0}{2} + \sum_{j=1}^{\infty} (b_j \operatorname{sen}(jx)),$$

donde

$$b_j = \frac{1}{\pi} \int_{-\pi}^{\pi} f(x) \operatorname{sen}(jx) dx \quad / \quad j = 1, 2, \dots$$

2.- Aproximación mediante polinomios trigonométricos:

Usando lo visto en el desarrollo de las series de Fourier, construimos un método para aproximar una función por polinomios trigonométricos.

Def. 64: **Polinomio trigonométrico de grado M:** es una serie finita de la forma:

$$T_M(x) = \frac{a_0}{2} + \sum_{j=1}^{\infty} (a_j \cos(jx) + b_j \operatorname{sen}(jx))$$

Teorema 32: Serie de Fourier discreta: si para $\{(x_j, y_j)\} / j=0, \dots, N$, existen $N+1$ puntos que cumplen que:

$$x_j = -\pi + \frac{2j\pi}{N}, \quad / j=0, 1, \dots, N$$

y dichos puntos están equiespaciados, y tomamos sus ordenadas como $y_j = f(x_j)$ para $f(x)$ periódica de periodo 2π , y si $2M < N$, entonces existe un polinomio trigonométrico $T_M(x)$ que minimiza la cantidad

$$\sum (f(x_k) - T_M(x_k))^2$$

Los coeficientes de este polinomio se determinan en estas fórmulas:

$$\begin{cases} a_j = \frac{2}{N} \sum_{k=1}^N f(x_k) \cos(jx_k) & / j=0, 1, \dots, M \\ b_j = \frac{2}{N} \sum_{k=1}^N f(x_k) \operatorname{sen}(jx_k) & / j=1, 2, \dots, M \end{cases}$$

Tema 7: Diferenciación e integración numérica

A.- Derivación numérica:

Para calcular una aproximación numérica de la derivada $f'(x)$ de una función $f(x)$,

$$f'(x) = \lim_{h \rightarrow 0} \frac{f(x+h) - f(x)}{h}$$

podemos elegir una sucesión $\{h_k\} \rightarrow 0$ y calcular los términos de la sucesión $\{D_k\}$ dados por:

$$D_k = \frac{f(x+h_k) - f(x)}{h_k} \quad / k = 1, 2, \dots$$

Nota 29: El problema para calcular esta aproximación estriba en elegir un buen h_k para que D_k converja a $f'(x)$

1.- Método de las diferencias centradas:

Si $f(x)$ puede evaluarse en un intervalo $[a, b]$ que contenga a x , entonces la mejor fórmula que involucra 2 puntos es la que usa abscisas situadas simétricamente a izquierda y derecha de x .

Teorema 33: Formula centrada de orden h^2 : si para una función $f(x) \in C^3$ en $[a, b]$, y $\{x-h, x+h\} \in [a, b]$, entonces:

$$f'(x) = \frac{f(x+h) - f(x-h)}{2h} + E_{trunc}(f, h),$$

$$\text{con } E_{trunc}(f, h) = \frac{-h^2 f^{(3)}(c)}{6} \in \mathcal{O}(h^2)$$

donde $c = c(x) \in [a, b]$ en el término del error de truncamiento.

El error total de la máquina se compone de este error de truncamiento más un inevitable error de redondeo.

Teorema 34: Formula centrada de orden h^4 : supongamos ahora que una función $f(x) \in C^5$ $[a, b]$, y además que los nodos $\{x-2h, x-h, x, x+h, x+2h\} \in [a, b]$, entonces ahora nos queda:

$$f'(x) = \frac{-f(x+2h) + 8f(x+h) - 8f(x-h) + f(x-2h)}{12h} + E_{trunc}(f, h),$$

$$\text{con } E_{trunc}(f, h) = \frac{-h^4 f^{(5)}(c)}{30} \in \mathcal{O}(h^4)$$

2.- Método de extrapolación de Richardson:

No es un método de derivación ni de integración, sino un método para la obtención eficiente de fórmulas de mayor orden (con un error menor) únicamente a partir de datos anteriores.

Se basa en la sustitución en f' del término h por $\frac{h}{2^n}$, con $n=0, \dots, k$, de modo que el error aumente de orden, y por tanto, disminuya de tamaño.

$$f'(x) = D_k(h) + E(f, h) \quad \text{con} \quad \begin{cases} D_k(h) = \frac{4^k D_{k-1}(h) - D_{k-1}(2h)}{4^k - 1} \\ E_{trunc}(f, h) \in \mathcal{O}(h^{2k+2}) \end{cases}$$

Veamos el proceso de construcción de la matriz del método de Richardson:

	<i>Error</i> $\in \mathcal{O}(h^2)$		
$h = \frac{b-a}{2}$	$D(h) = \frac{f(x+h) - f(x-h)}{2h}$	<i>Error</i> $\in \mathcal{O}(h^4)$	
$h = \frac{h}{2}$	$D(h/2) = \dots$	$\tilde{D}(h) = \frac{4D\left(\frac{h}{2}\right) - D(h)}{3}$	<i>Error</i> $\in \mathcal{O}(h^6)$
$h = \frac{h}{4}$	$D(h/4) = \dots$	$\tilde{D}\left(\frac{h}{2}\right) = \frac{4D\left(\frac{h}{4}\right) - D\left(\frac{h}{2}\right)}{3}$	$\tilde{\tilde{D}}(h) = \frac{16\tilde{D}\left(\frac{h}{2}\right) - \tilde{D}(h)}{15}$ <i>Error</i> $\in \mathcal{O}(h^8)$
$h = \frac{h}{8}$	$D(h/8) = \dots$	$\tilde{D}\left(\frac{h}{4}\right) = \frac{4D\left(\frac{h}{8}\right) - D\left(\frac{h}{4}\right)}{3}$	$\tilde{\tilde{D}}\left(\frac{h}{2}\right) = \frac{16\tilde{D}\left(\frac{h}{4}\right) - \tilde{D}\left(\frac{h}{2}\right)}{15}$ $\tilde{\tilde{\tilde{D}}}(h) = \frac{64\tilde{\tilde{D}}\left(\frac{h}{2}\right) - \tilde{\tilde{D}}(h)}{63}$

1.- Dividimos el intervalo de derivación $[a,b]$ en n partes suficientemente pequeñas, de manera que tengamos: $x_i = a + h \cdot i$, con $h = \frac{b-a}{n}$

2.- Entonces podemos construir esta matriz iterativamente, para ir minimizando el término del error sucesivamente:

2.1.- En cada iteración, el n° de partes se multiplica por 2 ($2n$), a la vez que cada subintervalo se divide por 2 ($h/2$), con lo que al final tendremos $2^k n$ partes.

2.2.- La matriz se construye por filas, según sea necesaria más precisión en el cálculo.

3.- El criterio de paro es: $|T^{(i)}(p) - T^{(i-1)}(p)| < \text{TOL} \rightarrow \text{FIN}$

La fórmula de la derivada segunda, a la que también se le puede aplicar Richardson, es:

$$f''(x) \approx \frac{f(x+h) - 2f(x) + f(x-h)}{h^2} + E_{trunc}(f, h) \quad / \quad E_{trunc}(f, h) \in \mathcal{O}(h^2)$$

B.- Integración numérica:

El cálculo de la integral de una función continua f en un intervalo $[a,b]$ puede aproximarse por varios métodos. Algunos se basan en las *fórmulas de Newton-Cotes*.

Nota 30: En Matlab, se puede integrar una función f en (a,b) con la orden `» quad8('f',a,b);`

1.- Fórmulas de Newton-Cotes:

Basándonos en el método del *polinomio de interpolación de Lagrange*, y el conjunto soporte $X = \{x_0, \dots, x_n\}$, donde $\{x_0=a, x_1=a+h, \dots, x_i=a+i \cdot h, \dots, x_n = a+n \cdot h = b\}$, procedemos así:

Llamamos $z(x) = (x-x_0)(x-x_1)\dots(x-x_n)$, y entonces los polinomios interpolantes de Lagrange quedan:

$$L_i(x) = \frac{\left(\frac{x-a}{b}\right)\left(\frac{x-a}{b}-1\right)\dots\left(\frac{x-a}{b}-n\right)}{\left(\frac{x-a}{b}-i\right) \cdot i! \cdot (n-i)! \cdot (-1)^{n-i}},$$

y haciendo $t = \frac{x-a}{b}$, tendríamos:

$$L_i(x) = \frac{t(t-1)\dots(t-n)}{(t-i) \cdot i! \cdot (n-i)! \cdot (-1)^{n-i}}$$

Por tanto, definimos los coeficientes de Newton-Cotes así:

Def. 65: Coeficiente de Newton-Cotes a_i :

$$a_i = \int_0^n L_i(x) dx = \dots \cdot b \cdot (-1)^{n-i} \frac{\binom{n}{i}}{n!} \int_0^n \frac{z(t)}{t-i} dt$$

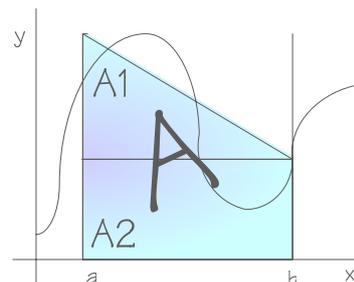
Nota 31: Los coeficientes de Newton-Cotes verifican que $a_k = a_{n-k}$.

2.- Método del Trapecio:

El área formada por la proyección de una función en $[a,b]$ se considera la integral de dicha función. Si consideramos un trapecio con los puntos máximo y mínimo en $[a,b]$, obtendremos el área total de la función en el intervalo como el área del trapecio que se formaría al aproximar la sección a la de esta figura:

$$A1 = \frac{b-a}{2} |f(b) - f(a)| \quad \text{y} \quad A2 = (b-a) \max(f(b), f(a))$$

$$A = A1 + A2 = \frac{b-a}{2} |f(b) - f(a)| + (b-a) \max(f(b), f(a))$$



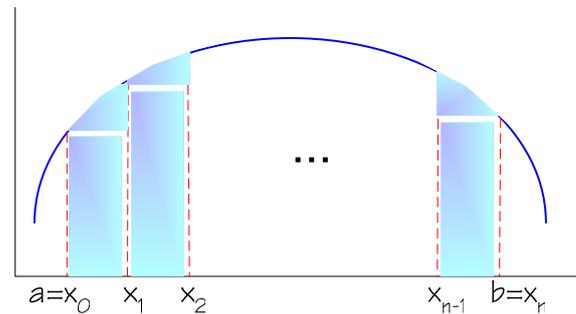
Está basada en la fórmula de Newton-Cotes en el caso $n=1$, donde $a_0 = a_1 = \frac{b-a}{2}$

El error cometido viene dado por $\varepsilon = \frac{-h^3 f''\left(\frac{a+b}{2}\right)}{12}$, donde el error $\varepsilon \in \mathcal{O}(h^3)$

3.- Método del Trapecio compuesto:

Valiéndose de la formulación anterior, delinea un malleado suficientemente fino en $[a,b]$, logrando n intervalos infinitesimales; es decir, calcula el área de la función integralmente, lo que equivale a una aproximación a la integral de la función.

$$h = \frac{b-a}{n} \quad \text{con:} \quad x_i = a + n * h$$



$$A \approx \int_a^b f(x) dx = \frac{b-a}{2} \left[|f(x_0) + f(x_1)| + |f(x_1) + f(x_2)| + \dots + |f(x_{n-1}) + f(x_n)| \right]$$

$$A = \frac{b-a}{2} \left[|f(a) + f(b)| + 2 \sum_{i=1}^{n-1} f(x_i) \right] + E_{tr}(f, b) \longrightarrow E_{tr} \in \mathcal{O} \left(\frac{-b^2(b-a)}{12} f''(\rho) \right) / \rho \in (a, b)$$

Nota 32: A esta fórmula se le puede aplicar el método de extrapolación de Richardson para refinarla en el término del error.

4.- Regla del Trapecio recursiva:

Def. 66: **Sucesión de aproximaciones de la regla del trapecio:** a partir de

$$T(0) = (f(a) + f(b)) * h / 2 \quad (\text{regla del trapecio con incremento } h = b - a),$$

y para cada $J \geq 1$, se define

$$T(J) = T(f, h) \quad (\text{regla del trapecio con incremento } h = \frac{b-a}{2^J}),$$

Teorema 35: Regla recursiva del Trapecio: teniendo en cuenta la definición anterior, la sucesión $\{T(J)\}$ de aproximaciones dadas por la regla del trapecio viene generada por la fórmula recursiva siguiente para $J = 1, 2, \dots$ y $\{x_k = a + kh\}$:

$$T(J) = \frac{T(J-1)}{2} + b \sum_{k=1}^M f(x_{2k-1})$$

5.- Método del Polinomio:

Calcula el polinomio de interpolación que se aproxime a $f(x)$ y evalúa la integral de $P_n(x)$ como similar a la de $f(x)$.

$$P_n(x) \equiv [a_n, \dots, a_1, a_0] \longrightarrow \int_a^b x^n dx = \left. \frac{x^{n+1}}{n+1} \right|_a^b = \frac{b^{n+1} - a^{n+1}}{n+1}$$

$$\int_a^b P_n(x) dx = \sum_{b=0}^n a_b \int_a^b x^b dx = \frac{b^{b+1} - a^{b+1}}{b+1}$$

6.- Método de integración de Romberg:

Consiste en usar una aproximación con incrementos h y $2h$, manipulando algebraicamente ambas respuestas para obtener una solución, de manera que cada nivel de mejora incrementa el orden del término del error de $\mathcal{O}(h^{2N})$ a $\mathcal{O}(h^{2N+2})$.

También puede aplicarse el método de extrapolación de Richardson visto en páginas anteriores.

Def. 67: Sucesión de aproximaciones de la Regla del Trapecio recursivo: a partir de $R(J,0) = T(0)$ (regla recursiva del trapecio) y para cada $J = 0, 1, \dots$ y $J \geq k$, definimos:

$$R(J, k) = \frac{4^k R(J, k-1) - R(J-1, k-1)}{4^k - 1}$$

Este método se construye como el de extrapolación de Richardson, por filas:

J	R(J,0)	R(J,1)	R(J,2)	R(J,3)	R(J,4)	...
0	R(0,0)					
1	R(1,0)	R(1,1)				
2	R(2,0)	R(2,1)	R(2,2)			
3	R(3,0)	R(3,1)	R(3,2)	R(3,3)		
4	R(4,0)	R(4,1)	R(4,2)	R(4,3)	R(4,4)	
...

C.- Integración o cuadratura adaptativa:

La integración o cuadratura adaptativa trata de manejar problemas donde es necesario un método que vaya ajustando el incremento de la integración de manera que sea menor en aquellas porciones de curva en la que aparezcan oscilaciones más pronunciadas, por ejemplo, el método de Simpson.

Según el autor, la integral definida de una función puede entenderse o bien como una suma de incrementos infinitesimales, conformando un área en su conjunto, o bien como la función inversa a la derivada definida.

7.- Método de Simpson:

Se basa en la *regla de Simpson* y en las *fórmulas de Newton-Côtes*, por tanto es muy similar a la del *Trapezio compuesto*.

Usa los *coeficientes de Newton-Côtes* para $n=2$, donde

$$x_0 = a, \quad x_1 = t = \frac{b+a}{2}, \quad x_2 = b.$$

Teorema 36: Regla de Simpson: Cuando sólo se logran 2 subintervalos en $[a,b]$, se tiene que:

$$S(a,b) = \frac{b-a}{6} (f(a) + 4f(t) + f(b))$$

donde $t = \frac{a+b}{2}$ es el centro de $[a,b]$ y $h = \frac{b-a}{2}$,

o lo que es lo mismo, para $a_0 = a, a_2 = b, a_1 = \frac{b-a}{2}$, se tiene que:

$$\int_a^b f(x) dx = \frac{b-a}{6} \left(f(a) + 4f\left(\frac{a+b}{2}\right) + f(b) \right)$$

El error cometido viene dado por

$$\varepsilon = \frac{-b^5 f^{(4)}\left(\frac{a+b}{2}\right)}{90} \quad / \quad \varepsilon \in O(h^5)$$

Además, si $f \in C^4[a,b]$, entonces existe un punto $d_1 \in [a,b]$ tal que:

$$\int_a^b f(x) dx = S(a,b) - \frac{b^5 f^{(4)}(d_1)}{90}$$

8.- Regla compuesta de Simpson: (refinamiento)

Con 4 subintervalos en $[a,b]$, se obtiene dividiendo este intervalo en 2 subintervalos de igual tamaño $[a_1,b_1]$ y $[a_2,b_2]$, y un incremento de $h/2$. Es decir, para $n=4$ con coeficientes de Newton-Cotes.

Se aplica la fórmula anterior de $S(a,b)$ en cada trozo, donde $a_1=a$, $b_1=a_2=c$, $b_2=b$, y c_1 y c_2 son respectivamente los puntos medios de $[a_1,b_1]$ y $[a_2,b_2]$:

$$S(a_1,b_1) + S(a_2,b_2) = \frac{h}{6}(f(a_1) + 4f(c_1) + f(b_1)) + \frac{h}{6}(f(a_2) + 4f(c_2) + f(b_2))$$

Además, si $f \in C^4[a_k,b_k]$, entonces existe un punto $d_2 \in [a_k,b_k]$ tal que:

$$\int_{a_k}^{b_k} f(x)dx = S(a_1,b_1) + S(a_2,b_2) - \frac{h^5}{16} * \frac{f^{(4)}(d_2)}{90}$$

Tema 8: Problemas de valor inicial y de contorno.

A.- Introducción a la ecuación diferencial:

Las ecuaciones diferenciales definen la evolución de un sistema en el tiempo. En ellas, la incógnita no es un número, sino una función definida en un intervalo, y además intervienen como incógnitas la propia función y una o varias sus derivadas, que dependen del tiempo y la posición anterior $f(y)$.

Buscamos el valor de la función $y=f(t)$ que dependa de ciertos parámetros:

$$y' = \frac{dy(t)}{dt} = f(y, t)$$

Nota 33: Las ecuaciones diferenciales son una herramienta importante en las matemáticas, usadas habitualmente para construir modelos matemáticos de problemas de la ciencia y la ingeniería.

Salvo en contadas ocasiones, las ecuaciones en diferencia son de difícil resolución.

Sabiendo que p , q y r son funciones continuas en un intervalo $I \subset \mathbb{R}$, definimos:

Def. 68: Ecuaciones diferenciales de primer orden: $y' + p(t)y = q(t)$

Def. 69: Ecuaciones diferenciales de segundo orden: $y'' + p(t)y' + q(t)y = r(t)$

Si $r(t)=0$ en I , la ecuación diferencial de segundo orden es homogénea. Si $p(t)$ y $q(t)$ son constantes en I , dicha ecuación es de coeficientes constantes.

Def. 70: Problema de Cauchy o del valor inicial: una ecuación diferencial $y'=f(y,t)$, donde se conoce el valor de la función en el instante inicial, esto es, su valor inicial $y(t_0)=\alpha$, y deseamos conocer la evolución de un sistema desde ese momento. Generalmente $t_0=0$.

Los problemas de valor inicial se abordan dividiendo las abscisas del intervalo $[a,b]$ en n subintervalos equipartitos, con las particiones dadas por

$$t_k = a + k \cdot h \quad / \quad k = 0, 1, \dots, n.$$

Equivale a una gráfica de malleado fino, con un paso suficientemente pequeño, por ejemplo $h=0.01$, o bien con n pasos mediante la fórmula

$$h = (b - a) / n$$

Nota 34: El valor del incremento h se llama *tamaño de paso*.

1.- Método poligonal de Euler:

El método de Euler es de 1 paso, de orden 1, y poco eficiente, ya que al incrementarse el número de nodos también lo hace el del *error de truncamiento* de la fórmula de Taylor en la que se basa y el *error de redondeo*. El resultado de unir todos los puntos solución es una línea quebrada.

Def. 71: Aproximación de Euler: Si $y'=f(y,t)$ en $[t_0, t_n]$, con $y(t_0)=y_0$, para $t_{i+1}=t_i+h$, y con $i=0,1,.. n-1$, definimos $y(t_{i+1}) \approx y_{i+1} = y_i + h \cdot f(t_i, y_i)$, entonces:

$$f(t_i, y(t_i)) \approx \frac{y(t_{i+1}) - y(t_i)}{h}$$

Algoritmo 15: ALGORITMO DE EULER

$$\left\{ \begin{array}{l} f'(y_i, t_i) = f(y, t) \\ f(y_0, t_0) = \alpha \\ f(y_{i+1}, t_{i+1}) \approx y_i + h \cdot f(y_{i+1}, t_i) \end{array} \right\} \begin{array}{l} t \in [t_0, t_n] \\ t_{i+1} = t_i + h \\ i = 0, 1, \dots, n-1 \end{array}$$

2.- Método de la serie de Taylor:

Teorema 37: Fórmula de Taylor de 1 variable:

$$f(x + \delta) = \sum_0^n f^{(i)}(x) * \frac{\delta^i}{i!} + \overbrace{f^{(n+1)}(\alpha)}^{\text{ERROR}} * \frac{h^{n+1}}{(n+1)!} \approx f(x) + \delta * f'(x)$$

Teorema 38: Teorema de Taylor para diferenciales: para $y(t) \in C^{n+1}[t_0, b]$ con desarrollo de Taylor de orden n alrededor de un punto $t=t_k \in [t_0, b]$ dado por:

$$y(t_k + h) = y(t_k) + h * T_n(t_k, y(t_k)) + O(h^{n+1})$$

$$T_n(t_k, y(t_k)) = \sum_{j=1}^n \frac{y^{(j)}(t_k)}{j!} h^{j-1}, \quad \text{con } y^{(j)} = f^{(j-1)}(t, y(t))$$

Teorema 39: Regla de la cadena para derivar una función de 2 variables:

$$f'(x, y) = \frac{\partial f}{\partial x} f'(x) + \frac{\partial f}{\partial y} f'(y) \Rightarrow y^{(j)} = P^{(j-1)} f(t, y(t)), \quad \text{con } P = \left(\frac{\partial}{\partial t} + f \frac{\partial}{\partial y} \right)$$

La fórmula del método de Taylor queda numéricamente así:

$$y_{k+1} = y_k + d_1 h + \frac{d_2 h^2}{2!} + \frac{d_3 h^3}{3!} + \dots + \frac{d_n h^n}{n!} + O(h^{n+1})$$

siendo $d_j = y^{(j)}(t_k) / j = 1, 2, \dots, n$ en cada paso $k = 0, 1, \dots, n-1$

3.- Métodos de Runge-Kutta:

Se construyen a partir del método de Taylor de orden n , con un error de orden $n + 1$, sin evaluar derivadas parciales, y evaluando a cada paso la función en varios puntos.

Usaremos $n=4$, denotado RK4, ya que es estable, preciso y fácil de programar. Calcula la aproximación y_{k+1} así:

$$y_{k+1} = y_k + w_1 k_1 + w_2 k_2 + w_3 k_3 + w_4 k_4 \quad \begin{cases} k_1 = h * f(t_k, y_k) \\ k_2 = h * f(t_k + a_1 h, y_k + b_1 k_1) \\ k_3 = h * f(t_k + a_2 h, y_k + b_2 k_1 + b_3 k_2) \\ k_4 = h * f(t_k + a_3 h, y_k + b_4 k_1 + b_5 k_2 + b_6 k_3) \end{cases}$$

Emparejando con Taylor, y tras algunas operaciones ...

$$y_{k+1} = y_k + \frac{h(k_1 + 2k_2 + 2k_3 + k_4)}{6} \quad \begin{cases} k_1 = f(t_k, y_k) \\ k_2 = f\left(t_k + \frac{h}{2}, y_k + \frac{h}{2} f_1\right) \\ k_3 = f\left(t_k + \frac{h}{2}, y_k + \frac{h}{2} f_2\right) \\ k_4 = f(t_k + h, y_k + 3f_3) \end{cases}$$

4.- Control del error con métodos de Runge-Kutta-Fehlberg (RKF):

El error de truncamiento local τ , relacionado con el error global de un método de diferencia, puede usarse ventajosamente para aproximar el tamaño de paso óptimo para controlar el error global. Se considera el error de truncamiento si reemplazamos h por q^*h , con $q \gg 0$, y acotado superiormente, donde:

$$\tau_{i+1}(qh) \approx k(qh)^n = q^n (kb^n) \approx \frac{q^n}{b} (\tilde{y}_{i+1} - y_{i+1})$$

Para acotar τ con la tolerancia admitida (ε), escogemos q tal que se cumpla:

$$\frac{q^n}{b} |\tilde{y}_{i+1} - y_{i+1}| \approx |\tau_{i+1}(qh)| \leq \varepsilon, \quad \text{que finalmente queda:} \quad q \leq \sqrt[n]{\frac{\varepsilon b}{|\tilde{y}_{i+1} - y_{i+1}|}}$$

Presentado en 1970 por Erwin Fehlberg, el método RKF consiste en proporcionarle al método de Runge-Kutta un error de truncamiento local de orden 5.

$$\tilde{y}_{k+1} = y_k + \frac{16}{135} k_1 + \frac{6656}{12,825} k_3 + \frac{28,561}{56,430} k_4 - \frac{9}{50} k_5 + \frac{2}{55} k_6,$$

para estimar el error local en el método de Runge-Kutta de orden 4:

$$y_{k+1} = y_k + \frac{25}{216} k_1 + \frac{1408}{2565} k_3 + \frac{2197}{4101} k_4 - \frac{1}{5} k_5.$$

Donde en cada iteración se calculan los siguientes 6 valores:

$$\begin{aligned}k_1 &= h * f(t_k, y_k) \\k_2 &= h * f\left(t_k + \frac{b}{4}, y_k + \frac{k_1}{4}\right) \\k_3 &= h * f\left(t_k + \frac{3}{8}b, y_k + \frac{3k_1 + 9k_2}{32}\right) \\k_4 &= h * f\left(t_k + \frac{12}{13}b, y_k + \frac{1932k_1 - 7200k_2 + 7296k_3}{2197}\right) \\k_5 &= h * f\left(t_k + b, y_k + \frac{439}{216}k_1 - 8k_2 + \frac{3680}{513}k_3 - \frac{845}{4104}k_4\right) \\k_6 &= h * f\left(t_k + \frac{b}{2}, y_k + \frac{8}{27}k_1 + 2k_2 - \frac{3544}{2565}k_3 + \frac{1859}{4104}k_4 - \frac{11}{40}k_5\right)\end{aligned}$$

El tamaño de paso óptimo q se determina en la práctica de otra manera, para que valga la pena el incremento del costo.

Debido al alto costo computacional, en realidad se tiende a elegir un valor de q conservador. Para el método RKF con $n=4$, la elección usual es, de hecho, la siguiente:

$$q = \sqrt[4]{\frac{\varepsilon h}{2|\tilde{y}_{i+1} - y_{i+1}|}} = 0,8409 * \sqrt[4]{\frac{\varepsilon h}{|\tilde{y}_{i+1} - y_{i+1}|}}$$

El valor de q determinado en el i -ésimo paso, se usa con dos propósitos diferentes:

- 1) para rechazar la elección inicial h en el paso si es necesario, y recalcularlo usando $q*b$, y
- 2) para predecir una elección inicial h suficientemente ajustada para el paso siguiente $i+1$.

B.- Métodos multipaso:

Frente a los métodos anteriores, conocidos como *métodos de 1 paso*, se colocan los *métodos multipaso*. La diferencia estriba en lo siguiente:

Los *métodos monopaso* involucran información de uno de los puntos de red anteriores t_i en la aproximación para un punto de red t_{i+1} , y no retienen tal información para aproximaciones futuras. Consecuentemente, toda la información utilizada por esos métodos se obtiene dentro del intervalo en el cual se está aproximando la solución.

Los *métodos multipaso*, por el contrario, abarcan aproximaciones en más de un punto de red anterior a t_{i+1} , para determinar la aproximación en dicho punto.

Los métodos multipaso son también conocidos como métodos de *predicción-corrección* o *predictor-corrector*.

5.- Método de Gragg de 2 pasos de orden 2:

Realiza los cálculos invocando al método de Euler y también basándose en el método del punto medio para la expansión del error. Para el problema del valor inicial $\{ y' = f(t, y); y(a) = \alpha; / a \leq t \leq b \}$ el método de Gragg inicialmente propone este esquema:

$$\begin{aligned}
 y_0 &= \alpha \\
 y_1 &= y_0 + h * f(t_0, y_0) \xrightarrow{\text{basado en Euler}} \text{invocando método de Euler} \\
 y_{i+1} &= y_{i-1} + 2h * f(t_i, y_i) \xrightarrow{\hspace{1.5cm}} \text{basado en el método del Punto medio}
 \end{aligned}$$

6.- Método de Gragg extrapolado:

Ahora se incorpora la técnica de extrapolación expuesta en capítulos anteriores al esquema de diferencia de Gragg planteado originariamente para obtener un procedimiento de extrapolación. Así logra fórmulas de mayor orden (con un error menor) desde datos anteriores.

$y_{1,1} = f(t + h_0)$		
$y_{2,1} = f(t + h_1)$	$y_{2,2} = \frac{h_0^2 y_{2,1} - h_1^2 y_{1,1}}{h_0^2 - h_1^2}$	
$y_{3,1} = f(t + h_2)$	$y_{3,2} = \frac{h_1^2 y_{3,1} - h_2^2 y_{2,1}}{h_1^2 - h_2^2}$	$y_{3,3} = \frac{h_0^2 y_{3,2} - h_2^2 y_{2,2}}{h_0^2 - h_2^2}$

C.- Problemas de contorno:

Un *problema de contorno* es un *problema de valor de frontera* para ecuaciones diferenciales ordinarias. Estos problemas involucran una ecuación diferencial de segundo orden de la forma $\{ y'' = f(t, y, y'), / a \leq t \leq b \}$ junto con las condiciones de frontera $y(a) = \alpha, y(b) = \beta$.

Teorema 40: Problema de contorno: si la función f en el problema de valor de frontera $\{ y'' = f(t, y, y'), / a \leq t \leq b, y(a) = \alpha, y(b) = \beta \}$ y $\partial f / \partial y \wedge \partial f / \partial y'$ son continuas en el conjunto $D = \{ (t, y, y'), / a \leq t \leq b, -\infty \leq y \leq \infty, -\infty \leq y' \leq \infty, \}$, y si se cumple que:

- 1.- $\frac{\partial f}{\partial y}(t, y, y') > 0 \quad / \forall (t, y, y') \in D, y$
- 2.- Existe una constante M con $\frac{\partial f}{\partial y'}(t, y, y') < M \quad / \forall (t, y, y') \in D,$

entonces el problema de contorno tiene una única solución.

Def. 72: Problema lineal de contorno: problema de contorno con la ecuación diferencial lineal, o sea, cuando puede expresarse en la forma $f(x, y, y') = p(x)y' + q(x)y + r(x)$

Teorema 41: Problema lineal de contorno: suponiendo que la función f en el problema lineal de contorno $\{ y'' = p(t)y' + q(t)y + r(t), a \leq t \leq b, y(a) = \alpha, y(b) = \beta \}$ satisface que p, q y r son continuas en $[a, b]$, y $q(t) > 0$ en $[a, b]$, entonces tiene una solución única.

Para aproximar la solución única, se consideran 2 problemas de valor inicial especiales:

$$1.- y'' = p(t)y' + q(t)y + r(t), a \leq t \leq b, y(a) = \alpha, y'(a) = 0, \text{ con solución } y_1,$$

$$2.- y'' = p(t)y' + q(t)y, a \leq t \leq b, y(a) = 0, y'(a) = 1, \text{ con solución } y_2,$$

entonces la solución al problema inicial vendrá dada por:

$$y(t) = y_1(t) + \frac{\beta - y_1(b)}{y_2(b)} y_2(t).$$

Este tipo de problema se suele solucionar con *métodos de disparo* o *tanteo*, que disparan en a 2 aproximaciones hasta que llega a b : $y'(a) = \alpha$ y $y'(b) = \beta$

El *método de disparo lineal* está basado en este reemplazo del problema de contorno por estos 2 problemas de valor inicial.

Ahora planteamos el sistema de ecuaciones diferenciales:

$$\{ y(t) = u(t) + C^*v(t); y'' = u''(t) + C^*v''(t) \}, \text{ de donde, sustituyendo:}$$

$$y'' = p^*u' + q^*u + r + C^*(p^*v' + q^*v) = p^*y' + q^*y + r$$

$$/ y' = p^*u' + C^*p^*v'; y = q^*u + q^*v$$

$$y(a) = u(a) + C^*v(a) = \alpha + 0 = \alpha$$

$$y(b) = u(b) + C^*v(b) = \beta \Rightarrow C = [\beta - u(b)] / v(b)$$

Nota 35: Calcular $u(b)$ y $v(b)$ por RK, y obtener el valor de C , que es una constante.

La solución general es:

$$y(t) = u(t) + \frac{\beta - u(b)}{v(b)} v(t)$$

o lo que es lo mismo

$$y(t) = u(t) + C \cdot v(t)$$

y para la derivada

$$y' = u'(t) + C^*v'(t)$$

Tema Misceláneo

A.- Notas de conocimientos útiles:

1.- Binomio de Newton

$$(a+b)^n = \sum_0^n \binom{n}{i} * a^{n-i} * b^i \mapsto \binom{n}{i} = \frac{n!}{i!(n-i)!}$$

2.- Fórmula de Taylor de 1 variable:

$$f(x+\delta) = \sum_0^n f^{(i)}(x) * \frac{\delta^i}{i!} + \overbrace{f^{(n+1)}(\alpha) * \frac{\delta^{n+1}}{(n+1)!}}^{\text{ERROR}} \approx f(x) + \delta * f'(x)$$

3.- Fórmula de Taylor de 2 variables:

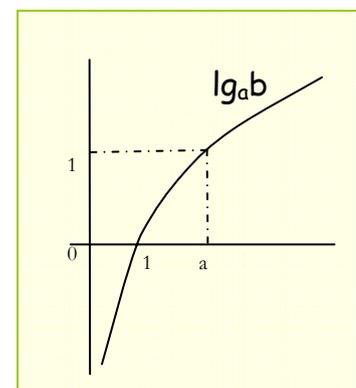
$$f(x+\delta, y+\tau) \approx \sum_0^n \frac{f(x, y)}{i!} * \left(\frac{\partial f(x, y)}{\partial x} * \delta + \frac{\partial f(x, y)}{\partial y} * \tau \right)^i$$

4.- Polinomio de Taylor:

$$\left\{ \begin{array}{l} P_n(x) = f(x_0) + f'(x_0) * (x-x_0) + \frac{f''(x_0)}{2!} (x-x_0)^2 + \dots + \frac{f^{(n)}(x_0)}{n!} (x-x_0)^n \\ P_n(x) = \sum_{i=0}^n \frac{f^{(i)}(x_0)}{i!} (x-x_0)^i \end{array} \right. \quad R_n(x) = \frac{f^{(n+1)}(\xi(x))}{(n+1)!} (x-x_0)^{n+1}$$

5.- Relaciones logarítmicas:

$$\left\{ \begin{array}{l} \lg a * b = \lg a + \lg b \\ \lg \frac{a}{b} = \lg a - \lg b \\ \lg a^b = b * \lg a \\ \lg_a b = x \Leftrightarrow a^x = b \end{array} \right\} \quad \left\{ \begin{array}{l} \lg 0 = -\infty \\ \lg 1 = 0 \\ \lg \infty = \infty \\ \lg_a a^b = b \end{array} \right\}$$



6.- Relaciones trigonométricas:

(a) →

Grados	0	30	45	60	90
Radianes	0	$\pi/6$	$\pi/4$	$\pi/3$	$\pi/2$
Cos	0	1/2	$\sqrt{2}/2$	$\sqrt{3}/2$	1
Sen	1	$\sqrt{3}/2$	$\sqrt{2}/2$	1/2	0
tg	0	$\sqrt{3}/3$	1	$\sqrt{3}$	∞

$$(b) \rightarrow \text{sen}^2 x + \text{cos}^2 x = 1 \quad \left\{ \begin{array}{l} \text{sen} \frac{x}{2} = \sqrt{\frac{1 - \text{cos} x}{2}} \\ \text{cos} \frac{x}{2} = \sqrt{\frac{1 + \text{cos} x}{2}} \end{array} \right.$$

$$(c) \rightarrow \left\{ \begin{array}{l} \text{sen} x \pm \text{sen} y = 2 \text{sen} \frac{x \pm y}{2} * \text{cos} \frac{x \mp y}{2} \\ \text{cos} x + \text{cos} y = 2 \text{cos} \frac{x + y}{2} * \text{cos} \frac{x - y}{2} \\ \text{cos} x - \text{cos} y = 2 \text{sen} \frac{x + y}{2} * \text{sen} \frac{x - y}{2} \end{array} \right.$$

$$(d) \rightarrow \left\{ \begin{array}{l} \text{sen}(x \pm y) = \text{sen} x * \text{cos} y \pm \text{cos} x * \text{sen} y \\ \text{cos}(x \pm y) = \text{cos} x * \text{cos} y \mp \text{sen} x * \text{sen} y \end{array} \right\} \quad \text{tg}(x \pm y) = \frac{\text{tg} x \pm \text{tg} y}{1 \mp \text{tg} x * \text{tg} y}$$

7.- Normas vectoriales:

Las normas vectoriales miden la distancia (al error) entre 2 vectores.

$$(a) \text{ norma euclídea (longitud real): } \|x\|_2 = \sqrt{x_1^2 + \dots + x_n^2}$$

$$(b) \text{ norma infinita: } \|x\|_\infty = \max_{1..n} |x_i|$$

$$(c) \text{ criterio _ paro: } \|p_k - p_{k-1}\| < \text{tol} \Rightarrow \text{fin}$$

Nota 36: Usaremos las normas equivalentes (a) y (b) como criterios de paro (c).

8.- Phi: Número de oro de la Ciencia:

Phi, considerado número más bello del universo, responde a la razón $\frac{1 + \sqrt{5}}{2} = 1.618034$.

Revela una *Porción Áurea*. Es muy importante en la Arquitectura, la Astronomía, etcétera. Tiene un papel básico en cuanto a molde constructivo de la Naturaleza. La famosa serie de *Fibonacci* se obtiene sumando los dos números precedentes; pero también los cocientes de los precedentes forman una progresión que tiende a Phi.

Nota 37: Ver anexo 4 sobre el número Phi.

B.- Formulas de derivadas e integrales sencillas

FUNCIÓN		DERIVADA	FUNCIÓN		INTEGRAL
$k * f$		$k * f'$	$f^n * f'$		$\frac{f^{n+1}}{n+1} \Leftrightarrow n \neq -1$
$f \circ g$		$f'(g) * g'$	$\frac{f'}{f}$		$\ln f$
$f \pm g$		$f' \pm g'$	$e^f * f'$		e^f
$f * g$		$f' * g + f * g'$	$a^f * f'$		$\frac{a^f}{\ln a}$
f/g		$\frac{f' * g - f * g'}{g^2}$	$\frac{f'}{\sqrt{f^2 + 1}}$		$\ln(f + \sqrt{f^2 + 1})$
f^n		$n * f' * f^{n-1}$	$\frac{f'}{\sqrt{f^2 - 1}}$		$\ln(f + \sqrt{f^2 - 1})$
$\lg_a f$		$\frac{f' * \lg a * e}{f}$	a		$a * x$
a^f		$\frac{f' * a^f * \ln a}{f}$	$\frac{f'}{f}$		$\ln f$
f^g		$g' * g^f * \ln f + f' * g * f^{g-1}$	$\cos(a + b * x)$		$\frac{\text{sen}(a + b * x)}{b}$
$\text{sen } f$		$f' * \cos f$	$\sqrt{1 + \cos x}$		$\sqrt{1 + \cos x}$
$\cos f$		$-f' * \text{sen } f$	$x * \sqrt{1 + x^2}$		$3 * \sqrt[3]{(x^2 + 1)^2}$
$\text{tg } f$		$\frac{f'}{\cos^2 f} = f' + f * \text{tg}^2 f$	$\ln x$		$x * \ln x - x$
$\arcsen f$		$\frac{f'}{\sqrt{1 - f^2}}$	$\frac{\cos x}{1 + \cos x}$		$x - \text{tg} \frac{x}{2}$
$\arccos f$		$-\frac{f'}{\sqrt{1 - f^2}}$	$\frac{\sqrt{\ln x}}{x}$		$\frac{2}{3} \sqrt{\ln^3 x}$
$\arctg f$		$\frac{f'}{1 + f^2}$	$\frac{x}{\sqrt{2x^2 + 3}}$		$\frac{1}{2} \sqrt{2x^2 + 3}$
$\text{arccosec } f$		$\frac{-f'}{f * \sqrt{1 - f^2}}$	$\frac{1}{\sqrt{4 - (x + 2)^2}}$		$\arcsen\left(\frac{x + 2}{2}\right)$
$\text{arcsec } f$		$\frac{f'}{f * \sqrt{1 - f^2}}$	$\frac{\text{sen } 2x}{(1 + \cos 2x)^2}$		$\frac{1}{2 * (1 + \cos 2x)}$
$\text{arctg } f$		$\frac{-f'}{1 - f^2}$	$e^x \text{sen } x$		$\frac{e^x \text{sen } x - e^x \cos x}{2}$

C.- Notas de utilidad para los ejercicios

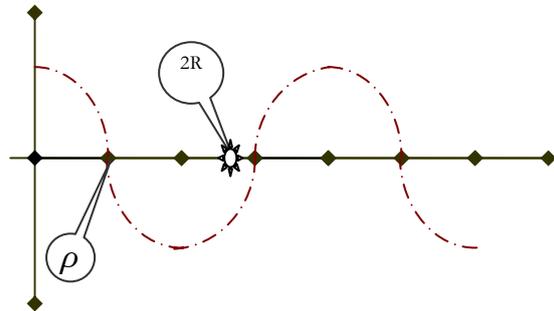
1.- Problema de la esfera.

En una esfera de radio R , con densidad ρ , calcular el volumen de esfera que se hunde en el agua (masa de agua que desplaza al hundirse).

E.g.: $R=20$ cm; $\rho=0,61$ (madera de pino)

La solución se calcula con:

$$\rho^3 - 3R\rho^2 + 4R^3\rho = 0$$



2.- Introducción de flags para simular un debugger.

Cada x líneas se deben introducir flags (valores “bandera” de las variables y principales funciones), para prever posibles fallos y medir el tiempo con fines comparativos o para que se delaten cuellos de botella (operaciones que ocupen en exceso el procesador).

En Matlab es muy fácil; sólo hay que escribir la lista de variables separadas por comas, y su valor se visualiza automáticamente. ¡Ojo con funciones que alteren el valor de las entradas!

3.- Indicar la potenciación de matrices en Matlab.

Matlab no hace potencias directas de matrices, y debe indicársele que potencie cada elemento con la notación punto tras el signo que cierre la matriz. E.g.: problema de la esfera:

$$y = (x.^3) - 3*0'2*(x.^2) + 4*(0'2^3)*0'61 = 0 ;$$

4.- Sumar n números eficientemente.

Tomamos 2^k números (completar sumando 0 hasta tal cantidad, si es necesario), debemos hacerlo de manera decreciente para minimizar el error de cálculo.

5.- Restar 2 números muy próximos.

Para restar 2 números próximos debemos sumar y restar dicha resta por su conjugado.

$$[(a-b)*(a+b)/(a+b)].$$

6.- Matrices inversas en Matlab.

Las matrices inversas son muy caras computacionalmente, pero en Matlab se resuelven con facilidad si se plantean como este sistema de ecuaciones:

$$X = A^{-1}*b, \text{ con } n^3 \text{ operaciones, se reduce en Matlab a } X=A \setminus b \text{ / } \frac{n^3}{3} \text{ operaciones.}$$

7.- Adecuar cálculos.

Hay que evitar cálculos excesivos, porque llevan a errores que se potencian cada iteración. Igualmente deben evitarse las derivadas numéricas, que dan inestabilidad a los métodos.

Al tratar con números muy pequeños, por ejemplo, cuando se controla el error, debemos usar logaritmos.

$$\log_{10} 10^{-n} = -n ,$$

O sea que:

$$x=a*10^{-n} \Rightarrow \log_{10} a*10^{-n} =b$$

habrá E[b] cifras exactas como mínimo, donde E[b] es la función parte entera de b.

$$\log_{10}|E_{n+1}| \approx \log_{10} \lambda + \alpha * \log_{10}|E_n| \Rightarrow \alpha \approx \frac{\log_{10}|E_{n+1}|}{\log_{10}|E_n|} = \frac{\ln|E_{n+1}|}{\ln|E_n|}$$

8.- Fórmulas de resolución de ecuaciones matemáticas.

Desde la existencia de *Pitágoras* (S. III A.C.), y más, desde el imperio babilónico, tienen soluciones con pequeñas fórmulas las ecuaciones como:

$$2x + 1 = 0; \quad \text{ó} \quad x^2 + x - 7 = 0$$

Delferro, Tartaglia y Cardán encontraron las raíces de ecuaciones de tercer grado, y desde 1540, se conoce la solución para ecuaciones de cuarto grado, como los tipos:

$$x^3 + x^2 - x + 7 = 0 \quad \text{ó} \quad x^4 + x - 2 = 0 ;$$

Pero otras de tipo complejo o de grado superior, no tienen solución fácil, como estas:

$$\sin^2 x + 2x - 7x^3 + e^x = 0;$$

La resolución de ecuaciones de grado superior a 4 había sido estudiada en serio por Tschirnhaus , Euler , Bézout , Vandermonde , Edward Waring y Lagrange, entre otros. Pero fue *Abel*, matemático del S.XIX, quien demostró que para ecuaciones de grado superior a 4 no existen fórmulas simbólicas (Aritmética exacta), sino que se usan sólo métodos numéricos aproximados (Aritmética finita).

Galois, antes de morir, deja sus trabajos a un amigo, que se los entrega a *Gauss*, y es *Gauss* quien termina sus investigaciones. *Gauss* murió en 1855, pero ya había escrito una tesis titulada “*Tesis fundamental del Álgebra*”, donde demostró que una ecuación de grado n tiene n raíces.

9.- Descubrimiento de las Derivadas y las Integrales.

Newton, estudioso de la Física como rama de la Matemática, define derivadas o *fusiones*, para calcular velocidad y aceleración, y luego define integrales o *cuadraturas* para calcular el *Trabajo*.

Índice del temario

Tema 1: Análisis del error.....	1
A.- Estudio y tipología del error :	1
B.- Representación numérica:.....	1
C.- Ejemplos:	3
Tema 2: Ecuaciones no lineales.....	5
A.- Introducción:.....	5
B.- Métodos iniciales:.....	6
1.- Método de Bisección:	6
2.- Cálculo de las cotas de una función:	7
3.- Método del Punto Fijo:	7
4.- Método de la Regula Falsi (posición falsa).....	9
C.- Métodos de mayor convergencia:.....	10
1.- Método de Newton-Raphson-Fourier:	10
a.- Elección del punto inicial de iteración p_0	11
b.- Convergencia y error:.....	12
2.- Método de la Secante:.....	13
3.- Método “Leap-frog”:.....	13
4.- Método de Horner (valor de un polinomio en punto).....	14
Tema 3: Sistemas de ecuaciones lineales	17
A.- Álgebra matricial para sistemas lineales:.....	17
1.- Planteamiento de sistemas de ecuaciones lineales:.....	17
2.- Normas:.....	18
3.- Caracterización de una matriz:	19
B.- Planteamiento de los algoritmos iterativos:	20
C.- Métodos iterados:	22
1.- Método de Jacobi:.....	22
2.- Método de Gauss-Seidel:	23
3.- Método generalizado de Jacobi-Gauss-Seidel:.....	24
4.- Métodos SOR (Successive Over-Relaxation):	24
Tema 4: Sistemas de ecuaciones no lineales	26
A.- Método de Newton global:	26
B.- Método del gradiente o máximo descenso:.....	27
C.- Método iterativo del Punto Fijo:	28
D.- Método de Seidel:.....	29
E.- Método iterativo de Newton:.....	30
Tema 5: Interpolación polinomial.....	31
A.- Polinomio de Interpolación:	31
B.- Polinomio de Interpolación de Lagrange:	32
C.- Interpolación de Newton:	33
1.- Método de las Diferencias Divididas:	33
2.- Método de las Diferencias finitas:	35
D.- Nodos de Chébyshev:.....	36
1.- Polinomios de Chébyshev y propiedades:.....	36
2.- Nodos de Chébyshev en otros intervalos:.....	37
E.- Método de Extrapolación de Richardson:	37

Tema 6: Ajuste de curvas.....	39
A.- Mínimos cuadrados:	39
1.- Tipos de errores:.....	39
2.- Rectas de regresión en mínimos cuadrados:.....	39
3.- Ajuste exponencial:	40
4.- Ajuste de curvas por linealización de los datos:.....	40
B.- Interpolación polinomial a trozos:.....	41
1.- Cerchas o 'splines':.....	41
2.- Cerchas cúbicas:	41
a.- Construcción de una cercha cúbica interpoladora:	42
b.- Tipos de cerchas cúbicas interpoladoras:.....	42
C.- Series de Fourier y polinomios trigonométricos:	43
1.- Series de Fourier:.....	43
2.- Aproximación mediante polinomios trigonométricos:.....	45
Tema 7: Diferenciación e integración numérica.....	46
A.- Derivación numérica:	46
1.- Método de las diferencias centradas:	46
2.- Método de extrapolación de Richardson:	47
B.- Integración numérica:.....	48
1.- Fórmulas de Newton-Côtes:.....	48
2.- Método del Trapecio:.....	48
3.- Método del Trapecio compuesto:.....	49
4.- Regla del Trapecio recursiva:.....	49
5.- Método del Polinomio:.....	50
6.- Método de integración de Romberg:.....	50
C.- Integración o cuadratura adaptativa:	51
7.- Método de Simpson:.....	51
8.- Regla compuesta de Simpson: (refinamiento).....	52
Tema 8: Problemas de valor inicial y de contorno.....	53
A.- Introducción a la ecuación diferencial:	53
1.- Método poligonal de Euler:	54
2.- Método de la serie de Taylor:	54
3.- Métodos de Runge-Kutta:.....	55
4.- Control del error con métodos de Runge-Kutta-Fehlberg (RKF):.....	55
B.- Métodos multipaso:	56
5.- Método de Gragg de 2 pasos de orden 2:.....	57
6.- Método de Gragg extrapolado:	57
C.- Problemas de contorno:.....	57
Tema Misceláneo	59
A.- Notas de conocimientos útiles:.....	59
B.- Formulas de derivadas e integrales sencillas.....	61
C.- Notas de utilidad para los ejercicios	62
Índice del temario	64