

Métodos numéricos

Resumen práctico

Diego González

2007

diesgomo@gmail.com

1. Errores

1.1 Representación de números

- Representación de un número¹:

a se representa como $v = (-1)^s \cdot (1.m)_2 \cdot 2^e$ donde:

Formato	Cantidad de bits	s	$1.m$	e	e_{\min}	e_{\max}
Simple	32	1 bit	1 + 23 bits	8 bits	-126	127
Doble	64	1 bit	1 + 52 bits	11 bits	-1022	1023

- Definición de redondeo:
El redondeo $fl(x)$ de x es α el número de punto flotante más cercano a x
- Definición de truncamiento (chopping)²:
 $\bar{x} \in FP / \bar{x} = \max_{y \in FP} \{y \leq x\}$
- Definición del épsilon de la máquina:
 ε_m es el menor número en punto flotante tal que $\varepsilon_m + 1 > 1$
- Definición de redondeo³:
 $\bar{x} \in FP / \bar{x} = \min_{y \in FP} \{y - x\}$
- Definición de error absoluto:
 $E_x = x - \bar{x}$
- Definición de error relativo⁴:
 $\delta_x = e_x = \frac{|E_x|}{x}$
- Error de representación:
 $\bar{x} = x(1 + \delta_x)$ con $|\delta_x| < \varepsilon_M$

¹ m es la mantisa, $1.m$ está en base 2 (el . es el decimal binario), y el coeficiente de m indica si está normalizado (1) ó denormalizado (0).

$e_{\min} \leq e \leq e_{\max}$

² Es el acto de quedarse con el que tiene todos los bits de la representación correcta

³ Dado un número x representarlo con el número de punto flotante más cercano es el acto de redondear

⁴ Si se da un error como porcentaje asumiremos que se trata de un error relativo.

El error relativo de \bar{x} es $\delta = \frac{\text{error absoluto}}{\text{valor verdadero}}$ siempre que valor verdadero $\neq 0$

1.2 Errores en la representación

- Tipos de errores:

Nombre	Descripción	Error relativo
Cancelación catastrófica	Es la pérdida de precisión (aumento del error relativo) cuando se suma/resta	$\left \frac{(\bar{x} - x) - (\bar{y} - y)}{x - y} \right = \left \frac{x\delta_x - y\delta_y}{x - y} \right \leq$ $\leq \frac{x + y}{x - y} \varepsilon_M \quad \text{si } x - y \approx 0 \text{ ocurre la cancelación catastrófica}$
Error de truncamiento de un cociente incremental ⁵	Es el error al dar $\Delta_h f$ como valor de $f'(x)$	$\Delta_h f(x) = \frac{f''(\xi)h}{2}, \quad \xi \in [x, x+h]$
Error de truncamiento de la diferencia centrada ⁶	$\delta_h f(x) = \frac{f(x+h) - f(x-h)}{2h}$ aproxima a $f'(x)$	$\delta_h f(x) - f'(x) = \frac{f'''(\xi)h^2}{6},$ $\xi \in [x-h, x+h]$

- Error total de aproximación de $f'(x)$ ⁷:

$$e_t = e_t + e_r = \frac{f(x+h) - f(x)}{h} - f'(x) \quad \text{donde:}$$

$$e_r = \frac{f(x+h) - f(x-h)}{h} - \frac{f(x) - f(x)}{h} = \frac{\delta_1 f(x+h)}{h} - \frac{\delta_f f(x)}{h} = \frac{2\varepsilon_M |f(x)|}{h}$$

$$e_t = \frac{f''(\xi)h}{2}$$

- Cálculo del h_0 óptimo:

$$e_t \leq 2 \frac{|f(x)|\varepsilon_M}{h} + \frac{|f''(\xi)|h}{2} \xrightarrow{\text{Derivando}} h_0 = 2 \sqrt{\frac{|f(x)|}{|f''(\xi)|} \varepsilon_M} \approx \sqrt{\varepsilon_M}$$

- Extrapolación de Richardson⁸:

- Objetivo:

Calcular $\lim_{h \rightarrow 0} \bar{f}(x, h) = f(x)$. $\bar{f}(x, h)$ estimados de f dependiendo de h

- Resolución:

Se estima $\bar{f}(x, h) = f(x) + c_p h^p + o(h^r)$ con $r > p$ eliminando $c_p h^p$

- Conclusión:

$$f(x) \cong \frac{f(x, \frac{h}{q}) + \bar{f}(x, \frac{h}{q}) - \bar{f}(x, h)}{q^p - 1}$$

⁵ Para calcular el error relativo se hace Taylor:

$$f(x+h) = f(x) + f'(x)h + \frac{f''(\xi)h^2}{2!}, \quad \xi \in [x, x+h]$$

⁶ Para calcular el error relativo se hace Taylor.

⁷ e_t = error de truncamiento. e_r = error de redondeo.

⁸ δ_h nota $\delta_h f(x)$. El proceso sistemático es escribir la igualdad en función de δ_h y dejar lo que interesa despejar.

- Teorema de Richardson:

$$f(x) \cong f(x) + \sum_{i=1}^{\infty} c_i h^{p_i} \quad \text{con } p_i < p_{i+1}, \quad i \geq 1$$

$$\left. \begin{array}{l} \text{Se define } \left\{ \begin{array}{l} \overline{f_1}(x, h) = \overline{f}(x, h) \\ \overline{f_{k+1}}(x, h) = \overline{f_k}\left(x + \frac{h}{q}\right) + \frac{\overline{f_k}\left(x, \frac{h}{q}\right) - \overline{f_k}(x, h)}{q^{p_k} - 1}, \quad q > 1 \end{array} \right. \right\} \Rightarrow \overline{f_n}(x, h) = \overline{f}(h, k)$$

- Simple:

Este sirve para disminuir el error de truncamiento para cualquier serie (en particular, el de la diferencia centrada) en el orden de h^4 ($O(h^4)$).

En él se aproxima $f'(x) - \frac{h^4}{100.120} f^{(5)}(x)$ como $\delta_{h/10} + \frac{\delta_{h/10} - \delta_h}{99}$

- Repetida:

$$\left. \begin{array}{l} R_h = f(x) + ch^4 + c_1 h^6 + \dots \\ R_{h/10} = f'(x) + \frac{ch^4}{10^4} + \frac{c_1 h^6}{10^6} + \dots \end{array} \right\} \Rightarrow f'(x) + kh^6 + \dots = R_{h/10} + \frac{R_{h/10} - R_h}{9999}$$

- Fórmula del error:

$$f: R^n \rightarrow R. \Delta f = f(x + \Delta x) - f(x) \Rightarrow \Delta f = \frac{\partial f(x + \theta \Delta x)}{\partial x_1} \Delta x_1 + \dots + \frac{\partial f(x + \theta \Delta x)}{\partial x_n} \Delta x_n \Rightarrow$$

$$\Rightarrow \Delta f = \sum_{i=1}^n \frac{\partial f(x + \theta \Delta x)}{\partial x_i} \Delta x_i$$

- Corolario de la fórmula del error:

$$|\Delta f| \leq \sum_{i=1}^n \left| \frac{\partial f(x + \theta \Delta x)}{\partial x_i} \right| |\Delta x_i|$$

1.3 Condiciones

- Definición de número de condición de un algoritmo⁹:

Es el número de condición de un algoritmo A con una entrada x es:

$$C_A(x) = \frac{\|A^{-1}(\overline{A}(x))\|}{\|x\|_{\mathcal{E}_M}} = \frac{\|\delta_x\|}{\|x\|_{\mathcal{E}_M}}$$

- Definición de número de condición de un problema¹⁰:

El número de condición de un problema P con una entrada x es:

$$C_{P(x)} = \max_{\substack{\|\delta_x\| < \varepsilon_M \\ \|x\|}} \left\{ \frac{\|P(x + \delta_x) - P(x)\| / \|P(x)\|}{\|\delta_x\| / \|x\|} \right\}$$

⁹ Es la perturbación “soportable” de la entrada

¹⁰ Mide el impacto sobre los resultados ante una perturbación en los datos de entrada

2. Sistemas de ecuaciones lineales

2.1 Descomposición LU sin intercambio de filas¹¹

- Definición de matriz L y de matriz U :

$$L = \begin{pmatrix} 1 & 0 & \dots & 0 \\ l_{21} & 1 & \dots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ l_{n1} & \dots & \dots & 1 \end{pmatrix} \quad U = \begin{pmatrix} u_{11} & \dots & \dots & u_{1n} \\ 0 & u_{22} & \dots & \vdots \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \dots & u_{nn} \end{pmatrix}$$

- Descomposición LU de A ¹²:
 U se obtiene de la escalerización de A , y L de los cofactores obtenidos al escalerizar A .
- Resolución de $Ax = b$:
Se resuelve y de $Ly = b$ ¹³ y x de $Ux = y$ ¹⁴

2.2 Descomposición LU en el caso general

- Matriz P :
Si $|A| \neq 0$, entonces, existe P matriz de permutación de A tal que PA tiene descomposición LU : $PA = LU$.
Esto se logra poniendo ceros en todas las filas y un uno por cada fila, cuya ubicación será en la columna cuyo índice coincida con el índice de la fila origen¹⁵. Esto es, si se intercambian las filas i y r en la matriz original, se intercambian las filas i y r de la matriz identidad correspondiente (del mismo tamaño que la matriz original).

¹¹ En esta sección se asume A escalerizable (de rango n) no singular, asegurando la existencia de la expresión $A = LU$

¹² Esta descomposición es única. Los cofactores son los resultados de las divisiones de las entradas con objeto de eliminar determinado valor.

¹³ Sustitución hacia adelante: $y_i = b_i - \sum_{j=1}^{i-1} a_{ij}x_j$ con $i = 2, \dots, n$

¹⁴ Sustitución hacia atrás: $x_i = \frac{d_i - \sum_{j=i+1}^n a_{ij}x_j}{a_{ii}}$ con $i = n-1, \dots, 1$

¹⁵ Ejemplo:

$$A = \begin{pmatrix} 1 & 2 & 3 \\ 3 & 1 & 0 \\ 2 & 1 & 5 \end{pmatrix} \Rightarrow P = \begin{pmatrix} 0 & 0 & 1 \\ 1 & 0 & 0 \\ 0 & 1 & 0 \end{pmatrix}, \text{ pues, } PA \text{ la matriz de filas permutadas será } PA = \begin{pmatrix} 2 & 1 & 5 \\ 1 & 2 & 3 \\ 3 & 1 & 0 \end{pmatrix}$$

- Algoritmo para hallar la descomposición LU de PA^{16} :
 1. Hacer de P y L la matriz identidad $n \times n$
 2. Para i variando entre 1 y n hacer:
 3. Si $a_{ii} = 0$ buscar el primer $j > i$ tal que $a_{ij} \neq 0$. Intercambiar en P la fila i con la fila j , intercambiar en A la fila i con la fila j , intercambiar en L la fila i con la fila j , para la matriz L dejar incambiada la diagonal principal (donde la entrada es 1)
 4. Para $j > i$ hasta que $j = n$, hacer $l_{ji} = \frac{a_{ji}}{a_{ii}}$. Cambiar la fila F_j de A por $F_j - l_{ji}F_i$. Insertar en la entrada (j,i) de L el valor l_{ji}
 5. Tomar como nuevas matrices A , P y L a las obtenidas con las transformaciones anteriores.
 6. Volver a 2. incrementando en 1 el valor de i .
 7. Al finalizar se tiene como último valor de A a la matriz U , y las correspondientes P y L .
- Resolución de $Ax = b$:
Se resuelve $P Ax = Pb$, esto es $LUx = Pb$, esto es, $Ly = Pb$ y luego $Ux = y$.

2.3 Pivoteo

- Pivoteo parcial:
Si una matriz se escaleriza hasta la fila $k-1$, intercambiar la fila k con la fila j ($j \geq k$) tal que $a_{jk} = \max_{i \geq k} \{ |a_{ik}| \}$ produce que los multiplicadores que se calculan a continuación sean ≤ 1
- Pivoteo total:
Si una matriz se escaleriza hasta la fila $k-1$, se halla el que tiene el valor absoluto más grande en el cuadro a_{ij} ($n \geq i, j \geq k$), el intercambio de la fila r con la fila k y la columna s con la columna k es un mejor procedimiento, pero más costoso ($O(x^3)$)
- Casos en que no es necesario realizar pivoteos durante la eliminación Gaussiana:
 - A es diagonal dominante: $|a_{ii}| > \sum_{i \neq j=1}^n |a_{ij}| \quad \forall i = 1, \dots, n$
 - A es simétrica definida positiva: $A = A^t, x^t Ax > 0 \quad \forall x \in \mathbb{R}^n \setminus \vec{0}$

2.4 Matrices esparzas

- Definición de matriz esparza¹⁷:
Son las que cumplen que:
 1. Tienen muchos 0
 2. Tienen una estructura apta para ser explotadas
- Definición de matriz banda¹⁸:
Es una $\mathcal{M}_{n \times n} / \exists m, m \ll n / \text{si } |i-j| > m \Rightarrow a_{ij} = 0$

¹⁶ Los elementos a_{ij} a los que se refiere no son los originales de A , sino los que se obtienen al ir aplicando los sucesivos pasos.

¹⁷ Las que se presentan en esta sección son ejemplos de matrices esparzas

¹⁸ Esto es, una matriz cuadrada en la que todos los elementos son cero con excepción de una banda centrada sobre la diagonal principal. Una matriz que cumple esto se llama bandeada.

2.6 Métodos iterativos de resolución

- Descripción²¹:

La idea es aproximar la solución de $AX = b$ con $X^{(1)}, \dots, X^{(r)}, \dots$ converge a X

Se parte de elegir $M/MX^{(k+1)} = (M - A)X^{(k)} + b$

- Definición de $e^{(k)}$:

$$e^{(k)} = X^{(k)} - X^{(\infty)}$$

- Definición de Q ²²:

$$Q = M^{-1}(M - A)$$

- Condición necesaria de convergencia:

$\|Q\| < 1 \Rightarrow$ el método iterativo converge

- Condición necesaria y suficiente para la convergencia de un método iterativo:

Un método iterativo converge si $\rho(Q) < 1$

- Definición de q^k ²³:

$$q^k = \frac{\|e^{(k+1)}\|}{\|e^{(k)}\|}$$

- Proposición:

$$\lim_{k \rightarrow \infty} q^k = \rho(Q)$$

- Definición de E, D y F :

Son las matrices con ceros en todas las entradas excepto las siguientes:

$$A = \begin{pmatrix} & & & & -F \\ & \ddots & & & \\ & & \ddots & & \\ & & & D & \\ -E & & & & \ddots \end{pmatrix}$$

- Método de Jacobi:

- Forma matricial: $DX^{(k+1)} = (E + F)X^{(k)} + b$

- Término i -ésimo: $x_i^{(k+1)} = \frac{b - \sum_{j=1}^n a_{ij}x_j^{(k)}}{a_{ii}}$

- Método de Gauss-Siedel²⁴:

- Forma matricial: $(D - E)X^{(k+1)} = FX^{(k)} + b$

- Término i -ésimo: $x_i^{(k+1)} = \frac{b - \sum_{j=1}^{i-1} a_{ij}x_j^{(k+1)} - \sum_{j=i+1}^n a_{ij}x_j^{(k)}}{a_{ii}}$

²¹ $X^{(\infty)}$ notará la solución del sistema

²² Observar que $e^{(k)} = M^{-1}(M - A)e^{(k-1)} = Qe^{(k-1)}$

²³ q^k mide la velocidad de convergencia

²⁴ A diferencia del método de Jacobi este método considera calculados los términos de un paso para calcular otros términos en el mismo paso.

- Mejoras en la convergencia por medio de la relajación²⁵:

Se considera $M = \frac{1}{w}D - E$. En cada paso se obtiene $Q_w = M^{-1}(M - A)$ y se halla

$$|Q_w - \lambda Id| = p_w(x) \rightarrow G_w(Q).$$

Generalmente se determina empíricamente y lo que hace es, luego de calcular cada nuevo valor, se modifica por un promedio ponderado de los resultados de las iteraciones anterior y actual

²⁵ Si $w \equiv 1$ se obtiene el método de Gauss-Siedel

3. Sistemas no lineales²⁶

3.1 Método de bipartición²⁷

- Descripción:
Dado un intervalo $[a, c]$ se considera b como el punto medio, luego se analiza si $f(a)f(b)$ es menor, mayor ó igual a 0. Si es igual a 0, entonces, b es raíz. Si es menor a 0, entonces, se analiza ese tramo repitiendo el procedimiento que termina cuando $f(a)f(x) < h$ donde h es una cota preestablecida y x es uno de los bordes (resultado de la recursión). Si es mayor a 0, se analiza el tramo contiguo.
- Formalización:
Dados a_0, b_0 el intervalo generado es $I_k = (a_k, b_k)$ con $k \in \mathbb{N}$. El punto medio buscado será dado por la sucesión $m_k = \frac{a_{k-1} + b_{k-1}}{2}$ y se define
$$(a_k, b_k) = \begin{cases} (m_k, b_{k-1}) & \text{si } f(m_k) < 0 \\ (a_{k-1}, m_k) & \text{si } f(m_k) > 0 \end{cases}$$
Resolviendo la ecuación de recurrencia se tiene que la raíz $\alpha = m_{n+1} \pm d_n$ donde $d_n = 2^{-n-1}(b_0 - a_0)$ y n es el paso de recursión según la tolerancia prefijada de 2^{-n-1}
- Detención del algoritmo:
 - Por número de iteraciones: Puede calcularse a priori para $e_D \in (0, 1]$ el error deseado como $n = \log_2 \left(\frac{b-a}{e_D} \right)$
 - Por error relativo²⁸: $e_r = 100 \left| \frac{x_{\text{actual}} - x_{\text{anterior}}}{x_{\text{actual}}} \right|$
- Limitaciones:
Detecta la existencia de una sola raíz sin indicar su multiplicidad o la existencia de otras raíces en el intervalo inicial.

3.2 Método de iteración simple de punto fijo o método iterativo general (MIG)²⁹

- Objetivo:
Obtener una nueva aproximación x_{i+1} del valor de una raíz de una función según x_i
- Procedimiento:
Dada $f(x) = 0$ se lleva mediante transformaciones algebraicas o sumando x a ambos lados a la forma $g(x) = x$ para hallar su punto fijo (el valor de x tal que aplicar g es como aplicar la identidad)
- Error:
$$e_r = 100 \left| \frac{x_{i+1} - x_i}{x_{i+1}} \right|$$

²⁶ El objetivo de esta sección es aproximar el valor de la raíz de una función

²⁷ Obtenido de la demostración del teorema de Bolzano-Weierstrass.

Éste es un método cerrado (usa intervalos)

²⁸ Éste es menor o igual al absoluto

²⁹ Éste y los siguientes son los métodos abiertos (usan los valores de las soluciones)

- Convergencia:

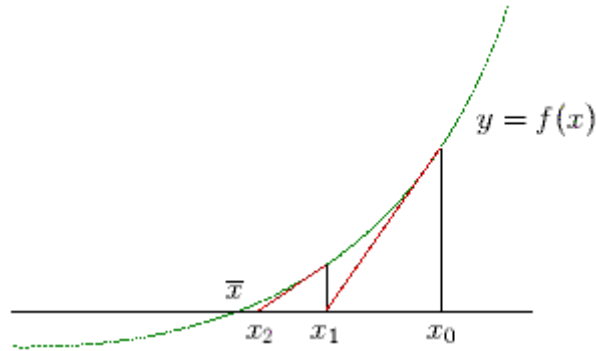
El método converge $\Leftrightarrow |g'(x)| < 1$ en la región de interés

- Definición de orden de convergencia³⁰:

Si $x_n \xrightarrow{n \rightarrow \infty} \alpha$, su orden de convergencia es $p \Leftrightarrow \lim_{n \rightarrow \infty} \frac{|x_{n+1} - \alpha|}{|x_n - \alpha|^p} = \beta \in \mathbb{R} \setminus \{0\}$

3.3 Método de Newton-Raphson

- Descripción:



Este método comienza con x_0 un punto "razonablemente cerca de la raíz" y se arma la

sucesión $x_{n+1} = x_n - \frac{f(x_n)}{f'(x_n)}$ donde el error vendrá dado por $\left| \frac{f(x)}{m} \right|$ donde $m = \min_{x \in [a,b]} |f'(x)|$

- Error:

$$e_{r,i+1} = -\frac{f''(x_R)}{2f'(x_R)} e_{r,i}^2 \text{ donde } x_R \text{ es el valor verdadero de la raíz}$$

- Convergencia:

En general converge eficientemente aunque pueden presentarse problemas para ciertas funciones y ciertos valores iniciales, en cuyos casos la convergencia es muy lenta o directamente no converge.

- Método de Newton-Raphson para varias variables³¹:

Se considera el sistema $\begin{cases} J_f(x_i) \delta_i = -f(x_i) \\ x_{i+1} = x_i + \delta_i \end{cases}$ de donde se despeja δ_i de la primera

ecuación y se utiliza en la segunda para obtener x_{i+1} y de esta forma los puntos de la sucesión (por lo que en cada paso debe resolverse ese sistema)

3.4 Método de la secante

- Descripción³²:

Dados x_0 y x_1 se calcula $f(x_1)$ y $f(x_0)$ para aproximar la sucesión.

$$x_{n+2} = x_{n+1} - f(x_{n+1}) \frac{x_{n+1} - x_n}{f(x_{n+1}) - f(x_n)}$$

- Velocidad de convergencia:

$$\frac{1}{2}(1+\sqrt{5}) \sqrt{\frac{f''(x_R)}{2f'(x_R)}} \text{ donde } x_R \text{ es el valor verdadero de la raíz}$$

³⁰ En el MIG, p es el orden de la primera derivada no nula de g en α ; y $\beta = \frac{1}{p!} \left| \frac{d^p}{dx^p} g(\alpha) \right|$

³¹ En este caso no convergirá para matrices singulares (con determinante nulo)

³² Igual al método de Newton-Raphson considerando cuerdas en lugar de derivadas

3.5 Métodos de aceleración y optimización de la convergencia

- Método de Steffensen aplicado a Newton-Raphson de varias variables:
Consiste en aproximar las derivadas parciales como:

$$\frac{\partial}{\partial x_j} f_i(x_k) \approx \frac{f_i(x_k + h_j e_j) - f_i(x_k)}{h_j} \text{ donde } e_j \in \mathbb{M}_{n \times 1} \text{ tiene todas sus entradas nulas}$$

excepto en el lugar j que tiene un 1. Este método considera $h_j = f_j(x_k)$

- Extrapolación de Aitken:

Sirve para acelerar la convergencia de un método cuando $\delta_i = \frac{x_i - x_{i-1}}{x_{i-1} - x_{i-2}} \Rightarrow$

$$\text{La extrapolación obtenida será } x_{i+1} = x_i - \frac{(x_i - x_{i-1})^2}{x_i - 2x_{i-1} + x_{i-2}}$$

- Método de regula falsi:

Método lineal cuya utilidad es la de encontrar un intervalo pequeño que contiene la raíz y luego aplicar otro método para hallarla. Este método considera

$$y_{n+1} = \frac{b_n - a_n}{f(b_n) - f(a_n)} \text{ y luego } f(y_{n+1})f(a_n) \text{ es } \begin{cases} < 0 \Rightarrow \begin{cases} a_{n+1} = a_n \\ b_{n+1} = y_{n+1} \end{cases} \\ > 0 \Rightarrow \begin{cases} a_{n+1} = y_{n+1} \\ b_{n+1} = b_n \end{cases} \end{cases}$$

4. Ecuaciones diferenciales ordinarias (EDO)

4.1 Problema de las EDO

- Sistema de ecuaciones a resolver numéricamente:

$$\begin{cases} y' = f(x, y(x)) \\ y(a) = c \end{cases}, \quad x \in [a, b]$$

- Sucesión de la partición:

$$x_n = x_0 + nh \quad \text{donde} \quad \begin{cases} x_0 = a \\ h = \frac{|b-a|}{N} \end{cases} \quad \text{para cierto } N \in \mathbb{N}$$

- Generación de la sucesión solución:

$$y_n / y_n \cong y(x_n)$$

- Notación de solución exacta:

$$\text{Dado } \begin{cases} y' = f(x, y) \\ y(x_0) = c \end{cases} \text{ se nota su solución exacta como } y(x, x_0, c)$$

- Definición de error local:

$$\varepsilon_L = y_{n+1} - y(x_{n+1}, x_n, y_n)$$

- Definición de error global:

$$E_{n+1} = y_{n+1} - y(x_{n+1}, x_0, c)$$

- Definición de error local de truncamiento:

$$\tilde{y}(x) \text{ solución de } \begin{cases} y' = f(x, y) \\ y(x_n) = y_n \end{cases} \text{ en } [x_n, x_{n+1}] \Rightarrow \varepsilon_T(h) = \tilde{y}(x_{n+1}) - y_{n+1}$$

- Definición de consistencia y orden de consistencia:

$$\text{Un método es consistente} \Leftrightarrow \frac{1}{h} \max |\varepsilon_T(h)| \xrightarrow{h \rightarrow 0} 0 \text{ y su orden es el del infinitésimo } \varepsilon_T(h)$$

- Definición de estabilidad de un método:

$$\text{Un método es estable} \Leftrightarrow \exists \alpha / |\overline{E_n}| \leq \alpha \quad \forall n \geq 0$$

- Definición del problema test:

$$\begin{cases} y' = qy \\ y(0) = 1 \end{cases}$$

- Definición de región de estabilidad:

Son los puntos del plano complejo $z = hq/y_n$ aplicado al problema test están acotados

4.2 Método de Euler hacia adelante

- Recurrencia del método³³:

$$\begin{cases} y_{n+1} = y_n + hf(x_n, y_n) \\ y_0 = c \end{cases}$$

³³ Una forma de mejorar la aproximación es usar la recurrencia

$$y_{n+1} = y_n + f(x_n, y_n)h + \frac{f'(x_n, y_n)h^3}{3!}$$

- Error local de truncamiento³⁴:

$$|\varepsilon_T(h)| \leq \frac{h^2 \|\tilde{y}\|_{\infty, [a, b]}}{2}$$

- Error de redondeo:

$$\varepsilon_R(h) \leq \varepsilon_M \|\tilde{y}\|_{\infty, [a, b]}$$

- Error global³⁵:

$$\text{Si } \left| \frac{\partial}{\partial y} f(x, y) \right| \leq L, \quad \varepsilon_n \leq \varepsilon \Rightarrow \begin{cases} E_n \leq \varepsilon \frac{e^{hL} - 1}{e^{hL} - 1} \\ L \neq 0, \quad |Lh| \ll 1 \Rightarrow E_n \leq \frac{\varepsilon}{h} \frac{e^{nhL} - 1}{L} \\ L = 0 \Rightarrow E_n \leq n\varepsilon \end{cases}$$

- Región de estabilidad:

$$z \in \mathbb{C} / \|1 + z\| \leq 1$$

4.3 Método del trapecio

- Recurrencia del método³⁶:

$$\begin{cases} y_{n+1} = y_n + \frac{1}{2} h (f(x_n, y_n) + f(x_{n+1}, y_{n+1})) \\ y_0 = c \end{cases}$$

- Región de estabilidad:

$$z \in \mathbb{C} / \operatorname{Re} z < 0$$

4.4 Método de Euler hacia atrás

- Recurrencia del método³⁷:

$$\begin{cases} y_{n+1} = y_n + hf(x_{n+1}, y_{n+1}) \\ y_0 = c \end{cases}$$

- Sobre las ecuaciones no lineales:

Cuando no es posible hallar y_{n+1} explícitamente puede comenzarse aplicando el método de Euler hacia adelante: $y_{n+1}^{(0)} = y_n + hf(x_n, y_n)$ y luego hallar el punto fijo con el método de Euler hacia atrás: $y_{n+1}^{(k+1)} = y_n + hf(x_{n+1}, y_{n+1}^{(k)})$

- Región de estabilidad:

$$z \in \mathbb{C} / 1 \leq \|1 - z\|$$

³⁴ $\|\tilde{y}\|_{\infty, [a, b]}$ nota el máximo de la derivada en $[a, b]$

³⁵ Así, $E_{n+1}(h) \leq c_1 h + \frac{c_2}{h}$ con $\begin{cases} c_1 = \frac{1}{2} c \|\tilde{y}\|_{\infty, [a, b]} \\ c_2 = \varepsilon_M c \|\tilde{y}\|_{\infty, [a, b]} \end{cases} \Rightarrow h_{\text{óptimo}} = \sqrt{\frac{c_2}{c_1}}$

³⁶ En el caso en que y_{n+1} no puede despejarse puede usarse el MIG, una iteración de punto fijo.

³⁷ Se estima la derivada en el punto de llegada en lugar de el de salida.

4.5 Método del punto medio

- Recurrencia del método:

$$\begin{cases} y_{n+1} = y_{n-1} + 2hf(x_n, y_n) \\ y_0 = c \\ y_1 = d \end{cases}$$

- Región de estabilidad:

$$ai \in \mathbb{R}/a \in (-1, 1)$$

4.6 Métodos de Runge-Kutta

- Recurrencia de los métodos³⁸:

$$\begin{cases} y_{n+1} = y_n + hf(\xi_n, y(\xi_n)) & \text{con } \xi_n = x_n + h\theta_n \\ y_0 = c \end{cases}$$

- Recurrencia equivalente genérica³⁹:

$$\begin{cases} y_{n+1} = y_n + \sum_{i=1}^v w_i k_i & \text{con } k_i = hf\left(x_n + c_i h, y_n + \sum_{j=1}^{i-1} a_{ij} k_j\right) \\ y_0 = c \end{cases} \text{ donde } w_i, c_i, a_{ij} \text{ son arbitrarios}$$

³⁸ Considerando $\theta = \frac{1}{2}$ se obtiene el paso $\frac{1}{2}h$ y con él la aproximación de Runge-Kutta resultado del método explícito de un paso: $y_{n+1} = y_n + hf\left(x_n + \frac{1}{2}h, y_n + \frac{1}{2}hf(x_n, y_n)\right)$

³⁹ Ejemplos:

- Método de Runge-Kutta de segundo orden:

$$\begin{cases} k_1 = hf(x_n, y_n) \\ k_2 = hf(x_n + \frac{1}{2}h, y_n + \frac{1}{2}k_1) \\ y_{n+1} = y_n + \frac{1}{2}(k_1 + k_2) \\ y_0 = c \end{cases}$$

- Método de Runge-Kutta de cuarto orden:

$$\begin{cases} k_1 = hf(x_n, y_n) \\ k_2 = hf(x_n + \frac{1}{2}h, y_n + \frac{1}{2}k_1) \\ k_3 = hf(x_n + \frac{1}{2}h, y_n + \frac{1}{2}k_2) \\ k_4 = hf(x_n + h, y_n + k_3) \\ y_{n+1} = y_n - \frac{1}{6}(k_1 + 2k_2 + 2k_3 + k_4) \\ y_0 = c \end{cases}$$

5. Mínimos cuadrados

5.1 Problema de mínimos cuadrados (PMC)⁴⁰

- Descripción del problema:

Se intenta aproximar el par (t_j, b_j) de forma polinomial $f(t_j) = \sum_{i=0}^n x_i t_j^i \quad \forall j = 1, \dots, m$.

$$\text{Para ello se define } A = \begin{pmatrix} t_1^n & t_1^{n-1} & \dots & 1 \\ \vdots & \vdots & \ddots & \vdots \\ t_m^n & t_m^{n-1} & \dots & 1 \end{pmatrix}, \quad X = \begin{pmatrix} x_1 \\ \vdots \\ x_n \end{pmatrix}, \quad b = \begin{pmatrix} b_1 \\ \vdots \\ b_m \end{pmatrix}$$

Y se intentará minimizar los residuos

- PMC:

$$\min_{x \in \mathbb{R}^n} \|b - Ax\| \text{ con } A \in \mathbb{M}_{m \times n}, \quad m \geq n, \quad b \in \mathbb{R}^m$$

- Sistema de ecuaciones normales⁴¹:

$$A \in \mathbb{M}_{m \times n}, \quad m \geq n, \quad b \in \mathbb{R}^m, \quad S = [A^{(1)}, \dots, A^{(n)}], \quad x \in \mathbb{R}^n \text{ solución del PMC,}$$

$$y = Ax \text{ proyección ortogonal de } b \text{ en } S \Rightarrow A^t Ax = A^t b$$

- Solución al problema de mínimos cuadrados:

$$x = (A^t A)^{-1} A^t b$$

- Unicidad de las soluciones:

$$\text{rango}(A^t A) \begin{cases} = n \Rightarrow \text{La solución es única} \\ < n \Rightarrow \text{Hay más de una solución} \end{cases}$$

5.2 Descomposición QR

- Definición de Q y R⁴²:

$$Q \in \mathbb{M}_{m \times m} \text{ ortogonal, y } R \in \mathbb{M}_{m \times n} : R = \begin{pmatrix} R_1 \\ 0 \end{pmatrix} \text{ con } R_1 \in \mathbb{M}_{n \times n} \text{ triangular y } 0 \in \mathbb{M}_{(m-n) \times n}$$

- Existencia de la descomposición QR:

$$\text{Si } A \in \mathbb{M}_{m \times n} \text{ con } m \geq n \Rightarrow \exists Q, R / A = QR$$

- Aplicación de la descomposición QR al PMC:

$$\min_{x \in \mathbb{R}^n} \|Ax - b\| = \min_{x \in \mathbb{R}^n} \|Rx - Q^t b\| = \min_{x \in \mathbb{R}^n} \|R_1 x - d_1\| \text{ con } d = Q^t b = \begin{pmatrix} d_1 \\ d_2 \end{pmatrix} \text{ y } \begin{cases} d_1 \in \mathbb{R}^n \\ d_2 \in \mathbb{R}^{m-n} \end{cases} \Rightarrow R_1 x = d_1$$

⁴⁰ En los siguientes métodos de existir más de una solución se elige la de norma mínima

⁴¹ Surge de aplicar al PMC el siguiente teorema:

S subespacio vectorial de \mathbb{R}^n , $b \in \mathbb{R}^m$ y $P_S(b)$ una proyección ortogonal sobre S \Rightarrow

$$\Rightarrow \begin{cases} \exists! y = P_S(b) / (b - y) \in S^\perp \\ y = P_S(b) / \|b - y\| = \min_{s \in S} \|b - s\| \end{cases}$$

⁴² En Matlab $x = A \backslash b$ da la solución de mínimos cuadrados calculada con descomposición QR

5.3 Método de la descomposición en valores singulares (SVD)

- Teorema de SVD⁴³:

$A \in \mathbb{M}_{m \times n}(\mathbb{R})$, $\text{rango} A = r \Rightarrow \exists$ una descomposición de A de la forma $A = U \Sigma V^t$

$$\text{con } \begin{cases} U \in \mathbb{M}_{m \times m}(\mathbb{R}) \text{ ortogonal} \\ \Sigma = \begin{pmatrix} \Sigma_r & 0 \\ 0 & 0 \end{pmatrix} \text{ con } \Sigma_r = \begin{pmatrix} \sigma_1 & & 0 \\ & \ddots & \\ 0 & & \sigma_r \end{pmatrix} \in \mathbb{M}_{r \times r}(\mathbb{R}) \text{ con } \sigma_i \geq \sigma_j \geq 0 \quad \forall i < j \\ V \text{ ortogonal} \end{cases}$$

- Teorema de solución al PMC⁴⁴:

La solución viene dada por $\dot{x} = V \begin{pmatrix} \Sigma_r^{-1} & 0 \\ 0 & 0 \end{pmatrix} U^t b$

5.4 Mínimos cuadrados no lineales y método de Gauss-Newton

- Problema objetivo:

Si se intentara ajustar (t_i, y_i) a una función $f(x, t)$, lo que se deberá hallar es

$$\min_{x \in \mathbb{R}^n} \sum_{i=1}^m r_i^2(x) \text{ con } r_i(x) = y_i - f(x, t_i)$$

- Método de Gauss-Newton⁴⁵:

En el k -ésimo paso se tiene el punto x_k y se aproxima $r(x)$ como

$\tilde{r}_k(x) = r(x_k) + J_r(x_k)(x - x_k)$. Para hallar x_{k+1} se procede como:

1. Se resuelve el PMC: $\min_{\rho \in \mathbb{R}^n} \|r(x_k) + J_r(x_k)\rho\|$
2. Se considera $x_{k+1} = x_k + \rho$ y se retorna a 1. de no cumplirse la condición de parada

- Unicidad de la solución del método de Gauss-Newton:

$$\text{Si } \text{rango} J_r(x_k) \begin{cases} = n \text{ se tiene una única solución dada por} \\ \rho = -(J_r^t(x_k) J_r(x_k))^{-1} J_r^t(x_k) r(x_k) \\ < n \text{ se considera } \hat{\rho} \text{ dada por la SVD de } J_r(x_k) \end{cases}$$

⁴³ Los σ_i son conocidos como valores singulares

⁴⁴ En las hipótesis del teorema de SVD

⁴⁵ Las posibles condiciones de parada son $\begin{cases} k > Iter \\ \|r(x_k)\| < \varepsilon_1 \\ \|x_k - x_{k+1}\| < \varepsilon_2 \end{cases}$

6. Interpolación

6.1 Problema de la interpolación

- Objetivo:

Dados $\{(x_i, y_i), i = 0, \dots, n / f(x_i) = y_i\}$ encontrar $P_n(x) / P_n(x_i) = y_i \quad \forall i = 0, \dots, n$

- Matriz de Vermonde⁴⁶:

$$X\alpha = \begin{pmatrix} 1 & x_0 & \dots & x_0^n \\ \vdots & \vdots & \ddots & \vdots \\ 1 & x_n & \dots & x_n^n \end{pmatrix} \begin{pmatrix} \alpha_1 \\ \vdots \\ \alpha_n \end{pmatrix} = \begin{pmatrix} y_0 \\ \vdots \\ y_n \end{pmatrix} = y$$

- Error al interpolar:

$$\text{Si } x \in [a, b] \Rightarrow E(x) = f(x) - P_n(x) \leq \frac{\|f^{(n+1)}\|_{[a,b]} (b-a)^{n+1}}{(n+1)!}$$

- Fenómeno de Rounge:

A medida que se agregan más puntos el error no está acotado porque $\|f^{(n+1)}\|_{[a,b]}$ crece más rápidamente que $(n+1)!$

6.2 Interpolación de Lagrange

- Polinomio interpolante:

Se pedirá que los coeficientes de los polinomios cumplan que

$$l_j(x_i) = \begin{cases} 0 & \text{si } i \neq j \\ 1 & \text{si } i = j \end{cases} \Rightarrow P_n(x) = \sum_{j=0}^n y_j l_j(x)$$

- Coefficientes del polinomio interpolante:

$$l_j(x) = \frac{\prod_{i \neq j} (x - x_i)}{\prod_{i \neq j} (x_j - x_i)}$$

- Desventaja:

Cada vez que se agrega un nuevo punto es necesario recalcular todos los coeficientes del polinomio interpolante

6.3 Interpolación de Newton

- Ventaja:

No necesita recalcular todos los coeficientes del polinomio interpolante cuando se agrega el k -ésimo punto

- Polinomio interpolante:

$$\begin{cases} P_k(x) = P_{k-1}(x) + q_k(x) & \forall i = 0, \dots, k \\ P_k(x_i) = f(x_i) & \forall k = 0, \dots, n \end{cases} \text{ con } q_k(x) = a_k \prod_{i=0}^{k-1} (x - x_i)$$

$$\text{con } a_k = \frac{f(x_k) - P_{k-1}(x_k)}{\prod_{i=0}^{k-1} (x_k - x_i)}$$

⁴⁶ Se obtiene expresando matricialmente el problema de la interpolación

- Notación recursiva:

Se define $a_k = f[x_0, \dots, x_k]$, por lo que se tiene

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

$$= \sum_{i=0}^n \left(\prod_{j=0}^{i-1} (x - x_j) \right) f[x_0, \dots, x_i] \quad \text{donde } f[x_i, \dots, x_k] = \frac{f[x_{i+1}, \dots, x_k] - f[x_i, \dots, x_{k-1}]}{x_k - x_i}$$

6.4 Interpolación de Hermite

- Objetivo:

Encontrar $P(x)$ tal que $P(x_i) = f(x_i)$ y $P'(x_i) = f'(x_i)$ con $f(x_i), f'(x_i)$ datos dados

- Polinomio de Hermite:

$$H_{2n+1}(x) = \sum_{i=0}^n y_i h_i(x) + \sum_{i=0}^n y'_i \bar{h}_i(x) \quad \text{donde:}$$

$$\begin{cases} h_i(x_j) = \begin{cases} 1 & \text{si } i \neq j \\ 0 & \text{si } i = j \end{cases} ; & \frac{\partial}{\partial x} h_i(x_j) = 0 \quad \forall j = 0, \dots, n \\ \bar{h}_i(x_j) = 0 \quad \forall j = 0, \dots, n ; & \frac{\partial}{\partial x} \bar{h}_i(x_j) = \begin{cases} 1 & \text{si } i = j \\ 0 & \text{si } i \neq j \end{cases} \end{cases}$$

- Coeficientes del polinomio interpolante usando los coeficientes del polinomio interpolante resultado de la interpolación de Lagrange:

$$h_i(x) = (1 - 2l'_i(x)(x - x_i))l_i^2(x)$$

$$\bar{h}_i(x) = (x - x_i)^2 l_i^2(x)$$

6.5 Interpolación a trozos⁴⁷

- Interpolación lineal a trozos:

Utiliza las cuerdas entre los puntos dados:

$$L(x) = y_i \frac{x - x_{i+1}}{x_i - x_{i+1}} + y_{i+1} \frac{x - x_i}{x_{i+1} - x_i}, \quad x \in [x_i, x_{i+1}]$$

- Error:

$$|f(x) - L(x)| \leq \|f''\|_{\infty} \frac{h^2}{8}$$

- Planteo:

$$\text{Hallar } S_i(x) \text{ en } [x_i, x_{i+1}] \text{ tal que } \begin{cases} S_i(x_i) = y_i & S_i(x_{i+1}) = y_{i+1} \\ S'_i(x_i) = m_i & S'_i(x_{i+1}) = m_{i+1} \end{cases}$$

- Forma de la solución:

$$S_i(x) = y_i L_i^{(0)}(x) + y_{i+1} L_i^{(1)}(x) + m_i L_i^{(2)}(x) + m_{i+1} L_i^{(3)}(x)$$

- Interpolación Hermite cúbico⁴⁸:

$$m_i = \frac{y_{i+1} - y_{i-1}}{x_{i+1} - x_{i-1}}; \quad m_0 = \frac{y_1 - y_0}{x_1 - x_0}; \quad m_n = \frac{y_n - y_{n-1}}{x_n - x_{n-1}}$$

⁴⁷ Estos estiman entre un par de puntos una función interpolante. De esta forma no se tiene un solo polinomio que interpola todos los puntos, sino, uno por cada par de puntos del conjunto.

⁴⁸ Los siguientes métodos de interpolación estiman los $m_i = y'_i = f'(x_i)$ en caso de que estos no sean dados

- Interpolación de spline cúbico⁴⁹:

$$m_0 = \frac{y_1 - y_0}{x_1 - x_0}; \quad m_n = \frac{y_n - y_{n-1}}{x_n - x_{n-1}}. \text{ Los } m_i \text{ deben despejarse del sistema generado por}$$

$$\text{los } i = 1, \dots, n-1 \text{ y compuestos por las ecuaciones } \frac{1}{2}m_{i-1} + 2m_i + \frac{1}{2}m_{i+1} = 3\left(\frac{y_{i+1} - y_{i-1}}{2h}\right)$$

6.6 Curvas de Bézier

- Definición del Polinomio de Bézier:

$$P_n(x) = \sum_{i=0}^n p_i B_i^n(x) \text{ donde los } B_i^n(x) \text{ es el } n\text{-ésimo polinomio de Bernstein y cada } p_i$$

es un punto (llamado de control) dado

- Definición iterativa del polinomio de Bernstein⁵⁰:

$$B_i^n(x) = \frac{(b-x)^{n-1}(x-a)^i}{(b-a)^n} \frac{n!}{i!(n-i)!}$$

- Definición del Polinomio de Bézier vectorial (paramétrico):

$$\overline{P}_n(t) = \sum_{i=0}^n \overline{p}_i B_i^n(t) \text{ donde los } B_i^n(t) \text{ es el } n\text{-ésimo polinomio de Bernstein y cada}$$

\overline{p}_i es un conjunto de puntos (llamados de control) dados.

⁴⁹ Este método impone que la derivada segunda sea continua: $S'_{i-1}(x_i) = S'_i(x_i)$

⁵⁰ Definición recursiva del polinomio de Bernstein:

$$\begin{cases} B_{-1}^n(x) = B_{n+1}^n(x) = 0 \\ B_1^1(x) = 1 \\ B_i^n(x) = \frac{(b-x)B_i^{n-1}(x) + (x-a)B_{i-1}^{n-1}(x)}{b-a} \end{cases}$$