RESUMEN | Modelación Numérica (1C2025)

☒ MODELADO:

	DATOS ⇨	ALGORITMO ⇨	RESULTADOS
	"Entrada"	Proceso / Precisión	"Salida"
Mundo ideal	Exactos	∞ / ∞	Exactos
Mundo real	Con error	finito / finita	Con error

ERRORES:

☒ FUENTES DE ERROR:

Errores de entrada: Mediciones, cálculos previos, etc.

Errores de truncamiento: Proceso finito (los cálculos no pueden ser al infinito)

Errores de redondeo: Precisión finita

Otras fuentes de error: Modelado o errores humanos

☒ ERROR ABSOLUTO Y ERROR RELATIVO:

a : valor exacto (en gral. desconocido)

$\hat{a}$: valor estimado

Error absoluto exacto (con signo) δa: $\boxed{a = \hat{a} + \delta a}$

Error relativo exacto (con signo) r_a: $\boxed{r_a = \dfrac{\delta a}{a}}$

$$\left.\begin{matrix} a = \hat{a} + \delta a \\ r_a = \dfrac{\delta a}{a} \end{matrix}\right\} \Rightarrow \boxed{a = \frac{\hat{a}}{1 - r_a}}$$

* Cotas y errores se expresan con 1 o a lo sumo 2 dígitos.

☒ COTAS DE ERROR ABSOLUTO Y RELATIVO:

Cota del error absoluto (positiva) Δa: $\boxed{a = \hat{a} \pm \Delta a}$

Cota del error relativo (positiva) R_a: $a = \hat{a}(1 \pm R_a) \Rightarrow \boxed{R_a = \dfrac{\Delta a}{\hat{a}}}$

☒ EXPRESIONES DE LOS ERRORES:

Expresión en función de la cota de error absoluto. No se deben expresar los dígitos no significativos.

El valor a queda representado como $a = \hat{a} \pm \Delta a$.

DÍGITOS SIGNIFICATIVOS:

$$\Delta a = \underbrace{0{,}0000\ldots00}_{\text{Decimales significativos (ds)}}\overset{\text{Decimal medianamente significativo (dms)}}{4}\underbrace{34\ldots}_{\text{Decimales no significativos}}$$

Decimal medianamente significativo (dms) → Solo si < 5

→ puede variar 4 hacia arriba o 4 hacia abajo.

Dígito Significativo (DS): Desde el primer dígito hasta el último ds.

Dígito Medianamente Significativo (DMS): Coincide con el dms.

REDONDEO:

Redondeo truncado: Se escribe hasta el último DS o hasta el DMS, si existe.

Redondeo simétrico: El último DS o el DMS, si existe, se deja tal cual si el primer dígito no significativo es menor que 5, o se lo incrementa en 1 en caso contrario.

☒ REPRESENTACIÓN INTERNA DE NÚMEROS REALES:

PUNTO FLOTANTE:

Notación exponencial ($\forall\ a \in \mathbb{R}$):

$a = m \cdot 10^q$

q entero: Exponente

m: Mantisa

Normalización → $0,1 \leq |m| < 1$

Representación flotante normalizada (computadora):

$\hat{a} = \hat{m} \cdot 10^q$

q entero: simple precisión ± 39 $(\cong 2^{2^7-1})$

doble precisión ± 308 $(\cong 2^{2^{10}-1})$

Mantisa normalizada: $0,1 \leq |\hat{m}| < 1$

$\hat{m}$ tiene t dígitos (precisión de los números que puede representar la máquina).

Error de representación: $\boxed{a - \hat{a} = (m - \hat{m})\, 10^q}$

OBS: Cuando el resultado de una operación resulta en un valor con exponente fuera del rango representable se produce un error:

si $q > 0$ ⇒ overflow → Interrupción del programa

si $q < 0$ ⇒ underflow → Usualmente se asigna 0.

Cota del error absoluto de la mantisa: $\boxed{|m - \hat{m}| \leq \begin{cases} 0,5 \cdot 10^{-t} \\ 10^{-t} \end{cases}}$ Redondeo simétrico / Redondeo truncado

Cota del error relativo por representación interna: $\boxed{\left|\dfrac{\hat{a} - a}{a}\right| \leq \underbrace{u}_{\text{Unidad de máquina}} = \begin{cases} 0,5 \cdot 10^{-t+1} \\ 10^{-t+1} \end{cases}}$ R. simétrico / R. truncado

☒ PROPAGACIÓN DE ERRORES

Sea $y = F(x_1; \dots; x_n)$, con $x_j = \hat{x}_j + \delta x_j$

Entonces:

$y = \hat{y} \pm \Delta y$ con $\hat{y} = F(P)$ siendo $P = (\hat{x}_1; \dots; \hat{x}_n)$

ERRORES DE ENTRADA:

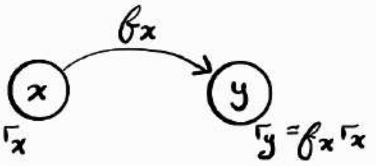
Fórmula general de propagación de errores de entrada: $\boxed{\delta y = \sum_{j=1}^{n} \left.\dfrac{\partial F}{\partial x_j}\right|_P \delta x_j}$

Cota del error absoluto: $\boxed{\Delta y = \sum_{j=1}^{n} \left| \left.\dfrac{\partial F}{\partial x_j}\right|_P \right| \Delta x_j}$

CASOS PARA F(x):

$$\delta y = F'(x)\, \delta x$$

$$r_y = \frac{x}{y} F' r_x = \underbrace{f_x}_{\text{factor de amplificación}} r_x \quad \Rightarrow$$



← Forma gráfica (celular)

CASOS PARA $F(x_1; x_2)$:

	EXACTO		COTA	
$+$	$\delta y = \delta x_1 + \delta x_2$	$r_y = \frac{x_1}{y} r_1 + \frac{x_2}{y} r_2$	$\Delta y = \Delta x_1 + \Delta x_2$	$R_y = \left\lvert \frac{x_1}{x_1+x_2} \right\rvert R_1 + \left\lvert \frac{x_1}{x_1+x_2} \right\rvert R_2$
$-$	$\delta y = \delta x_1 - \delta x_2$	$r_y = \frac{x_1}{y} r_1 - \frac{x_2}{y} r_2$	$\Delta y = \Delta x_1 + \Delta x_2$	$R_y = \left\lvert \frac{x_1}{x_1-x_2} \right\rvert R_1 + \left\lvert \frac{x_1}{x_1-x_2} \right\rvert R_2$ (*)
$\cdot$	$\delta y = x_2 \delta x_1 + x_1 \delta x_2$	$r_y = r_1 + r_2$	$\Delta y = \lvert x_2 \rvert \Delta x_1 + \lvert x_1 \rvert \Delta x_2$	$R_y = R_1 + R_2$
$/$	$\delta y = \frac{\delta x_1}{x_2} + \frac{\delta x_2}{x_1}$	$r_y = r_1 - r_2$	$\Delta y = \left\lvert \frac{1}{x_2} \right\rvert . \Delta x_1 + \left\lvert \frac{x_1}{-x_2^2} \right\rvert \Delta x_2$	$R_y = R_1 + R_2$

$$\underbrace{\qquad\qquad}_{r_y = f_1 r_1 + f_2 r_2}$$

Se suele usar multiplicar y dividir por el conjugado. → para eliminar la resta.

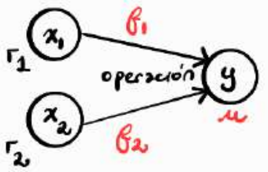
(*) Efecto cancelación de términos: Si $x_1 \cong x_2 \Rightarrow R_y$ no está acotada. Conviene usar otra expresión equiv.

ERRORES DE TRONCAMIENTO:

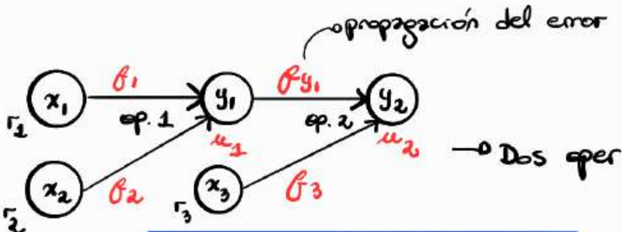
Cota del error de truncamiento: $\boxed{\Delta y \sim \lvert y_{[N+1]} - y_{[N]} \rvert}$

ERRORES DE REDONDEO:

Gráficas de procesos:



→ Una operación



→ Dos operaciones

Regla general de propagación de errores relativos:

$$\boxed{r_y = \sum_{j=1}^{n} f_j r_j + \sum_{i=1}^{M} g_i u_i}$$

n: Cantidad de datos de entrada.
f_j: Factores de amplificación de errores de entrada. } Dep. del problema
M: Cantidad de operaciones del algoritmo de cálculo.
g_i: Factores de amplificación de errores de redondeo. } Dep. del problema y del algoritmo.

Cota del error relativo: $\boxed{R_y = \sum_{j=1}^{n} \lvert f_j \rvert R_j + F_u \cdot u}$

Factor global de amplificación de errores de redondeo: $\boxed{F_u = \sum_{i=1}^{M} \lvert g_i \rvert}$

☒ ESTABILIDAD:

ESTABILIDAD DEL PROBLEMA (matemático):

$\lvert f_j \rvert < 1 \Rightarrow$ problema estable (o bien condicionado) para x_j
$\lvert f_j \rvert >> 1 \Rightarrow$ problema inestable (o mal condicionado) para x_j

Número de condición del Problema: $\boxed{C_p = \sum_{j=1}^{n} \lvert f_j \rvert}$

$C_p < 1 \Rightarrow$ problema estable (o bien condicionado)
$C_p >> 1 \Rightarrow$ problema inestable (o mal condicionado)

ESTABILIDAD DEL ALGORITMO (numérico):

↳ Sensibilidad del resultado frente a errores de redondeo en las operaciones.

Número de condición del algoritmo: $\boxed{C_A = \frac{F_u}{C_p}}$

$C_A < 1$ ⇒ Algoritmo estable (o bien condicionado)

$C_A \gg 1$ ⇒ Algoritmo inestable (o mal condicionado)

$C_{A_1} < C_{A_2}$ ⇒ Algoritmo A_1 mejor condicionado que algoritmo A_2.

Condición de estabilidad del algoritmo: $\boxed{C_A \ll \frac{R}{u}}$

☒ PERTURBACIONES EXPERIMENTALES:

Alternativa para analizar propagación de errores de entrada y de redondeo.
Se analizan por separado y recien al final se hace un análisis de la estabilidad.

$$\begin{tikzcd}
\text{DATOS} \arrow[r, Rightarrow] \arrow[d] & \text{ALGORITMO} \arrow[r, Rightarrow] \arrow[d] & \text{RESULTADOS} \arrow[d] \\
(x_1; \dots ; x_n) \arrow[d] & \text{Proceso } (t) \arrow[d] & y \arrow[d] \\
(x_1; \dots ; x_n) & \text{Proceso } (2t) & y_{2t}
\end{tikzcd}$$

(programamos en doble precisión: $u = 0{,}5 \cdot 10^{-t+1}$)

Variación relativa del resultado por cambio de precisión: $\boxed{\omega_{2t} = \left| \frac{y_{2t} - y}{y_{2t}} \right|}$

$\boxed{F_u = \frac{\omega_{2t}}{u}}$ $\boxed{C_A = \frac{F_u}{C_p}}$

Pasos (asumiendo $y = F(x_1; x_2)$):

1. Se miden valores $x_1 = a \pm \Delta a$ y $x_2 = b \pm \Delta b$.
2. Teniendo en cuenta Δa y Δb, variamos los valores de a y b por separado y calculamos $y = F(x_1; x_2)$ para esos valores:

x_1	x_2	$F(x_1;x_2)$
a_1	b	$F(a_1;b)$
a	b_1	$F(a;b_1)$
a	b	$F(a;b)$
a	b_2	$F(a;b_2)$
a_2	b	$F(a_2;b)$

3. $\hat{F} = F(a;b)$
4. Utilizamos los valores para calcular las derivadas de la cota de error:

$$\Delta F = \left| \frac{\partial F}{\partial a} \right| \Delta a + \left| \frac{\partial F}{\partial b} \right| \Delta b$$

$$\left| \frac{\partial F}{\partial a} \right| \cong \left| \frac{\delta F}{\delta x_1} \right| = \left| \frac{F(a_1;b) - F(a_2;b)}{a_1 - a_2} \right|$$

$$\left|\frac{\partial F}{\partial b}\right| \cong \left|\frac{\delta F}{\delta x_2}\right| = \left|\frac{F(a\,;\,b_1) - F(a\,;\,b_2)}{b_1 - b_2}\right|$$

Al obtener ΔF, la redondeamos simétricamente.

5. Podemos escribir $F = \hat{F} \pm \Delta F$ con las correctas cifras significativas.

Nota: La técnica se utiliza cuando no podemos calcular las derivadas de la cota ya sea por ser muchas o por no tener el algoritmo $y = F(x_1; \dots; x_n)$.

SISTEMAS DE ECUACIONES LINEALES: MÉTODOS ALGEBRÁICOS

Problema tipo:

Resolver el SEL definido por las n ecuaciones E_i:

$$\begin{aligned} E_1 &= a_{11}x_1 + a_{12}x_2 + \dots + a_{1K}x_K = b_1 \\ E_2 &= a_{21}x_1 + a_{22}x_2 + \dots + a_{2K}x_K = b_2 \\ &\dots \\ E_n &= a_{n1}x_1 + a_{n2}x_2 + \dots + a_{nK}x_K = b_n \end{aligned}$$

☒ MÉTODO DE ELIMINACIÓN DE GAUSS: $2/3\, n^3$ operaciones.

Buscamos $x \in \mathbb{R}^{n \times 1}$ tal que $Ax = b$, donde:

$$A = \begin{pmatrix} a_{11} & a_{12} & \dots & a_{1K} \\ a_{21} & a_{22} & \dots & a_{2K} \\ \dots & & & \\ a_{n1} & a_{n2} & \dots & a_{nK} \end{pmatrix} \begin{matrix} E_1 \\ E_2 \\ \\ E_n \end{matrix} \in \mathbb{R}^{n \times n} \qquad b = \begin{pmatrix} b_1 \\ b_2 \\ \vdots \\ b_n \end{pmatrix} \in \mathbb{R}^{n \times 1}$$

Resolvemos el sistema de ecuaciones triangulando la matriz $(A \mid b)$ → Implica muchas operaciones.

PROPAGACIÓN DE ERRORES DE ENTRADA:

$$(A + \delta A)(x + \delta x) = (b + \delta b) \Rightarrow \boxed{\delta x = A^{-1}.\,\delta b - A^{-1}.\,\delta A . x}$$

↳ perturbación en la sol. (por prop. de errores de entrada en A y b).

COTA: $$\boxed{\frac{\|\delta x\|}{\|x\|} \leq \underbrace{\|A\|\|A^{-1}\|}_{K(A)} \left(\frac{\|\delta b\|}{\|b\|} + \frac{\|\delta A\|}{\|A\|}\right)}$$

→ vale porque $\|b\| \leq \|A\|\,\|x\|$

Número de condición de A: $\boxed{K = \|A\| . \|A^{-1}\|}$

K "pequeño" (orden 1) ⇒ A bien condicionada.

K "grande" (>> 1) ⇒ A mal condicionada ⇒ No es lo ideal resolver el SEL con el método de eliminación de Gauss.

Si $\delta A = 0 \Rightarrow C_p \leq K$

PROPAGACIÓN DE ERRORES DE REDONDEO:

Hay errores de redondeo por los límites de precisión.

$\hat{x}$: Solución obtenida

R: Residuo, $R = Ax - A\hat{x} = b - \hat{b} \iff \boxed{x - \hat{x} = \delta x = A^{-1}.\,R}$

COTA: $\boxed{\frac{\|\delta x\|}{\|x\|} \le K(A) \frac{\|R\|}{\|b\|}}$

Reducción de errores de redondeo:

- PIVOTEO PARCIAL:

  Si el pivote es pequeño en relación a los otros elementos de la columna que se está triangulando, entonces el multiplicador asociado puede incrementar los errores.

  Buscamos el menor r para el cual: $|a_{rk}| = \max |a_{jk}|$, con $k \le j \le n$, e intercambiamos las filas r y j.

- PIVOTEO TOTAL O COMPLETO:

  Aplicamos las permutaciones de filas del pivoteo parcial, y agregamos permutaciones de columnas seleccionando el "menor par" de enteros r y s para los cuales: $|a_{rs}| = \max |a_{ji}|$, con $k \le j, i \le n$. Luego se intercambian las filas r y j y las columnas i y s.

  ↳ En estos dos vectores tambien se deben realizar las operaciones de permutación para mantener el SLE equivalente.

☒ DESCOMPOSICIÓN LU: n^2 operaciones

Si existen L triangular inferior y U triangular superior tales que:

$A = LU \iff Ax = LUx = Ly = b \Rightarrow Ux = y$, y obtenemos x por sust. inversa.

↳ Vector incógnita transitorio → lo hayamos con sust. para adelante.

TEOREMA:

Dada $A \in \mathbb{R}^{n \times m}$ no singular, existe una matriz P de permutaciones, una matriz L triangular inferior con unos en la diagonal y una matriz U triangular superior tales que:

$PA = LU$, U: matriz triangulada que surge de la eliminación de Gauss.
L: se construye con los multiplicadores obtenidos en la eliminación.

☒ MATRICES ESPECIALES:

MATRIZ INVERSA:

Si conocemos A^{-1} ⇒ podemos resolver el sistema $Ax = b$ como $x = A^{-1} \cdot b$

↳ Podemos resolver el SEL $A \cdot A^{-1} = I$

MATRIZ DIAGONAL DOMINANTE:

A es diagonal dominante si: $|a_{ii}| > \sum_{\substack{j=i \\ j \neq i}}^{n} |a_{ij}|$.

Esta matriz no requiere pivoteo (parcial o total) al aplicar el método de eliminación de Gauss, siendo estable respecto a la propagación de errores de redondeo.

MATRIZ SIMÉTRICA Y DEFINIDA POSITIVA:

A simétrica: $a_{ij} = a_{ji}$

La eliminación de Gauss produce una matriz transformada también simétrica, por lo cual se reduce la cantidad de operaciones involucradas prácticamente a la mitad.

A simétrica definida positiva: $x' A x > 0 \quad \forall x \neq 0$

Este tipo de matriz no requiere pivoteo al aplicar el método de eliminación de Gauss, siendo estable respecto a la propagación de errores de redondeo.

MATRIZ BANDA:

Son matrices donde los elementos no nulos se localizan en una banda centrada a lo largo de la diagonal principal.

$a_{ij} = 0$ si $j > i + p$ o $i > j + q$

Ancho de banda: $w = p + q + 1$

Si no se aplica pivoteo, L y U también serán matrices banda con:

$l_{ij} = m_{ij} = 0$ si $j > i$ o $i > j + q$

$u_{ij} = 0$ si $i > j$ o $j > i + p$

} Cuando $p, q \ll n$ el costo computacional se reduce sensiblemente.

Si A es tridiagonal, $p = q = 1$ y solo se requieren $3(n-1)$ sumas y multiplicaciones, y $2n-1$ divisiones.

☑ REFINAMIENTO ITERATIVO:

Al calcular esto lo hacemos en doble precisión.

MINIMIZACIÓN DEL RESIDUO:

$\hat{x}^2 = \hat{x}^1 + \delta x^1$

$$R^1 = b - A\hat{x}^1 \iff A\,\delta x^1 = R^1 \Rightarrow A(\hat{x}^1 + \delta x^1) = b \iff A\hat{x}^2 = b$$ → Precisión mejorada.

El proceso se repite hasta alcanzar la precisión deseada:

$$\begin{cases} R^k = b - A\hat{x}^k \\ A\,\delta x^k = R^k \\ \hat{x}^{k+1} = \hat{x}^k + \delta x^k \end{cases}$$

Eurística: en cada paso del refinamiento, trabajando con t dígitos para la mantisa, se mejoran q dígitos de precisión (que convergen), donde:

$q = t - p$

$p = \log_{10}(\kappa(A))$

$\kappa(A) \cong \dfrac{\|\delta x^1\|}{\|x^1\|} \cdot 10^t$ → Se calcula solo en el primer paso.

SISTEMAS DE ECUACIONES LINEALES: MÉTODOS ITERATIVOS

Suponemos que x se puede resolver de forma iterativa:

$$\boxed{x^{(k)} = T \underbrace{x^{(k-1)}}_{\text{Solución anterior}} + c} \Rightarrow \lim_{k \to +\infty} x^{(k)} = x$$ (converge a la sol).

Según el $x^{(0)}$ (semilla) que elijamos, nos acercaremos más o menos a la solución.

T es una matriz y c un vector. Ambos constantes. Hay distintos métodos para obtenerlos.

☒ MÉTODO DE JACOBI → Despejando x_i de la ecuación E_i $(a_{ii} \neq 0)$.

Nos queda para cada x_i:

$$x_i = \underbrace{-\frac{1}{a_{ii}} \sum_{\substack{j=1 \\ j\neq i}}^{n} a_{ij}\, x_j}_{Tx} + \underbrace{\frac{b_i}{a_{ii}}}_{c} \quad ; \; i = 1, \dots, n$$

Entonces: $T_{ij} = -\dfrac{a_{ij}}{a_{ii}} (1 - \delta_{ij}) \quad ; \; i, j = 1, \dots, n$

$\delta = 1$ si $i = j$ ⇒ diagonal nula.
$\delta = 0$ si $i \neq j$

$$c = \frac{b_i}{a_{ii}} \quad ; \; i = 1, \dots, n$$

Finalmente: $\boxed{x_i^{(k)} = -\dfrac{1}{a_{ii}} \displaystyle\sum_{\substack{j=1 \\ j\neq i}}^{n} a_{ij}\, x_j^{(k-1)} + \dfrac{b_i}{a_{ii}}}$ $\; ; \; i = 1, \dots, n \; ; \; k \geq 1$ → Comenzamos eligiendo una semilla $x^{(0)}$.

Hallar $x^{(k)}$ de forma matricial:

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

D: Diagonal; L: Triangular inferior; U: Triangular superior

$$\Rightarrow \boxed{x^{(k)} = \underbrace{D^{-1}(L+U)}_{T_J} x^{(k-1)} + \underbrace{D^{-1}b}_{c_J}}$$

Criterio de corte:

Iteramos hasta que: $\boxed{\dfrac{\| x^{(k)} - x^{(k-1)} \|}{\| x^{(k)} \|} < \text{Tolerancia (ej. } = 10^{-3})}$

Condición de convergencia:

Para lograr la convergencia $\forall x^{(0)}$ (semilla) es necesario y suficiente que:

$\boxed{\rho(T_J) = \max\limits_{1 \leq i \leq n} |\mu_i| < 1}$; $\rho(T_J)$: radio espectral de T_J
μ_i: autovalores de T_J

Propiedad: Si A es estrictamente diagonal dominante converge $\forall x^{(0)}$.

ERROR DE REDONDEO:

- Opacado por los errores de truncamiento.
- Se "absorbe" entre iteraciones.

PROPAGACIÓN DE ERROR DE ENTRADA:

errores absolutos de entrada.

$$\boxed{\| \hat{x}^{(k)} - x \| \leq \frac{\| \delta T_J \|}{1 - \| T_J \|} \| x^{(k-1)} \| + \frac{\| \delta c \|}{1 - \| T_J \|}}$$

ERROR DE TRUNCAMIENTO: Si $\| T_J \| < 1$ el método de Jacobi converge $\forall x^{(0)}$ y:

"Jacobi"

$$\boxed{\| x^{(k)} - x \| \leq \frac{\| T_J \|}{1 - \| T_J \|} \| x^{(k)} - x^{(k-1)} \|}$$

☒ MÉTODO DE GAUSS-SEIDEL:

↳ Parecido al de Jacobi, busca acelerar la convergencia.

↳ Despejamos cada x_i de E_i ($a_{ii} \neq 0$), pero cada vez que despejamos un x_i, el resto de $x_0, \dots, x_{i-1}$ usamos los que acabamos de calcular.

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

$i-1$ primeras incógnitas ya actualizadas en el paso iterativo k.

$n-i-1$ incógnitas sin actualizar, del paso iterativo $k-1$ (Jacobi).

Condición de convergencia:

↳ Idem Jacobi ($T_J \leftrightarrow T_{GS}$). Vale la condición y la propiedad.

↳ La convergencia de uno no implica la del otro, como regla general.

Errores: Idem Jacobi ($T_J \leftrightarrow T_{GS}$)

Hallar $x^{(k)}$ de forma matricial:

U es triangular superior con los valores de la iteración anterior (k-1).

L es triangular inferior con los valores de la iteración actual (k).

Successive Over-Relaxation

☒ MÉTODO SOR

↳ Se busca la reducción del error residual.

Si sacamos factor común de $x^{(k)}$ y luego sumamos y restamos $a_{ii} x_i^{(k-1)}$, podemos sacar de la expresión el error residual: $r^{(k)} = b - A\hat{x}^{(k-1)}$. Entonces tenemos que:

$$r_i^{(k)} = \begin{cases} b_i - \sum_{j=1}^{i=1} a_{ij} x_j^{(k)} - \sum_{j=1}^{n} a_{ij} x_j^{(k-1)} & \text{Gauss - Seidel} \\ b_i - \sum_{j=1}^{n} a_{ij} x_j^{(k-1)} & \text{Jacobi} \end{cases}$$

Entonces: $x_i^{(k)} = x_i^{(k-1)} + \frac{r_i^{(k)}}{a_{ii}} \Rightarrow x_i^{(k)} = x_i^{(k-1)} + \omega \frac{r_i^{(k)}}{a_{ii}} \Rightarrow$

Modificamos con un factor de peso o parámetro de relajación

En gral: $1 < \omega < 2$

$$\Rightarrow \boxed{x_i^{(k)} = (1-\omega)\, x_i^{(k-1)} + \omega \left[x_i^{(k)} \right]_{GS}} \quad ; \; i = 1, \dots, n \; ; \; k \geq 1$$

Hallar $x^{(k)}$ de forma matricial:

$x^{(k)} = T_\omega x^{(k-1)} + c_\omega$; $T_\omega = (D - \omega L)^{-1} \left[(1-\omega) D + \omega U \right]$

$c_\omega = \omega (D - \omega L)^{-1} b$

Condiciones de convergencia: $\boxed{\rho(T_\omega) \geq |\omega - 1|}$ si $a_{ii} \neq 0 \ \forall i$ $\Rightarrow$

$\Rightarrow 0 < \omega < 2$

- Si A es definida positiva y se cumple $0 < \omega < 2$, SOR converge $\forall x^{(0)}$.
- Si A es simétrica y definida positiva, SOR converge $\forall x^{(0)}$.
- Si A es definida positiva y tridiagonal:

$$\boxed{\omega_{\text{óptimo}} = \frac{2}{1 + \sqrt{1 - \rho(T_J)^2}}}$$

↳ $T_J = D^{-1}(L+U)$

Si A no cumple esto, veo cuantas iteraciones hay por cada ω posible y tomo el menor.
↳ Usar si A fijo, b variable y hay que calcular muchos sistemas $Ax = b$.

ECUACIONES NO LINEALES

Objetivo: Hallar $x : F(x) = 0$ (raíces de F).

Función no lineal ↲ ↳ "cero" de F

TEOREMA: Si $F(x) \in \overset{\text{continua}}{C^0}$ en $[a_0 ; b_0]$ y $F(a_0).F(b_0) < 0 \Rightarrow$
$\Rightarrow F(\alpha_K) = 0$ para un número impar de valores $\alpha_K \in [a_0 ; b_0]$. (número impar de raíces α_K)

☒ MÉTODOS DE ARRANQUE:

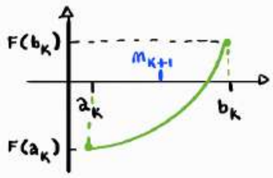
BISECCIÓN:

↳ Idea de intervalo "sandwich"

Generar una sucesión $I_K = [a_K ; b_K] : F(a_K).F(b_K) < 0 \wedge b_K - a_K \to 0$ con $K \to +\infty$

1. $\alpha \in I_K \ \forall K$
2. Converge (pq el intervalo se va achicando).

Algoritmo: construcción de I_K.



1. Punto medio: $m_{K+1} = \frac{a_K + b_K}{2} = \hat{\alpha}$

nueva aproximación

2. Suponer: $F(m_{K+1}) \neq 0$ (caso contrario $m_{K+1} = \alpha$)

3. Definir $I_{K+1} = [a_{K+1} ; b_{K+1}] = \begin{cases} [m_{K+1} ; b_K] & \text{si } F(m_{K+1}).F(a_K) > 0 \text{ (raíz en la derecha)} \\ [a_K ; m_{K+1}] & \text{si } F(m_{K+1}).F(a_K) < 0 \text{ (raíz en la izquierda)} \end{cases}$

↳ En el paso sig. achicamos y nos quedamos con la mitad que contiene la raíz.

Propiedades:

- Por construcción $F(a_K).F(b_K) < 0 \ \forall K$
- Luego de n pasos $\alpha \in I_n$ de longitud $\frac{b_0 - a_0}{2^n}$ ↪ proviene de haber partido a la mitad n veces el intervalo original.

Errores:

$\alpha = m_{k+1} \pm \Delta\alpha_{k+1}$

- Error absoluto: $\delta\alpha = \alpha - m_{k+1}$
- Cota del error absoluto: $\Delta\alpha_{k+1} = \frac{b_k - a_k}{2} = \frac{b_0 - a_0}{2^{k+1}}$ ↳ cant. iteraciones.
- Cota del error de truncamiento: $e_k = m_{k+1} - m_k$ (estimador)

Convergencia:

Como $\Delta\alpha_{k+1} \to 0$ si $k \to \infty$ $\Rightarrow$ $m_{k+1} \to \alpha$ si $k \to \infty$ $\Rightarrow$

$\Rightarrow$ el método es incondicionalmente convergente (en el mundo ideal).

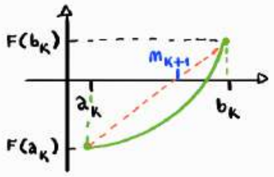
En el mundo real el método puede no converger, porque hay errores de redondeo que pueden cambiar signos.

REGULA-FALSI:

↳ Busca acelerar el proceso iterativo.

↳ Igual a bisección pero cambia la forma de calcular el estimador m_{k+1}.

$$m_{k+1} = a_k - (b_k - a_k)\frac{F(a_k)}{F(b_k) - F(a_k)}$$



Errores:

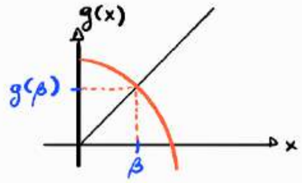
- Cota del error absoluto: Si F' es continua y $F' \neq 0 \Rightarrow |m_{k+1} - \alpha| \leq \underbrace{|m_{k+1} - m_k|}_{e_k}$

↳ cota del error de truncamiento.

☒ METODOS ITERATIVOS

PUNTO FIJO:

↳ Busca acelerar la convergencia.

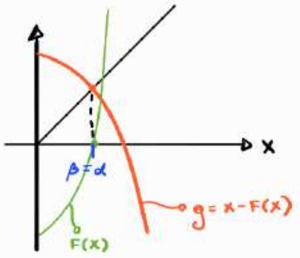


Objetivo: Hallar $x : g(x) = x$ (problema no lineal).

Punto fijo de g: valor β que satisface que $g(x) = x$.

Resolver $F(x) = 0$ es lo mismo que resolver $g(x) = x - q F(x)$; $q = \text{cte} \neq 0$

$\beta = \alpha$ es punto fijo de $g(x)$ $\Leftrightarrow$ β es raíz de $F(x)$.



Condiciones para la existencia y unicidad del punto fijo:

1. Si $g \in C[a;b]$ y $g(x) \in [a;b]$ $\forall x \in [a;b]$, entonces g tiene un punto fijo en $[a;b]$.
2. Si, además, $\exists\, g'(x)$ en $(a;b)$ y $\exists\, 0 < K < 1$ tal que $|g'(x)| \leq K < 1$ $\forall x \in (a;b)$, entonces el punto fijo es único.

Búsqueda de la solución:

1. Escoger una aproximación inicial x_0 → semilla.
2. Generar la sucesión $x_{K+1} = g(x_K)$; $K \geq 0$
   ↳ Función de iteración.
3. Hipótesis de convergencia: $\lim_{K \to \infty} x_{K+1} = \beta$ → punto fijo.
4. Seleccionar el óptimo → Comparando cotas de error de truncamiento entre nuestras opciones.

Convergencia: Teorema del punto fijo

Sea $g \in C[a;b]$ tal que $g(x) \in [a;b]\ \forall x \in [a;b]$. Si $\exists g'(x)$ en $(a;b)$ y $\exists\ 0 < K < 1$ tal que $|g'(x)| \leq K < 1\ \forall x \in (a;b)$, entonces $\forall x_0 \in (a;b)$ la sucesión $x_{K+1} = g(x_K)$ $K \geq 0$ converge al único punto fijo β en $[a;b]$.
↳ Que no se cumplan las cond. no implica que no converge.

Errores:

- Cota del error absoluto: $|x_n - \beta| \leq K^n \cdot \max(x_0 - a;\ b - x_0)$
  cte. de Lipschitz
  ↳ n-ésimo paso.

  Obs: K "grande" $K \to 1$ ⇒ lento
  K "chico" $K \to 0$ ⇒ rápido.

- Cota del error de truncamiento: $|x_{n+1} - \beta| \leq \dfrac{K}{1-K} |x_{n+1} - x_n|$

  Obs: Vale lo mismo sobre K "grande" y K "chico", pero además:
  si $K < \dfrac{1}{2}$ ⇒ podemos tomar $\dfrac{K}{1-K} \cong 0$.

NEWTON - RAPHSON:

↳ Caso particular de punto fijo donde $g_{NR}(x) = x - \dfrac{F(x)}{F'(x)}$

$$x_{K+1} = x_K - \frac{F(x_K)}{F'(x_K)}$$

Convergencia:

↳ Debe verificar el TPF.

1. $g_{NR}(x) \in [a;b]\ \forall x \in [a;b]$
2. $F'(x) \neq 0\ \forall x \in [a;b]$: $\exists g_{NR}(x)$.
3. $\exists F''(x)$: $\exists g'_{NR}(x)$.
4. $|g'_{NR}(x)| = \left| \dfrac{F \cdot F''}{(F')^2} \right| < 1\ \forall x \in [a;b]$

MÉTODO DE LA SECANTE:

↳ NR requiere conocer $F'(x)$ y evaluar $F'(x_K)$ $\forall K$ ⇒ planteamos alternativa para calcular F':

$$F'(x) = \lim_{\varepsilon \to 0} \frac{F(x+\varepsilon) - F(x)}{\varepsilon} \Rightarrow F'(x) \cong \frac{F(x+\varepsilon) - F(x)}{\varepsilon} \Rightarrow$$

$$\Rightarrow F'(x_K) \cong \frac{F(x_K) - F(x_{K-1})}{x_K - x_{K-1}}$$

$x_{k+1} = x_k - (x_k - x_{k-1})\left(\dfrac{F(x_k)}{F(x_k) - F(x_{k-1})}\right)$ → Para "arrancar" el método no basta con una semilla x_0. Necesitamos x_0 y x_1. Para hallarlos usamos métodos de arranque.

Convergencia:
↳ No converge incondicionalmente.

☒ ORDEN DE CONVERGENCIA:

P = orden de convergencia (tasa de reducción del error).
C = constante asintótica del error.

$$\text{Si } \exists P \neq 0 \text{ y } C \neq 0 \Rightarrow \lim_{k\to\infty} \frac{\overset{\text{error absoluto}}{e_{k+1}}}{e_k^P} = C$$

$$\lim_{k\to\infty} \frac{|e_k|}{|e_{k-1}|^P} = \lambda \underset{\substack{\downarrow \\ \text{Tomando} \\ k \text{ muy grande.}}}{\Rightarrow} \frac{|e_{k+1}|}{|e_k|^P} \cong \frac{|e_k|}{|e_{k-1}|^P} \Rightarrow P = \frac{\ln\left(\frac{e_{k+1}}{e_k}\right)}{\ln\left(\frac{e_k}{e_{k-1}}\right)}$$

; usamos $e_k \cong \Delta_k$
Cota del error absoluto en la iteración k.

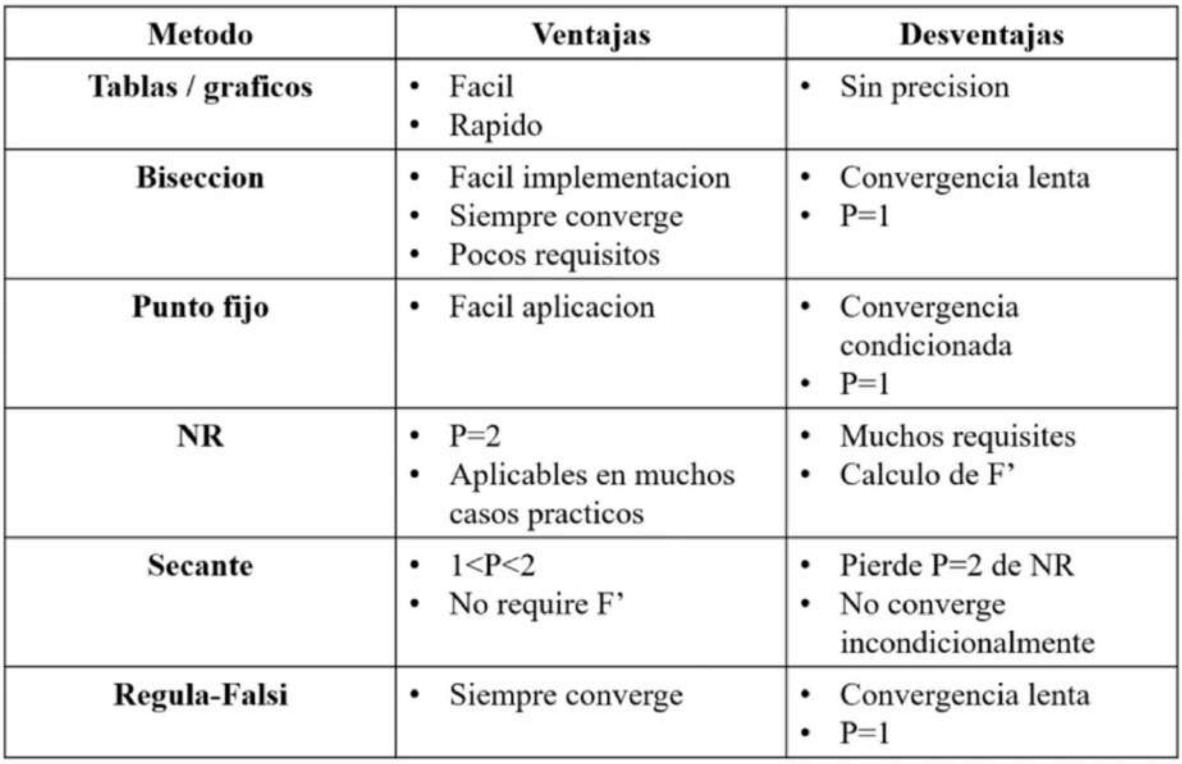
En Bisección y Regula-Falsi: $P = 1$.

$P = 1,\ C = g'(\beta)$.

En punto fijo: si $g'(\beta) \neq 0 \Rightarrow P = 1$ → orden de convergencia lineal.
si $g'(\beta) = 0 \Rightarrow P = 2$ → orden de convergencia cuadrático.

En Newton-Raphson: $P = 2$ (pues $g'_{NR}(\beta) = 0$).

En método de la secante: $1 < P < 2$ → método superlineal.



Metodo	Ventajas	Desventajas
Tablas / graficos	• Facil • Rapido	• Sin precision
Biseccion	• Facil implementacion • Siempre converge • Pocos requisitos	• Convergencia lenta • P=1
Punto fijo	• Facil aplicacion	• Convergencia condicionada • P=1
NR	• P=2 • Aplicables en muchos casos practicos	• Muchos requisites • Calculo de F’
Secante	• 1<P<2 • No require F’	• Pierde P=2 de NR • No converge incondicionalmente
Regula-Falsi	• Siempre converge	• Convergencia lenta • P=1

SISTEMA DE ECUACIONES NO LINEALES:

Problema tipo:

Resolver el SENL definido por las n ecuaciones E_i:

$$E_1 = F_1(x_1; x_2; \dots; x_n) = 0$$
$$E_2 = F_2(x_1; x_2; \dots; x_n) = 0$$
$$\dots$$
$$E_n = F_n(x_1; x_2; \dots; x_n) = 0$$

donde F_i son funciones no lineales y $\mathbb{F}(\mathbb{X}) = 0$, con: $\mathbb{F} = (F_1; F_2; \dots; F_n)^t$

$$\mathbb{X} = (x_1; x_2; \dots; x_n)^t$$

☒ METODO DE PUNTO FIJO → con/sin modificación de Gauss-Seidel.

$\mathbb{F}(\mathbb{X}) = 0$ es equivalente a $G(B) = B$, con B punto fijo de G, si se define:

$$G(\mathbb{X}) = \mathbb{X} - \mathbb{F}(\mathbb{X}), \text{ siendo } G = (g_1; g_2; \dots; g_n)^t$$

Obs: $A = B$, con A raíz de $\mathbb{F}$ y B punto fijo de G.

Convergencia: Teorema del Punto Fijo generalizado.

Sea $D = \{x : a_i \leq x_i \leq b_i \ \forall i = 1; \dots; n\}$. Si G es una función continua en $D \subseteq \mathbb{R}^n \rightarrow \mathbb{R}^n$ tal que: $G(\mathbb{X}) \in D \ \forall \mathbb{X} \in D$, entonces G tiene un punto fijo en D.

Si, además, existe $K < 1$ tal que $\left| \frac{\partial g_i}{\partial x_i} \right| \leq \frac{K}{n} \ \forall \mathbb{X} \in D$ y $\forall i, j = 1, \dots, n$, entonces la sucesión $\mathbb{X}_{K+1} = G(\mathbb{X}_K)$ con $K \geq 0$ converge al único punto fijo $B \in D \ \forall \mathbb{X}_0 \in D$.

Errores:

- Cota del error de truncamiento: $\|\mathbb{X}_n - B\|_\infty \leq \frac{K^n}{1-K} \|\mathbb{X}_1 - \mathbb{X}_0\|_\infty$

Obs: K "grande" $K \rightarrow 1$ ⇒ lento
K "chico" $K \rightarrow 0$ ⇒ rápido.

EJEMPLO:

$$\begin{cases} x_1 - \frac{1}{3}\cos(x_2) = \frac{1}{3} \\ \frac{1}{20} e^{-x_1} + x_2 = -\frac{1}{2} \end{cases} \Rightarrow \mathbb{F} = \begin{pmatrix} F_1(x_1; x_2) \\ F_2(x_1; x_2) \end{pmatrix} = \begin{pmatrix} x_1 - \frac{1}{3}\cos(x_2) - \frac{1}{3} \\ \frac{1}{20} e^{-x_1} + x_2 + \frac{1}{2} \end{pmatrix} = 0$$

$$G(\mathbb{X}) = \mathbb{X} - \mathbb{F}(\mathbb{X}) \Rightarrow G = \begin{pmatrix} g_1(x_1; x_2) \\ g_2(x_1; x_2) \end{pmatrix} = \begin{pmatrix} \frac{1}{3}\cos(x_2) + \frac{1}{3} \\ -\frac{1}{20} e^{-x_1} - \frac{1}{2} \end{pmatrix}$$

$$D = [-1; 1] \times [-1; 1]$$

TPF: $$\left.\begin{array}{l} |g_1(x_2)| \leq \frac{1}{3}|\cos(x_2)| + \frac{1}{3} \leq \frac{2}{3} < 1 \ ✓ \\ |g_2(x_1)| \leq \frac{1}{20}|e^{-x_1}| + \frac{1}{2} \leq \frac{e}{20} + 0,5 \leq 0,64 < 1 \ ✓ \end{array}\right\} \Rightarrow$$

$\Rightarrow G(\mathbf{x}) \in D \ \forall \ \mathbf{x} = (x_1; x_2)^t \in D$ y G tiene un punto fijo en D.

$\frac{\partial g_1}{\partial x_1} = \frac{\partial g_2}{\partial x_2} = 0$; $\frac{\partial g_1}{\partial x_2} = -\frac{1}{3}\operatorname{sen}(x_2) \Rightarrow \left|\frac{\partial g_1}{x_2}\right| \leq \frac{1}{3}$

$\frac{\partial g_2}{\partial x_1} = \frac{1}{20}e^{-x_1} \Rightarrow \left|\frac{\partial g_2}{\partial x_1}\right| \leq \frac{e}{20}$

tomando $K = 0,8 \Rightarrow \left|\frac{\partial g_i}{\partial x_j}\right| \leq \frac{K}{n} = \frac{0,8}{2} = 0,4 \ \forall i,j = 1;2 \Rightarrow$

$\Rightarrow$ la sucesión $\underbrace{\mathbf{x}_{K+1} = G(\mathbf{x}_K)}_{\begin{array}{l} x_1^{K+1} = \frac{1}{3}\cos(x_2^K) + \frac{1}{3} \\ x_2^{K+1} = -\frac{1}{20}e^{-x_1^K} - \frac{1}{2} \end{array}}$ converge a un único punto fijo en D.

Aplicando modificación de Gauss-Seidel: $x_1^{K+1} = \frac{1}{3}\cos(x_2^K) + \frac{1}{3}$

$x_2^{K+1} = -\frac{1}{20}e^{-x_1^{K+1}} - \frac{1}{2}$

☒ MÉTODO DE NEWTON:

$G(\mathbf{x}) = \mathbf{x} - J(\mathbf{x})^{-1} \cdot \mathbb{F}(\mathbf{x})$, con $J_{ij}(\mathbf{x}) = \frac{\partial F_i(\mathbf{x})}{\partial x_j}$.

$x^{K+1} = G(\mathbf{x}^K) = \mathbf{x}^K \underbrace{- J(\mathbf{x}^K)^{-1} \mathbb{F}(\mathbf{x}^K)}_{\mathbf{y}^K}$ $\Rightarrow J(\mathbf{x}^K) \cdot \mathbf{y}^K = -\mathbb{F}(\mathbf{x}^K) \rightarrow$ SEL.

Convergencia: cuadrática.

☒ MÉTODO CUASI-NEWTON:

↳ Para evitar el cálculo de J en casos reales (n grande), se utiliza el mismo concepto de la secante: aproximamos las derivadas.

$\frac{\partial F_i(\mathbf{x}^K)}{\partial x_j} \cong \frac{F_i(\mathbf{x}^K + h \cdot E_j) - F_i(\mathbf{x}^K)}{h}$; $E_j = (0; \dots; 0; 1; 0; \dots; 0)$

↳ j-ésima posición.

DIFERENCIACIÓN:

Problema tipo: hallar una aproximación para la derivada: $f^{(n)} = \frac{d^n f}{dx^n}$.

Solución: derivada numérica $f^{(n)} \cong \sum_i \underset{\substack{\downarrow \\ \text{factor de peso.}}}{w_i} \cdot f(\underset{\substack{\downarrow \\ \text{puntos de evaluación.}}}{x_i})$

☒ APROXIMACIONES PARA LA DERIVADA PRIMERA:

FÓRMULA DE 2 PUNTOS:

$x_0 \in [a;b]$ y $f(x) \in C^2[a;b]$, con $x_1 = x_0 + h$, con h suficientemente pequeño tal que $x_1 \in [a;b]$. Entonces:

$$\boxed{f'(x_0) \cong \frac{f(x_0+h) - f(x_0)}{h}}$$

; si $h > 0 \Rightarrow$ aproximación en adelanto.
si $h < 0 \Rightarrow$ aproximación en atraso.

Errores:

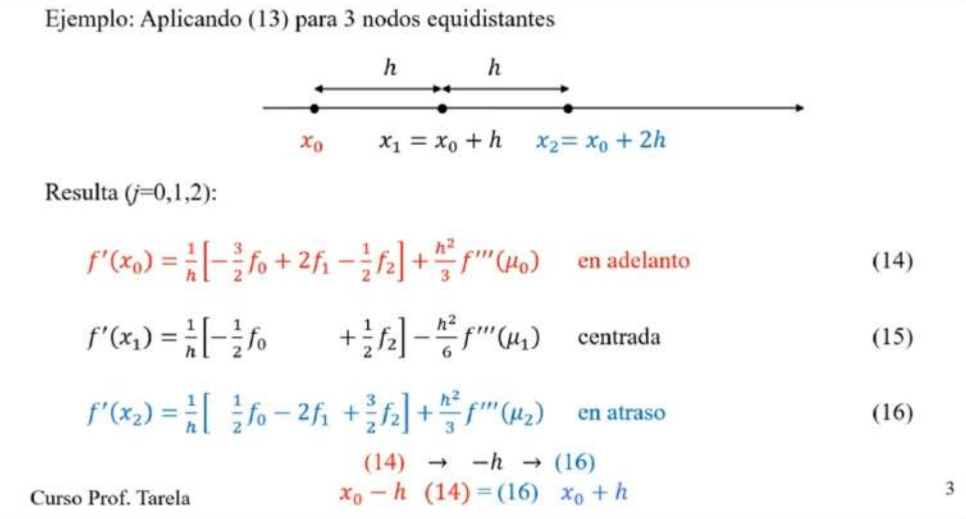
- Cota del error de truncamiento: $\frac{M}{2} \cdot h$, donde $M: |f''(x)| \leq M \quad \forall x \in [a;b]$

FÓRMULA DE N+1 PUNTOS:

$x = x_j$ para algún $0 \leq j \leq N$:

$$f'(x_j) = \underbrace{\sum_{k=0}^{N} f(x_k) \cdot L'_{Nk}(x_j)}_{\text{Fórmula N+1 puntos}} + \underbrace{\frac{f^{(N+1)}(\gamma(x_j))}{(N+1)!} \prod_{\substack{k=0 \\ k \neq j}}^{N} (x_j - x_k)}_{\text{Error de truncamiento.}}$$

EJEMPLO:



Ejemplo: Aplicando (13) para 3 nodos equidistantes

Resulta (j=0,1,2):

$$f'(x_0) = \frac{1}{h}\left[-\frac{3}{2}f_0 + 2f_1 - \frac{1}{2}f_2\right] + \frac{h^2}{3}f'''(\mu_0) \quad \text{en adelanto} \quad (14)$$

$$f'(x_1) = \frac{1}{h}\left[-\frac{1}{2}f_0 \qquad + \frac{1}{2}f_2\right] - \frac{h^2}{6}f'''(\mu_1) \quad \text{centrada} \quad (15)$$

$$f'(x_2) = \frac{1}{h}\left[\ \frac{1}{2}f_0 - 2f_1 + \frac{3}{2}f_2\right] + \frac{h^2}{3}f'''(\mu_2) \quad \text{en atraso} \quad (16)$$

$(14) \rightarrow -h \rightarrow (16)$

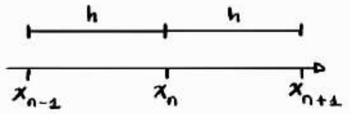
$x_0 - h \quad (14) = (16) \quad x_0 + h$

Curso Prof. Tarela 3

☒ MÉTODO DE LOS COEFICIENTES INDETERMINADOS:

↳ Consiste en desarrollar $f(x)$ en serie de Taylor alrededor del punto de la grilla deseado, y solicitar que cumpla con una forma de N+1 puntos.

EJEMPLO: derivada primera con 3 nodos.



$x_n - x_{n-1} = x_{n+1} - x_n = h$ (son equidistantes)

① $f_n' \cong a f_{n-1} + b \cdot f_n + c f_{n+1}$

② Desarrollando f en serie de Taylor, en los puntos en los cuales queremos la aproximación, y alrededor del punto en el cual vamos a especializar:

$$f_{n-1} = f_n - h f_n' + h^2 f_n'' - \frac{h^3}{6} f_n''' + O(h^4)$$

$$f_{n+1} = f_n + h f_n' + h^2 f_n'' + \frac{h^3}{6} f_n''' + O(h^4)$$

③ Despejo a, b, c:

$$f_n' + R_D = \underbrace{(a+b+c)}_{0} f_n + h \underbrace{(c-a)}_{1} f_n' + \underbrace{\frac{h^2}{2} (a+c)}_{0} f_n'' + \overbrace{\frac{h^3}{6} (c-a) f_n''' + O(h^4)}^{\text{Error de truncamiento}}$$

↳ para que se cumpla la igualdad.

Resulta SEL: $\begin{pmatrix} 1 & 1 & 1 \\ -h & 0 & h \\ h^2/2 & 0 & h^2/2 \end{pmatrix} \begin{pmatrix} a \\ b \\ c \end{pmatrix} = \begin{pmatrix} 0 \\ 1 \\ 0 \end{pmatrix} \Rightarrow$

$$\Rightarrow a = -\frac{1}{2h} \;;\; b = 0 \;;\; c = \frac{1}{2h}$$

Aproximación de la derivada: $f_n' \cong \frac{f_{n+1} - f_{n-1}}{2h} = \hat{f}_n'$

Cota del error de truncamiento → Como el término del error tiene infinitos términos, acotamos al máximo valor que puede tomar la expresión:

$$r_D \le \left| \frac{h^3}{6} \cdot (c-a) \cdot \max |f'''(\mu)| \right| \;;\quad x_0 \le \mu \le x_2 \quad \Rightarrow$$

$$\Rightarrow R_D = \frac{h^3}{6} f'''(\mu) \Rightarrow f_n' = \hat{f}_n' + R_D$$

Regla práctica: determinación del orden de precisión.

$$\underset{\text{num. de nodos}}{N} = \underset{\text{orden derivación}}{OD} + \underset{\text{orden precisión}}{OP}$$

Excepción: Si la aproximación es centrada y la paridad de N es distinta de la paridad del OD, entonces se gana un orden de precisión: OP = N − OD + 1.

PROBLEMAS DE VALORES INICIALES

Método numérico: $P^0 = P_0$; $P^{n+1} = a \cdot \Delta t + (1 - b \cdot \Delta t) P^n$; $n \geq 0$; P_{eq}

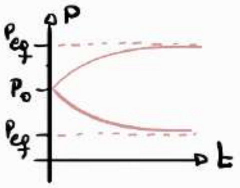
estado temporal

↳ Concentración inicial.

↳ Tiene que ser chico.

↳ Concentración en equilibrio.

P^{n+1} debe tender a P_{eq} {



¿Cómo determinar Δt?

☒ PROBLEMA DIFERENCIAL

Parámetro sobre el cual el sist cambia ("tiempo")

Ley de evolución

$$\frac{dy}{dt} = y' = \underbrace{f(t;y)}_{\text{fuente}} \quad ; \quad \underbrace{t \geq 0}_{\substack{\text{sentido de} \\ \text{flujo de la info.} \\ \text{(en 1 dirección)}}}$$ con $y(t_0) = y_0$ condición inicial. (estado de partida)

Buscamos $y(t)$ que satisfaga la ecuación y la condición inicial.

↓ Característica o prop. del sist.

Existencia y unicidad de la solución:

$D = \{(t;y) : t \geq 0, -\infty < y < \infty\}$

1. Si $f(t;y)$ es continua en D, entonces existe la solución $y(t)$.
2. Si $\frac{\partial f}{\partial y}$ es continua en D, entonces la solución es única.

Condición de estabilidad del problema:

Considerando el problema perturbado para $z(t)$:

$$\frac{dz}{dt} = f(t;z) + \underset{\text{perturbación}}{\delta(t)} \quad ; \quad \text{con } t \geq 0,$$

$z(t=0) = y_0 + \varepsilon_0$ perturbación

$|\varepsilon_0| < \varepsilon$

$|\delta(t)| < \varepsilon$

El problema es estable ("bien planteado") si $\exists K > 0$ tal que: $|z(t) - y(t)| < K\varepsilon$

Tipos de problemas:

a. 1 ecuación de orden 1

b. n ecuaciones de orden 1

c. 1 ecuación de orden n (con cambio de variable se puede transf. en b).

☒ PROCESO DE DISCRETIZACIÓN:

↳ En el mundo ideal $y(t)$ es continuo, pero en el mundo real es una sucesión de estados temporales discretos ⇒ para pasar al mundo real discretizamos (genera un error de truncamiento).

$t \rightarrow t^n$ (instantes discretos)

$y \rightarrow u^n \cong y(t^n)$ (variable discreta)

Pasos:

Sea $a \leq t \leq b$: $t^0 = a \wedge t^N = b$, entonces:

1. Discretización del "tiempo": $t \rightarrow t^n$; $n = 0, 1, 2, \ldots$

"Paso del tiempo": $\boxed{h = t^{n+1} - t^n}$

h, k, Δt

Si $h = cte. \Rightarrow \boxed{t^n = a + nh}$, con $\boxed{h = \frac{b-a}{N}}$

En gral: $t^n - t^{n-1} \neq t^{n+1} - t^n$.

2. Discretización de la EDO $\frac{dy}{dt} = f(t; y)$:

   1. $u^n \cong y(t^n)$

   2. $\frac{dy}{dt} \rightarrow \frac{u^{n+1} - u^n}{h}$

   3. $f(t; y) \rightarrow f(t^n; y^n) = f^n$ (fuente en el estado temporal n)

3. Discretización de la condición inicial (CI) $y(a) = \alpha$:

$$t^0 = a \Rightarrow u^0 = y(t^0) = y(a) \Rightarrow \boxed{u^0 = \alpha}$$

▨ **MÉTODO DE EULER** ⟶ método explícito ⟶ calcula u^{n+1} a partir de valores conocidos.

orden 1

h_{max} (no es incond. estable).

$\boxed{u^{n+1} = u^n + h f(t^n; u^n)}$; $n = 0, 1, 2, \ldots$

$\boxed{u^0 = \alpha}$

Vamos avanzando y obteniendo la sol. numérica hasta $t^N = b$.

Entonces: $\frac{dy}{dt} \longrightarrow \frac{u^{n+1} - u^n}{h} = f^n$

Error global: $E^n = u^n - y(t^n)$

$$\left(\frac{h_2}{h_1}\right)^{p} = \frac{E_2}{E_1}$$

p ⟶ orden del método.

⟶ Comparación del error entre 2 pasos (en función de h).

Error de discretización de la EDO.

Con quien aproximamos $\frac{dy}{dt}$ (en Euler sería f^{n+1})

Error de truncamiento: $e^{n+1} = \{\text{Operador discreto}\} - f(t^n; y(t^n))$

↳ Para el Método de Euler:

Si existe M tal que $|y''| \leq M \Rightarrow \boxed{e^{n+1} \sim \frac{h}{2} M}$

Error de redondeo:

Si δ da cuenta del error de redondeo y $|y''| \leq M$, el error de redondeo puede ser despreciable para: $h_{min} \cong \sqrt{\frac{2\delta}{M}}$.

Consistencia: un método de discretización es consistente si el error de truncamiento tiende a cero para el paso de discretización tendiendo a cero: $\lim_{h \to 0} e^{n+1} = 0$.

Estabilidad:

$$\left.\begin{array}{l} u^n \to u^n + \overset{\text{perturbación}}{\delta u^n} \\ u^{n+1} \to u^{n+1} + \delta u^{n+1} \end{array}\right\} \Rightarrow \text{Factor de amplificación: } \boxed{g^n \equiv \frac{\delta u^{n+1}}{\delta u^n}}$$

$\lvert g^n \rvert < 1$	estable
$\lvert g^n \rvert = 1$	marginalmente estable
$\lvert g^n \rvert > 1$	inestable (se amplificó la perturbación entre iteraciones)

Generalmente para hallar h pido que $|g^n| < 1$. El intervalo resultante no quiere decir que la sol. sea buena para todo el intervalo.

Obs: Si h resultó mayor a 0 en el intervalo ⇒ método incondicionalmente estable.

Convergencia: $\lim\limits_{\substack{h \to 0 \\ t^n \text{ fijo}}} u^n = y(t^n)$ ⟶ Difícil de demostrar.

Teorema de Lax: consistente + estable ⇒ convergente.

☒ MÉTODO FUERTEMENTE IMPLÍCITO (EULER INVERSO)

⟶ cálculo iterativo.

⟶ incond. estable (aunque pueden haber zonas de inestabilidad, por lo que h debe ser mayor a un h_{min}).

↳ Evaluar la fuente en adelanto:

$$\boxed{u^{n+1} \equiv u^n + h f(t^{n+1}; u^{n+1})}$$

Entonces: $\frac{dy}{dt} \longrightarrow \frac{u^{n+1} - u^n}{h} = f^{n+1}$

Estabilidad: $\delta u^{n+1} = \delta u^n - h \delta u^{n+1} \Rightarrow g^n \equiv \frac{\delta u^{n+1}}{\delta u^n} = \frac{1}{1+h} \Rightarrow$

⇒ $h > 0$ para que el método sea estable ⟶ No hay h_{max} ⇒ método incondicionalmente estable.

☒ MÉTODO IMPLÍCITO PONDERADO

⟶ orden 1

$$\boxed{u^{n+1} \equiv u^n + h\left[\beta f(t^{n+1}; u^{n+1}) + (1-\beta) f(t^n; u^n)\right]} \;; \text{ con } 0 \leq \overset{\text{parámetro de ponderación}}{\beta} \leq 1$$

Entonces: $\frac{dy}{dt} \longrightarrow \frac{u^{n+1} - u^n}{h} = \beta f^{n+1} + (1-\beta) \cdot f^n$

Si $\beta = 0$ ⇒ Euler (explícito)
Si $\beta = 1$ ⇒ Euler inverso (implícito)
Si $\beta = \frac{1}{2}$ ⇒ Crank-Nicholson (implícito).

Estabilidad: Incond. estable si $\beta \geq 1/2$

☒ MÉTODO DE CRANK-NICHOLSON ⟶ orden 2

↳ Método implícito ponderado, pero para $\beta = \frac{1}{2}$

$$\boxed{u^{n+1} = u^n + \frac{h}{2}\left[f(t^{n+1}; u^{n+1}) + f(t^n; u^n) \right]} \quad \Rightarrow \frac{dy}{dt} \longrightarrow \frac{u^{n+1} - u^n}{h} = \frac{f^{n+1} + f^n}{2}$$

☒ MÉTODOS DE RUNGE-KUTTA ⟶ métodos explícito

PREDICTOR EXPLÍCITO (RK2) ⟶ orden 2

1. Predicción:

$$u^{n+\frac{1}{2}} = u^n + \frac{h}{2} f(t^n; u^n) \quad ; \quad u^{n+\frac{1}{2}} \longrightarrow \text{variable intermedia (no es parte de la sol.)}$$

2. Corrección:

$$\boxed{u^{n+1} = u^n + h f(t^{n+\frac{1}{2}}; u^{n+\frac{1}{2}})}$$

RK4: ⟶ orden 4

1. Predicción:

$$u_*^{n+\frac{1}{2}} = u^n + \frac{h}{2} f(t^n; u^n)$$

$$u_{**}^{n+\frac{1}{2}} = u^n + \frac{h}{2} f(t^{n+\frac{1}{2}}; u_*^{n+\frac{1}{2}})$$

$$u_*^{n+1} = u^n + h f(t^{n+\frac{1}{2}}; u_{**}^{n+\frac{1}{2}})$$

2. Corrección:

$$\boxed{u^{n+1} = u^n + \frac{h}{6}\left[f(t^n; u^n) + 2f(t^{n+\frac{1}{2}}; u_*^{n+\frac{1}{2}}) + 2f(t^{n+\frac{1}{2}}; u_{**}^{n+\frac{1}{2}}) + f(t^{n+1}; u_*^{n+1}) \right]}$$

SISTEMAS DE ECUACIONES DIFERENCIALES:

Si el problema es lineal $\Rightarrow \frac{dy}{dt} = Ay$, con: $y^t = (y_1 \; y_2 \; \cdots \; y_n)$

Estabilidad: el problema es estable si: $\mathrm{Re}(\underset{\substack{\downarrow \\ \text{autovalores de A}}}{\rho_j}) < 0$

☒ MÉTODOS DE DISCRETIZACIÓN

↳ Para cada incógnita del sistema, se utiliza una variable discreta y se aplican los métodos para el caso de 1 ecuación de orden 1: $y_i(t^n) \cong u_i^n$.

Estabilidad:

El problema es estable si, realizando el análisis de perturbaciones, la matriz de amplificación cumple la siguiente condición para sus autovalores σ_j: $|\sigma_j| < 1 \;\; \forall j$.

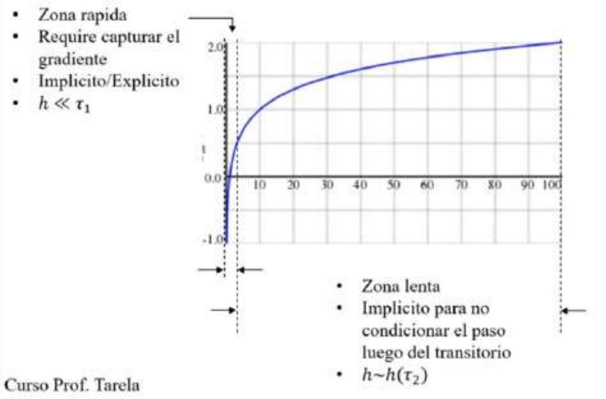
TIPOS DE PROBLEMAS DE VALORES INICIALES:

☒ ECUACIONES RÍGIDAS

↳ Problemas donde coexisten escalas de tiempo de evolución muy diferentes: $\tau_1 << \tau_2$

Sistemas rígidos de ecuaciones diferenciales:

- Soluciones del tipo $e^{\alpha t}$, con $\alpha \in C$ tal que $Re(\alpha) < 0$
- Problemas al resolverlas numéricamente
- Caso típico: $c_1 y'' + c_2 y' + c_3 y = f(t; y; y')$; con $|c_2| >> |c_1|$



- Zona rapida
- Require capturar el gradiente
- Implicito/Explicito
- $h \ll \tau_1$

- Zona lenta
- Implicito para no condicionar el paso luego del transitorio
- $h \sim h(\tau_2)$

Curso Prof. Tarela

☒ PROBLEMAS DE VALORES INICIALES CONSERVATIVOS

↳ se buscan soluciones numéricas donde se quiere preservar la característica de conservación de una propiedad importante del sistema.

OSCILADOR ARMÓNICO:

PVI:
$$u'' + \omega^2 u = 0$$
$$u(t=0) = u_0$$
$$u'(t=0) = u'_0$$

Aplicando RK2 de Punto Medio:

1. Hacemos que las ecuaciones sean de orden 1 introduciendo el cv $v = u'$.

$$\begin{cases} v' + \omega^2 u = 0 \\ u' - v = 0 \end{cases} \quad ; \; v(0) = u'(0) = u'_0 \;\; \wedge \;\; u(0) = u_0$$

2. Aplicamos el método para las variables discretas $u(t^n) \sim u^n$ y $v(t^n) \sim v^n$:

I. Predicción:

$$\begin{cases} v^{n+1/2} = v^n - h/2\, \omega^2 u^n \\ u^{n+1/2} = u^n + h/2\, v^n \end{cases}$$

II. Corrección:

$$\begin{cases} v^{n+1} = v^n - h\omega^2 u^{n+1/2} \\ u^{n+1} = u^n + h v^{n+1/2} \end{cases}$$

3. $p \equiv \dfrac{h^2\omega^2}{2}$ $\Rightarrow$
$$\begin{cases} v^{n+1} = v^n - h\omega^2(u^n + h/2\, v^n) = -h\omega^2 u^n + (1-p)v^n \\ u^{n+1} = u^n + h(v^n - h/2\, \omega^2 u^n) = (1-p)u^n + h v^n \end{cases} \Rightarrow$$

$$\Rightarrow \begin{vmatrix} u^{n+1} \\ v^{n+1} \end{vmatrix} = \begin{vmatrix} 1-p & h \\ -h\omega^2 & 1-p \end{vmatrix} \begin{vmatrix} u^n \\ v^n \end{vmatrix}$$

4. Como la sol. gral. del PVI es de la forma $u(t) \sim e^{\mu t}$ con $\mu \in C$, se propone:

$$u^n \sim e^{\mu t^n} = e^{\mu (nh)} = (e^{\mu h})^n = \gamma^n \quad ; \text{ con } \gamma \in C.$$

5. Utilizando $u^n = A\gamma^n$, como u' tiene el mismo modo de oscilación, se toma $v^n = B\gamma^n$:

$$\begin{vmatrix} A\gamma^{n+1} \\ B\gamma^{n+1} \end{vmatrix} = \begin{vmatrix} 1-p & h \\ -h\omega^2 & 1-p \end{vmatrix} \begin{vmatrix} A\gamma^n \\ B\gamma^n \end{vmatrix}$$

Como $\gamma^n \neq 0 \Rightarrow \begin{vmatrix} A\gamma \\ B\gamma \end{vmatrix} = \begin{vmatrix} 1-p & h \\ -h\omega^2 & 1-p \end{vmatrix} \begin{vmatrix} A \\ B \end{vmatrix} \Leftrightarrow$

$$\begin{vmatrix} 1-p-\gamma & h \\ -h\omega^2 & 1-p-\gamma \end{vmatrix} \begin{vmatrix} A \\ B \end{vmatrix} = M \begin{vmatrix} A \\ B \end{vmatrix} = \begin{vmatrix} 0 \\ 0 \end{vmatrix}$$

Para que el sist. tenga sol. no trivial, pidiendo $\det(M) = 0$ surgen:

$$\gamma_{1,2} = (1-p) \pm i\sqrt{2p}$$

Luego: $\gamma \in C \Rightarrow$ oscila ✓

$$|\gamma|^2 = \gamma_1 \gamma_2 = 1 + p^2 > 1 \Rightarrow \gamma^n \text{ amplifica} \Rightarrow \text{no es conservativo.}$$

Aplicando método de Nystrom:

1. Discretizar la EDO a orden 2: $u'' \overset{\sim}{=} \dfrac{u^{n+1} - 2u^n + u^{n-1}}{h^2}$
2. Aplicando el método: $\dfrac{u^{n+1} - 2u^n + u^{n-1}}{h^2} + \omega^2 u^n = 0$
3. $p^2 \equiv h^2\omega^2 \Rightarrow u^{n+1} = (2-p^2)u^n - u^{n-1}$
4. Nuevamente proponemos $u^n \sim \gamma^n$ con $\gamma \in C$.
5. Como $\gamma \neq 0 \Rightarrow \gamma^2 - (2-p^2)\gamma + 1 = 0 \Rightarrow \gamma_{1,2} = (1 - p^2/2) \pm p\sqrt{p^2/4 - 1}$

Para que la solución sea oscilatoria, debe asegurarse $\gamma \in C \Rightarrow$

$$\Rightarrow \frac{p^2}{4} - 1 < 0 \Leftrightarrow p < 2 \Rightarrow h < \frac{2}{\omega} \Rightarrow$$

$$\Rightarrow \gamma_{1,2} = (1 - p^2/2) \pm ip\sqrt{1 - p^2/4} \in C \Rightarrow \text{oscila ✓}$$

$$|\gamma|^2 = \gamma_1 . \gamma_2 = \dots = 1 \Rightarrow \gamma^n \text{ solo rota, es conservativo ✓}$$

PROBLEMAS DE VALORES DE CONTORNO:

$$\frac{d^2y}{dx^2} = f(x;y;y') \quad \text{o} \quad y'' = f(x;y;y') \quad \text{para} \quad \underbrace{a \leq x \leq b}_{\text{Intervalo de flujo de la info.}} \quad \text{con} \quad \left.\begin{array}{l} y(x=a)=\alpha \\ y(x=b)=\beta \end{array}\right\} \text{cond. de contorno}$$

fuente

x: Parámetro sobre el cual el int. cambia ("espacio")

ley de distribución

Queremos encontrar la distribución: $y(x)$

TEOREMA: Existencia y unicidad de la solución.

Sea $f(x;y;y')$ continua en $D = \{(x;y;y') : a \leq x \leq b, -\infty < y < +\infty, -\infty < y' < \infty\}$ y $\frac{\partial f}{\partial y}, \frac{\partial f}{\partial y'}$ continuas en D. Si:

- $\frac{\partial f}{\partial y}(x,y,y') > 0 \ \forall (x,y,y') \in D$ y
- $\exists M = \text{cte} : \left|\frac{\partial f}{\partial y}\right| \leq M \ \forall (x,y,y') \in D$,

entonces el problema tiene solución única.

PROCESO DE DISCRETIZACIÓN:

1. Discretizar el "espacio":

$x \rightarrow x_i \quad ; \quad i = 0, 1, \dots, N$

$a \leq x_i \leq b \quad$ con $\quad x_0 = a, \quad x_N = b$

"Paso espacial": $\Delta x = x_{i+1} - x_i \quad ; \quad$ si $\Delta x = \text{cte.} \Rightarrow x_i = a + i\Delta x$, con $\Delta x = \frac{b-a}{N}$

$\Delta x, h, k$

2. Discretizar la EDO.

$y \rightarrow u_i \cong y(x_i)$

$y'' \rightarrow \frac{u_{i+1} - 2u_i + u_{i-1}}{\Delta x^2}$ centrada $O(2)$

$y' \rightarrow \frac{u_i - u_{i-1}}{\Delta x}$ en atraso $O(1)$

$y' \rightarrow \frac{u_{i+1} - u_i}{\Delta x}$ en adelanto $O(1)$

$y' \rightarrow \frac{u_{i+1} - u_{i-1}}{2\Delta x}$ centrada $O(2)$

3. Discretizar las cond. de contorno (cc).

Las cc pueden ser más generales:

$c_1 y'(a) + c_2 y(a) = \alpha \quad ; \quad d_1 y'(b) + d_2 y(b) = \beta$ → cc tipo Rieman

Resultan los siguientes casos particulares:

- $c_1 = d_1 = 0$ → CC tipo Dirichlet
- $c_2 = 0$ o $d_2 = 0$ → CC tipo Neuman

<u>Nota</u>: en caso de tener una derivada como cc, ejemplo $y'(x_0)$, hay que aproximarla (tal que OP=2) y con la aproximación y'_0 buscar y_0.

☒ MÉTODO DE DIFERENCIAS FINITAS:

1. Discretización
2. Reemplazar las aproximaciones en la ecuación del problema.
3. Reordenar la ecuación resultante despejando los u_i para formar la ecuación en diferencias finitas.
4. Para cada $0 \leq i \leq N$, con N calculado en 1, calculo los u_i.
5. Resolver el SEL / SENL

☒ MÉTODO DE UPWINDING:

Se tiene $\alpha u'' + \beta u' + u = 0$

Si $\alpha \ll \beta \Rightarrow$ efecto capa límite (variación muy brusca de la solución).

Para disminuir las inestabilidades numéricas en problemas de capa límite, se utiliza el método de Upwinding.
El método consiste en descentrar la derivada primera, discretizándola en adelanto o en atraso:

Si $\beta > 0 \Rightarrow$ usamos der. prim. en adelanto.
Si $\beta < 0 \Rightarrow$ usamos der. prim. en atraso.

Este método nos da mayor precisión en la zona de la capa límite.

APROXIMACIÓN DE FUNCIONES

Problema: Aproximar f utilizando funciones "convenientes"

$$f \cong f^{*} \longrightarrow \text{función aproximante.}$$

$$\boxed{f^{*}(x) = c_0\,\varphi_0(x) + c_1\,\varphi_1(x) + \dots + c_n\,\varphi_n(x) = \sum_{j=0}^{n} c_j\,\varphi_j(x)}$$

$\varphi_j(x)$: $n+1$ funciones (no lineales) elegidas a priori.
c_j : $n+1$ parámetros a determinar.

Objetivo de la aproximación (discreta): obtener los $n+1$ coeficientes c_j de forma que se cumpla en forma exacta o lo más próxima posible $f^{*}(x_i) = f(x_i)$ para todos los valores de x_i con $i = 0, \dots, m$ los $m+1$ datos de la red.

$m < n \longrightarrow$ problema indeterminado (∞ soluciones)
$m = n \longrightarrow$ usualmente solución única (interpolación)
$m > n \longrightarrow$ solo eventualmente $f^{*}(x_i) = f(x_i)\ \forall i$. Problema sobredeterminado: sol. aprox.

☒ AJUSTE DISCRETO POR CUADRADOS MÍNIMOS: $(m > n)$

$$e = \sum_{i=0}^{m} e_i^2 = \sum_{i=0}^{m} (f_i - f_i^*)^2 = \sum_{i=0}^{m} \left(f_i - \sum_{j=0}^{n} c_j \varphi_j(x)\right)^2 \qquad \text{(error global)}$$

Como $e = e(c_j) \Rightarrow$ para minimizar el error e se debe cumplir que: $\frac{\partial e}{\partial c_K} = 0$, $K = 0, \dots, n \Rightarrow$

$$\Rightarrow \boxed{\sum_{j=0}^{n} c_j \langle \varphi_j ; \varphi_K \rangle = \langle f ; \varphi_K \rangle} \quad \text{con } K = 0, \dots, n,\ j = 0, \dots, n$$

producto interno

$n+1$ ecuaciones normales
$n+1$ incógnitas

Propiedades de ecuaciones normales:

1. Existencia y unicidad de la solución: si $\{\varphi_j\}$ son l.i. $\Rightarrow$ sol. única.
2. Ortogonalidad:

$$\langle \varphi_j ; \varphi_K \rangle = \|\varphi_K\|^2 \delta_{jK} \qquad \text{con} \quad \delta_{jK} = \begin{cases} 0 & \text{si } j \neq K \\ 1 & \text{si } j = K \end{cases} \quad \Rightarrow$$

$$\Rightarrow c_K = \frac{\langle f ; \varphi_K \rangle}{\|\varphi_K\|^2} \qquad \text{con} \quad K = 0, \dots, n$$

3. Ortonormalidad:

$$\langle \varphi_j ; \varphi_K \rangle = \delta_{jK} \qquad \text{con} \quad \delta_{jK} = \begin{cases} 0 & \text{si } j \neq K \\ 1 & \text{si } j = K \end{cases} \quad \Rightarrow$$

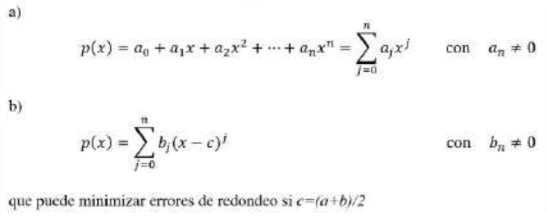
$$\Rightarrow c_K = \langle f ; \varphi_K \rangle \qquad \text{con} \quad K = 0, \dots, n$$

☒ INTERPOLACIÓN: $(m = n)$

Existencia de la solución:

Por Teo. de Weierstrass, si aumentamos el grado de un polinomio suficientemente, podemos representar cualquier función con un error tan pequeño como queramos.

Formas de expresar polinomios: $p(x) \in P_n[x]$



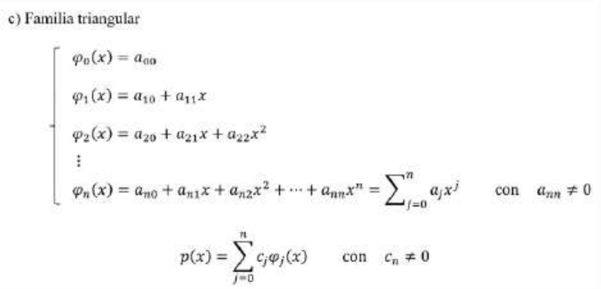
a)

$$p(x) = a_0 + a_1 x + a_2 x^2 + \cdots + a_n x^n = \sum_{j=0}^{n} a_j x^j \qquad \text{con} \quad a_n \neq 0$$

b)

$$p(x) = \sum_{j=0}^{n} b_j (x-c)^j \qquad \text{con} \quad b_n \neq 0$$

que puede minimizar errores de redondeo si *c=(a+b)/2*



c) Familia triangular

$$\begin{cases} \varphi_0(x) = a_{00} \\ \varphi_1(x) = a_{10} + a_{11}x \\ \varphi_2(x) = a_{20} + a_{21}x + a_{22}x^2 \\ \vdots \\ \varphi_n(x) = a_{n0} + a_{n1}x + a_{n2}x^2 + \cdots + a_{nn}x^n = \sum_{j=0}^{n} a_j x^j \qquad \text{con} \quad a_{nn} \neq 0 \end{cases}$$

$$p(x) = \sum_{j=0}^{n} c_j \varphi_j(x) \qquad \text{con} \quad c_n \neq 0$$

Problema:

Determinar un polinomio de grado n que concuerde con los valores $f_i \equiv f(x_i)$ de la función f sobre una red de $m+1$ puntos x_i.

Unicidad de la solución:

El problema tiene solución única, y utilizando la familia triangular, el polinomio interpolador es:

$$\boxed{p(x) = c_0 + c_1(x-x_0) + c_2(x-x_0)(x-x_1) + \ldots + c_m(x-x_0)(x-x_1)\cdot \ldots \cdot (x-x_{m-1})}$$

Error de truncamiento:

Sea $f \in C^{(m+1)}[a;b]$. Si $\overset{\text{pol. interp.}}{p(x)} \in P_m[x]$ tal que $p(x_i) = f(x_i)$; $i = 0, 1, \ldots, m$

$$\boxed{f(x) - p(x) = \frac{f^{(m+1)}(\gamma)}{(m+1)!}(x-x_0)\cdot(x-x_1)\cdot \ldots \cdot (x-x_m)}$$; con $\gamma \in [a;b]$ desconocido.

<u>Obs</u>: $f(x) - p(x) = 0 \quad \forall\, x = x_i$

Formas de econtrar el polinomio interpolador:

- Lagrange
- Newton
- Hermite
- Tchebychef

INTERPOLACIÓN DE LAGRANGE:

Sea f una función definida en $x_0, \ldots, x_m$, $m+1$ puntos distintos, entonces $\exists!\, p(x) \in P_n[x]$ con $n \leq m$ tal que $p(x_i) = f(x_i)$; $i = 0, 1, \ldots, m$

$$\boxed{p_L(x) = \sum_{k=0}^{n} L_{nk}(x) f(x_k)} \quad ; \quad L_{nk}(x) = \prod_{\substack{i=0 \\ i \neq k}}^{n} \frac{x - x_i}{x_k - x_i} \Rightarrow L_{nk}(x) = \begin{cases} 0 & j = i \neq k \\ 1 & j = k \end{cases}$$

Polinomio de Lagrange.

INTERPOLACIÓN DE NEWTON:

$$f[x_0, x_1, \ldots, x_{k-1}, x_k, x] \equiv \frac{f[x_0, x_1, \ldots, x_{k-1}, x] - f[x_0, x_1, \ldots, x_{k-1}, x_k]}{x - x_k} \Rightarrow$$

$$\Rightarrow f[x_0, x_1, \ldots, x_{k-1}, x_k] = c_k \quad ; \quad k \leq m$$

$$\boxed{p_N(x) = f_0 + \sum_{j=1}^{m} f[x_0, x_1, \ldots, x_j](x-x_0)(x-x_1)\ldots(x-x_{j-1})}$$

Polinomio de Newton.

Diferencias divididas: se peden calcular progresivamente observando que:

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

<u>Esquema de cálculo</u>:

$$\frac{f(x_1) - f(x_1)}{x_1 - x_0} = c_1 \qquad \frac{f[x_1, x_2] - f[x_0, x_1]}{x_2 - x_0} = c_2$$

$$\begin{array}{cc|lll}
x_0 & f_0 \to = c_0 & & & \\
x_1 & f_1 & f[x_0, x_1] & f[x_0, x_1, x_2] & f[x_0, x_1, x_2, x_3] = c_3 \\
x_2 & f_2 & f[x_1, x_2] & f[x_1, x_2, x_3] & \\
x_3 & f_3 & f[x_2, x_3] & &
\end{array}$$

INTERPOLACIÓN DE HERMITE:

↳ Mejorar la aproximación de f cuando se conocen una o más de sus derivadas en los puntos de interpolación.

- **Método directo:**

Sea $f \in C^1[a;b]$ y $x_0, x_1, \dots, x_m$ distintos $\in [a;b]$, entonces el polinomio de mínimo grado que coincide con f y f' en $x_0, x_1, \dots, x_m$ tiene grado a lo sumo $2m+1$ y está dado por el Polinomio de Hermite:

$$\boxed{H_{2m+1} = \sum_{j=0}^{m} f(x_j)\, H_{mj}(x) + \sum_{j=0}^{m} f'(x_j)\, \hat{H}_{mj}(x)}$$

, con:

$$H_{mj}(x) \equiv \{1 - 2(x - x_j)\, L'_{mj}(x_j)\}\, L^2_{mj}(x)$$

$$\hat{H}_{mj}(x) \equiv (x - x_j)\, L^2_{mj}(x)$$

$L_{mj}(x)$: polinomio de Lagrange

$$L'_{mj}(x) = \frac{dL_{mj}}{dx}$$

INTERPOLACIÓN DE TCHEBYCHEF:

↳ Para evitar el fenómeno de Runge → a mayor grado, peor convergencia en los extremos en funciones aproximantes de nodos equidistantes o funciones para en intervalos simétricos respecto al 0.

Minimización de la función error seleccionando nuevos puntos $x_0, x_1, \dots, x_m$ a partir de las raíces de los polinomios T_{m+1} (absisas de Tchebychef).

Los polinomios de Tchebychef constituyen una familia triangular de polinomios ortogonales y se definen en forma recursiva:

$$\boxed{T_{n+1}(x) = 2x\, T_n(x) - T_{n-1}(x)}$$

o en forma trigonométrica:

$$T_n(x) = \cos(n.\ \arccos(x)) = \text{coship}(n\ \text{arcoship}(x))$$

↱ Si se conoce f.

Pasos:

1) Si $t \in [a;b]$ se requiere aplicar una transformación lineal para $x \in [-1;1]$:

$$t = \frac{a+b}{2} + \frac{b-a}{2}\, x$$

2) Luego, se definen los nuevos $m+1$ puntos como las absisas de Tchebychef:

$$x_K = \cos\left(\frac{\pi}{2} \cdot \frac{2K-1}{n}\right)$$ → raíces de los polinomios de Tchebychef T_{m+1}

3) Se obtiene la nueva tabla de datos evaluando la función f en las absisas de Tchebychef: $[x_K ; f(x_K)]$

4) Se obtiene el polinomio interpolador utilizando alguna de las metodologías disponibles.

5) Se anti-transforma para volver al intervalo original $[a;b]$.

Ajuste vs. Interpolación:

- Usamos ajuste si tenemos una gran cantidad de datos (para mantener la simpleza de la aprox.).
- Problema mejor condicionado si se aproxima mediante cuadrados mínimos (ajuste) con un polinomio $P_n[x]$ de grado $n < 2\sqrt{m}$.

INTEGRACIÓN:

Problema: hallar una aproximación para la integral definida $I = \int_a^b f(x)\,dx$

Solución: cuadratura numérica: $I \cong \sum_i w_i f(x_i)$ $\Rightarrow$ cada método de cuadratura queda definido por el conjunto $\{w_i\}$ y por los puntos de evaluación $\{x_i\}$

☒ ESCALAS DE INTEGRACIÓN:

Descomponiendo el intervalo $[a,b]$ en sub-intervalos locales $[x_{i-1}, x_i]$:

$$I = \int_a^b f(x)\,dx = \sum_{i=1}^{N} \int_{x_{i-1}}^{x_i} f(x)\,dx = \sum_{i=1}^{N} I_i \quad , \text{ con } I_i \equiv \int_{x_{i-1}}^{x_i} f(x)\,dx$$

y, en el caso de sub-intervalos locales regulares:

$$x_i = a + ih \quad ; \quad i = 0, \dots, N \quad ; \quad h = \frac{b-a}{N}$$

Considerando que los sub-intervalos locales $[x_{i-1} ; x_i]$ son suficientemente pequeños, se desarrolla $f(x)$ alrededor del punto medio del sub-intervalo $x_i^{(m)} = \frac{1}{2}(x_{i-1} + x_i)$, nos queda:

$$I_i = h\left\{ \underset{\substack{\downarrow \\ n=0}}{f_m} + \frac{1}{24} \underset{\substack{\downarrow \\ n=2}}{f_m''} h^2 + \frac{1}{1920} \underset{\substack{\downarrow \\ n=4}}{f_m^{iv}} h^4 + \dots \right\}$$

Lo falta error de truncamiento.

☒ MÉTODO DEL RECTANGULO: → precisión 1

REGLA PARA SUB-INTEGRALES LOCALES:

Aproximación de I:

$$\boxed{R_i = h f_m = h f\left(\frac{x_{i-1} + x_i}{2}\right)}$$

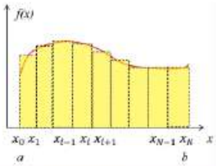
Error de truncamiento:

$$\boxed{R_i - I_i = h\left\{ -\frac{1}{24} f_m'' h^2 - \frac{1}{1920} f_m^{iv} h^4 + O(h^6) \right\}}$$

REGLA PARA INTEGRAL GLOBAL:

Aproximación de I:

$$\boxed{R = h \sum_{i=1}^{N} f\left(\frac{x_{i-1} + x_i}{2}\right)}$$



Error de truncamiento:

$$\boxed{R - I = \sum_{i=0}^{N} (R_i - I_i) = -\frac{1}{24} h^2 \underbrace{\left\{ \sum_{i=1}^{N} f_m'' h \right\}}_{\text{cte si } h \to 0} + O(h^4)}$$

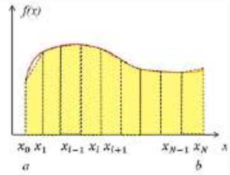
☒ MÉTODO DEL TRAPECIO: → precisión 1

REGLA PARA SUB-INTEGRALES LOCALES:

Aproximación de I: $\boxed{T_i = \frac{h}{2}\left(f(x_{i-1}) + f(x_i)\right)}$

Error de truncamiento: $\boxed{T_i - I_i = h\left\{\frac{1}{12} f_m'' h^2 + \frac{1}{480} f_m^{iv} h^4 + \dots\right\}}$ (orden 2)

REGLA PARA INTEGRAL GLOBAL:



Aproximación de I: $\boxed{T = h\sum_{i=1}^{N} \frac{f(x_{i-1}) + f(x_i)}{2}}$

Para minimizar errores se puede reducir el número de operaciones:

$$T = h\frac{f_0}{2} + h\sum_{i=1}^{N-1} f_i + h\frac{f_N}{2}$$

Error de truncamiento: Surge de R-I y sigue siendo de orden 2.

☒ EXTRAPOLACIÓN DE RICHARDSON: → precisión 2

Precisión: la precisión de una fórmula de cuadratura es el entero positivo más grande n tal que la fórmula es exacta para: x^k, con $k = 0, 1, \dots, n$

Buscamos obtener un resultado de precisión 2 utilizando un método de precisión 1.

Aproximación de I de orden $\overset{\text{orden}}{h^{j}}$: $\boxed{N_j(h) = N_{j-1}\left(\frac{h}{2}\right) + \left(\frac{N_{j-1}(h/2) - N_{j-1}(h)}{2^{j-1} - 1}\right)}$

Obs: No es restrictivo el uso de $h/2$, pero la aprox. solo es apta para esa partición de h.

☒ MÉTODO DE SIMPSON:

REGLA PARA SUB-INTEGRALES LOCALES:

Aproximación de I: $\boxed{S_i = \frac{h}{3}\left[f_{i-1} + 4f_i + f_{i+1}\right]}$

Error de truncamiento: $\boxed{S_i - I_i = -\frac{1}{90} f^{iv}(\mu) h^5}$

REGLA PARA INTEGRAL GLOBAL:

Aproximación de I: $\boxed{S(h) = \frac{h}{3}\left[f(a) + 2\overset{\text{puntos pares}}{\sum_{i=1}^{\frac{N}{2}-1} f_{2i}} + 4\overset{\text{puntos impares.}}{\sum_{i=1}^{\frac{N}{2}} f_{2i-1}} + f(b)\right]}$

para N par, con $x_i = a + ih$; $i = 0, \dots, N$ $\quad h = \frac{b-a}{N}$

Error de truncamiento: $\boxed{S - I = -\frac{b-a}{180} f^{iv}(\mu) h^4}$

☒ **MÉTODO DE ROMBERG:**

1) Utilizar la regla compuesta del Trapecio (intervalo global) como aproximación preliminar.
2) Aplica extrapolación de Richardson.

☒ **CUADRATURA DE GAUSS:**

↳ Buscamos seleccionar puntos óptimos para mejorar la aproximación.

Aproximación de I: $\boxed{I = \int_{-1}^{1} f(x)\,dx \cong G = \sum_{i=1}^{n} c_i f(x_i)}$; n: cantidad de puntos.

- $x_1, x_2, \dots, x_n \in [a;b]$
- $c_1, c_2, \dots, c_n$ tales que se minimice el error de truncamiento $G - I$

Obs: Los "puntos de Gauss se obtienen como las raíces de los polinomios de Legendre.

El criterio de selección de las $2n$ incógnitas $\{x_i, c_i\}$ es que integren en forma exacta el polinomio de mayor grado posible que se puede construir con $2n$ parámetros: $p(x) \in P_{2n-1}[x]$.
Otra forma de hacerlo es que, para hallar el valor de las $2n$ incógnitas $\{x_i, c_i\}$, queremos que G integre de forma exacta los polinomios $P_0[x], \dots, P_{2n-1}[x]$. O sea:

$$\begin{cases} \int_{-1}^{1} dx = c_1 + \dots + c_n \\ \vdots \\ \int_{-1}^{1} x^{2n-1} = c_1 \cdot x^{2n-1} + \dots + c_n \cdot x^{2n-1} \end{cases}$$

, y resolvemos el SENL.