
CÁLCULO NUMÉRICO

ApuntsFME

BARCELONA, OCTUBRE 2018

Última modificación: 24 de octubre de 2018.

This work is licensed under a [Creative Commons](#) “[Attribution-NonCommercial-ShareAlike 4.0 International](#)” license.



Contenidos

| | |
|--|-----------|
| 1. Ceros de funciones | 1 |
| 1.1. Métodos clásicos | 1 |
| 1.1.1. Método de la bisección | 1 |
| 1.1.2. Método de Newton | 1 |
| 1.1.3. Método de la secante | 2 |
| 1.2. Consistencia y convergencia | 2 |
| 1.2.1. Análisis del método de Newton | 5 |
| 1.2.2. Gráficos de convergencia | 6 |
| 1.2.3. Criterios prácticos de convergencia | 6 |
| 1.3. Métodos híbridos | 7 |
| 2. Sistemas de ecuaciones no lineales | 9 |
| 2.1. Método de Newton-Raphson | 9 |
| 2.2. Métodos casi-Newton | 10 |
| 2.2.1. Método de Broyden | 10 |
| 3. Aproximación funcional | 11 |
| 3.1. Interpolación polinómica | 11 |
| Teorema del residuo de Lagrange | 13 |
| 3.2. Interpolación de Splines | 14 |
| 3.2.1. Splines lineales ($m = 1$) \mathcal{C}^0 | 14 |
| 3.2.2. Splines cuadráticos ($m = 2$) \mathcal{C}^1 | 14 |
| 3.2.3. Splines cúbicos ($m = 3$) \mathcal{C}^1 | 15 |
| 3.2.4. Splines cúbicos ($m = 3$) \mathcal{C}^2 | 16 |
| Teorema de la suavidad del Spline natural | 17 |
| 3.3. Mínimos cuadrados | 18 |
| Índice alfabético | 19 |

Tema 1

Ceros de funciones

Los métodos numéricos nos proporcionan aproximaciones sucesivas de una raíz (α) de una función. Comenzaremos con una aproximación inicial x^0 y el método nos proporcionará x^1 . Más generalmente, dado un x^k , el método nos proporcionará x^{k+1} .

1.1. Métodos clásicos

1.1.1. Método de la bisección

El método está basado en el teorema de Bolzano. Consiste en tomar un intervalo (a, b) con $f(a)f(b) < 0$. Una vez encontrado este intervalo inicial, vamos dividiendo sucesivamente el intervalo por la mitad y tomando el subintervalo que cumpla la condición de que sus extremos tengan distinto signo. Reduciendo el error a la mitad.

Este método tiene la ventaja de ser muy robusto y nos permite calcular la raíz con una precisión arbitraria. El inconveniente principal de este método es la velocidad: es un método muy lento.

1.1.2. Método de Newton

Dada una aproximación de la raíz x^k , calcularemos la siguiente aproximación x^{k+1} como el cero de la recta tangente a f en x^k , es decir, aproximaremos $f(x) \approx f(x^k) + f'(x^k)(x - x^k)$ y x^{k+1} será el 0 de esta función aproximada:

$$x^{k+1} = x^k - \frac{f(x^k)}{f'(x^k)}.$$

La ventaja de este método es la velocidad (tiene convergencia cuadrática para raíces simples). El inconveniente principal es que no es tan robusto: necesita una aproximación inicial suficientemente buena. Además, este método requiere de conocer la derivada $f'(x)$, lo cual puede ser costoso o imposible.

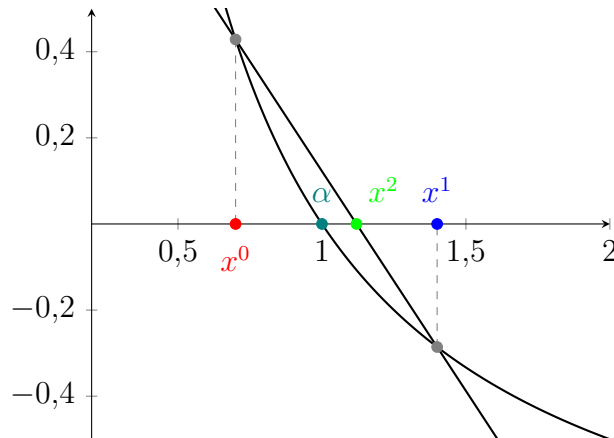


1.1.3. Método de la secante

Este método parte de dos aproximaciones iniciales x^0 y x^1 , tomaremos x^2 como el cero de la recta que pasa por x^0 y x^1 (substituye la derivada por la recta secante). Es decir, dados x^k y x^{k-1} ,

$$x^{k+1} = x^k - f(x^k) \frac{x^k - x^{k-1}}{f(x^k) - f(x^{k-1})}.$$

Las ventajas e inconvenientes de este método son casi idénticas a las del método de Newton. Este método es un poco más lento que el de Newton, aunque converge bastante rápidamente y, igual que el de Newton, el punto flaco es que no es tan robusto como el de la bisección.



Observación 1.1.1. El coste computacional se mide en número de evaluaciones de $f(x)$ (y de $f'(x)$). El método de la secante necesita 1 evaluación por iteración y por lo tanto, necesita menos que el método de Newton.

1.2. Consistencia y convergencia

Definición 1.2.1. Diremos que un método $x^{k+1} = \phi(x^k)$ es consistente si las raíces son los únicos puntos fijos, es decir,

$$f(\alpha) = 0 \iff \phi(\alpha) = \alpha.$$

Ejercicio 1.2.2. Un ejemplo es el método de Newton.

Si α es una raíz simple,

$$f(\alpha) = 0 \implies \phi(\alpha) = \alpha - \frac{f(\alpha)}{f'(\alpha)} = \alpha.$$

Por otro lado,

$$\phi(\alpha) = \alpha \implies \frac{f(\alpha)}{f'(\alpha)} = 0 \implies f(\alpha) = 0.$$

Con la raíz doble, no podemos evaluar el método de Newton en α , pero por la regla de l'Hôpital llegaríamos a la conclusión de que, tomando el límite de ϕ se sigue cumpliendo.

Análogamente, podemos deducir que el método de Newton es un método consistente para una raíz α de cualquier multiplicidad.

Definición 1.2.3. Diremos que un esquema es convergente si

$$\lim_{k \rightarrow \infty} x^k = \alpha.$$

Definición 1.2.4. Llamaremos error absoluto de la aproximación a

$$E^k = |x^k - \alpha|$$

y es equivalente que el método sea convergente a que $\lim_{k \rightarrow \infty} E^k = 0$.

Definición 1.2.5. Diremos que un esquema tiene convergencia lineal si

$$|E^{k+1}| \leq \lambda |E^k|, \quad \lambda < 1.$$

Definición 1.2.6. Diremos que un esquema tiene convergencia de orden p si

$$|E^{k+1}| \leq \lambda |E^k|^p.$$

Definición 1.2.7. Diremos que un esquema tiene convergencia superlineal si

$$|E^{k+1}| \leq \lambda_k |E^k|, \quad \lambda_k \rightarrow 0$$

o bien

$$|E^{k+1}| \leq \lambda |E^k|^q, \quad q \in (1, 2).$$

Proposición 1.2.8. Para estudiar la convergencia, en general, necesitamos estudiar E^{k+1} en función de E^k . Para ello, podemos usar la siguiente fórmula:

$$E^{k+1} = \phi'(\alpha)E^k + \frac{1}{2}\phi''(\alpha)(E^k)^2 + \dots$$

o más generalmente

$$E^{k+1} = \sum_{i=1}^{\infty} \frac{1}{i!} \phi^{(i)}(\alpha) (E^k)^i.$$

Demostración. Tenemos que

$$E^{k+1} = x^{k+1} - \alpha = \phi(x^k) - \phi(\alpha).$$

Aproximando ahora por Taylor (centrado en α):

$$E^{k+1} = \cancel{\phi(\alpha)} + \cancel{\phi(\alpha)} + \phi'(\alpha)(x^k - \alpha) + \dots$$

substituyendo ahora $x^k - \alpha = E^k$, obtenemos la fórmula original. \square

Proposición 1.2.9. Si $\phi'(\alpha) \neq 0$ y $|\phi'(\alpha)| < 1$, entonces el método converge linealmente.

Demostración. Veremos ahora que la convergencia es lineal:

$$|E^{k+1}| \leq |E^k| \left(|\phi'(\alpha)| + \underbrace{\left| \frac{1}{2} \phi''(\alpha) E^k + \mathcal{O}\left((E^k)^2\right) \right|}_{\leq \varepsilon} \right).$$

Consideramos ε tal que $|\phi'(\alpha)| + \varepsilon \leq 1$. Esto pasa si $|E^k|$ es suficientemente pequeño. Entonces

$$|E^{k+1}| \leq \lambda |E^k|,$$

con $\lambda = |\phi'| + \varepsilon < 1$. \square

Proposición 1.2.10. Si $\phi'(\alpha) = \dots = \phi^{(p-1)}(\alpha) = 0$ y $\phi^{(p)}(\alpha) \neq 0$. Tenemos que

$$E^{k+1} = \frac{1}{p!} \phi^{(p)}(\alpha) (E^k)^p + \frac{1}{(p+1)!} \phi^{(p+1)}(\alpha) (E^k)^{p+1} + \dots$$

Entonces, para E^k pequeño, se tiene que

$$|E^{k+1}| \approx \left| \frac{\phi^{(p)}(\alpha)}{p!} \right| (E^k)^p.$$

y la convergencia es de orden p .

Demostración. Se tiene que

$$|E^{k+1}| \leq |E^k|^p \left(\left| \frac{\phi^{(p)}(\alpha)}{p!} \right| + \underbrace{\left| \frac{\phi^{(p+1)}(\alpha)}{(p+1)!} (E^k) + \mathcal{O}\left((E^k)^2\right) \right|}_{\leq \varepsilon} \right).$$

Con $|E^k|$ suficientemente pequeño

$$|E^{k+1}| \leq |E^k|^p \underbrace{\left(\frac{\phi^{(p)}(\alpha)}{p!} + \varepsilon \right)}_{\lambda}.$$

\square

Definición 1.2.11. Sea ϕ un método lineal, por lo tanto,

$$|E^{k+1}| \leq \lambda |E^k|.$$

Llamaremos factor asintótico de convergencia a λ .

1.2.1. Análisis del método de Newton

Para estudiar la convergencia, primero es conveniente, conocer las derivadas de la función ϕ , así pues, tenemos

$$\begin{aligned}\phi(x) &= x - \frac{f(x)}{f'(x)}, \\ \phi'(x) &= \frac{f(x)f''(x)}{f'(x)^2}, \\ \phi''(x) &= \frac{f''(x)}{f'(x)} - \frac{2f(x)f''(x)^2}{f'(x)^3} + \frac{f(x)f'''(x)}{f'(x)^2}.\end{aligned}$$

- Caso de una raíz simple, $f(\alpha) = 0$ y $f'(\alpha) \neq 0$. En este caso,

$$\begin{aligned}\phi'(\alpha) &= 0, \\ \phi''(\alpha) &= \frac{f''(\alpha)}{f'(\alpha)}.\end{aligned}$$

En general $\phi''(\alpha) \neq 0$, por lo tanto el método de Newton tiene, en general, convergencia de orden 2.

- Caso de una raíz doble $f(\alpha) = f'(\alpha) = 0$ y $f''(\alpha) \neq 0$. Observamos que $\phi'(\alpha)$ no está bien definido, pero podemos calcular el límite, teniendo

$$\begin{aligned}\lim_{x \rightarrow \alpha} \phi'(x) &= \lim_{x \rightarrow \alpha} \frac{f(x)f''(x)}{f'(x)^2} \stackrel{\text{R.H.}}{=} \lim_{x \rightarrow \alpha} \frac{f'(x)f''(x) + f(x)f'''(x)}{2f'(x)f''(x)} = \\ &= \lim_{x \rightarrow \alpha} \frac{f'(x)f''(x)}{2f'(x)f''(x)} + \frac{f(x)f'''(x)}{2f'(x)f''(x)} = \frac{1}{2} + \lim_{x \rightarrow \alpha} \frac{f(x)f'''(x)}{2f'(x)f''(x)} \stackrel{\text{R.H.}}{=} \\ &= \frac{1}{2} + \lim_{x \rightarrow \alpha} \frac{f'(x)f'''(x) + f(x)f^{(4)}(x)}{2f''(x)^2 + 2f'(x)f'''(x)} = \frac{1}{2}.\end{aligned}$$

Por lo tanto es lineal con factor asintótico de convergencia igual a $\frac{1}{2}$.

- El caso de multiplicidad p , es decir, $f(\alpha) = \dots = f^{(p-1)}(\alpha) = 0$ y $f^{(p)} \neq 0$. Para resolver este caso, aplicaremos $2p - 2$ veces la regla de l'Hôpital

$$\begin{aligned}\lim_{x \rightarrow \alpha} \phi'(x) &= \lim_{x \rightarrow \alpha} \frac{f(x)f''(x)}{f'(x)f'(x)} \stackrel{\text{Leibniz}}{\stackrel{\text{R.H.}}{=}} \lim_{x \rightarrow \alpha} \frac{\sum_{k=0}^{2p-2} \binom{2p-2}{k} f^{(2p-k)}(x) f^{(k)}(x)}{\sum_{k=0}^{2p-2} \binom{2p-2}{k} f^{(2p-1-k)}(x) f^{(k+1)}(x)} = \\ &= \lim_{x \rightarrow \alpha} \frac{\binom{2p-2}{p} \cancel{f^{(p)}(x)} \cancel{f^{(p)}(x)}}{\binom{2p-2}{p-1} \cancel{f^{(p)}(x)} \cancel{f^{(p)}(x)}} = \frac{\binom{2p-2}{p}}{\binom{2p-2}{p-1}} = \frac{(p-1)!(p-1)!}{p!(p-2)!} = \\ &= \frac{p-1}{p} = 1 - \frac{1}{p}.\end{aligned}$$

El método es lineal con F.A.C. igual a $1 - \frac{1}{p}$.

1.2.2. Gráficos de convergencia

Definición 1.2.12. Definimos el error relativo como

$$r^k = \frac{E^k}{\alpha} = \frac{x^k - \alpha}{\alpha}.$$

Observación 1.2.13. x^k tiene q cifras significativas correctas si $|r^k| \leq \frac{1}{2}10^{-q}$.

Caso de convergencia lineal

$$\begin{aligned} |E^{k+1}| &\approx \lambda |E^k|, \lambda = F.A.C. \stackrel{x^{k+1} \approx \phi(x^k)}{=} |\phi'(\alpha)|, \\ \frac{|E^k|}{|\alpha|} &\approx \lambda^k \frac{|E^0|}{|\alpha|} \implies |r^k| \approx \lambda^k |r^0|. \end{aligned}$$

Si la convergencia es de orden $p \geq 2$,

$$\begin{aligned} |E^k| &\approx \lambda \left(|E^{k-1}| \right)^p \approx \lambda \left(\lambda |E^{k-2}|^p \right)^p = \lambda^{1+p} |E^{k-2}|^{p^2} \approx \lambda^{p+1} \left(\lambda |E^{k-3}|^p \right)^{p^2} = \dots \\ &\approx \lambda^{1+p+\dots+p^{k-1}} |E^0|^{p^k} \stackrel{\text{serie geométrica}}{=} \lambda^{\frac{p^k-1}{p-1}} |E^0|^{p^k}, \\ \frac{|E^k|}{|\alpha|} &\approx \lambda^{\frac{p^k-1}{p-1}} \frac{|E^0|^{p^k}}{|\alpha|^{p^k}} |\alpha|^{p^k-1}, \\ |r^k| &\approx A^{p^k} \cdot B, \\ \log |r^k| &\approx \log B + p^k \log A, \text{ donde } A = \lambda^{\frac{1}{p-1}} |r^0| |\alpha|, B = \lambda^{\frac{-1}{p-1}} |\alpha|^{-1}. \end{aligned}$$

Ejemplo 1.2.14. El error nos proporciona información sobre el tipo de convergencia.

| | Bisección (lineal $\lambda = 1/2$) | Newton (cuadrática) |
|---------|--|------------------------|
| $ r^0 $ | $0,1 \cdot 10^{-1}$ | $0,1 \cdot 10^{-1}$ |
| $ r^1 $ | $0,5 \cdot 10^{-2}$ | $0,1 \cdot 10^{-2}$ |
| $ r^2 $ | $0,25 \cdot 10^{-2}$ | $0,1 \cdot 10^{-4}$ |
| $ r^3 $ | $0,12 \cdot 10^{-2}$ | $0,1 \cdot 10^{-8}$ |

1.2.3. Criterios prácticos de convergencia

Si queremos q cifras significativas correctas (C.S.C.), pararemos el algoritmo cuando

$$\frac{1}{2}10^{-q} > |r^k| = \frac{|x^k - \alpha|}{|\alpha|} \stackrel{\alpha \approx x^{k+1}}{\approx} \frac{|x^k - x^{k+1}|}{|x^{k+1}|}.$$

En caso que $\alpha \approx 0$, podemos hacer una variación

$$|x^k - \alpha| < \varepsilon |x^k| + \varepsilon.$$

Conviene comprobar el error en el “residuo” $f(x)$. El criterio puede ser combinado

$$|x^k - \alpha| + |f(x^k)| < \varepsilon_f.$$

1.3. Métodos híbridos

Método de Brent: véase [Brent's method - Wikipedia](#).

Tema 2

Sistemas de ecuaciones no lineales

Dada $\vec{f}: \mathbb{R}^n \rightarrow \mathbb{R}^n$, queremos encontrar $\vec{\alpha} \in \mathbb{R}^n$ tal que $\vec{f}(\vec{\alpha}) = \vec{0}$.

2.1. Método de Newton-Raphson

Dada una aproximación $\vec{x}^k \approx \vec{\alpha}$, consideramos la aproximación lineal de $\vec{f}(\vec{x})$ en \vec{x}^k

$$\vec{f}(\vec{x}) \approx \vec{f}(\vec{x}^k) + \frac{\partial \vec{f}}{\partial \vec{x}}(\vec{x}^k)(\vec{x} - \vec{x}^k) + \mathcal{O}(\|\vec{x} - \vec{x}^k\|^2) = 0,$$
$$\vec{x}^{k+1} = \vec{x}^k - \left[\frac{\partial \vec{f}}{\partial \vec{x}}(\vec{x}^k) \right]^{-1} \vec{f}(\vec{x}^k), \text{ donde } J = \frac{\partial \vec{f}}{\partial \vec{x}} \text{ es la jacobiana.}$$

Se puede ver que tiene convergencia cuadrática (de orden 2) si $\det(J(\alpha)) \neq 0$.

Observación 2.1.1. No calcularemos J^{-1} (porque es aproximadamente el coste de resolver n sistemas lineales). Calculamos:

- i) $J(\vec{x}^k) \Delta \vec{x}^k = -\vec{f}(\vec{x}^k) \longrightarrow \Delta \vec{x}^k,$
- ii) $\vec{x}^{k+1} = \vec{x}^k + \Delta \vec{x}^k.$

En Matlab: $x = x - (J \setminus f)$ resuelve el sistema.

¿Qué hacemos si no podemos calcular $J = \frac{\partial \vec{f}}{\partial \vec{x}}$? *Aproximación de las derivadas:*

$$\frac{\partial \vec{f}}{\partial x_i}(\vec{x}^k) \approx \frac{\vec{f}(\vec{x}^k + h\vec{e}_i) - \vec{f}(\vec{x}^k)}{h} \text{ con } h \text{ pequeña,}$$
$$J = \frac{\partial \vec{f}}{\partial \vec{x}} = \left[\frac{\partial \vec{f}}{\partial x_1}, \dots, \frac{\partial \vec{f}}{\partial x_n} \right].$$

Observación 2.1.2. Si Newton converge pero no con convergencia cuadrática, hay que comprobar que $J = \frac{\partial \vec{f}}{\partial \vec{x}}$ en la implementación.

2.2. Métodos casi-Newton

Si no podemos calcular $J = \frac{\partial \vec{f}}{\partial \vec{x}}$, también podemos usar métodos casi-Newton.

- $\vec{x}^{k+1} = \vec{x}^k - [S^k]^{-1} \vec{f}(\vec{x}^k)$, donde $S^k \approx J(\vec{x}^k)$.
- S^k se calcula a partir de S^{k-1} , con la *ecuación secante* ($S^k \Delta \vec{x}^k = \Delta \vec{f}^k$):

$$\begin{cases} \Delta \vec{x}^k = \vec{x}^k - \vec{x}^{k-1}, \\ \Delta \vec{f}^k = \vec{f}(\vec{x}^k) - \vec{f}(\vec{x}^{k-1}). \end{cases}$$

Observación 2.2.1. En una dimensión, es el método de la secante. $S^k (x^k - x^{k-1}) = f^k - f^{k-1}$, por lo tanto

$$S^k = \frac{f^k - f^{k-1}}{x^k - x^{k-1}}.$$

En n dimensiones, la ecuación secante no determina de manera única S^k . Se añaden otras condiciones que generan muchos métodos casi-Newton.

2.2.1. Método de Broyden

El método de Broyden es un método casi-Newton que impone

- i) $S^k \Delta \vec{x}^k = \Delta \vec{f}^k$ (ecuación secante),
- ii) $(S^k - S^{k-1}) \vec{\omega} = \vec{0}$, $\forall \vec{\omega} \perp \Delta \vec{x}^k$.

Podemos conseguir ii) con $S^k = S^{k-1} + \vec{u} (\Delta \vec{x}^k)^T$. Ahora imponemos i)

$$\Delta \vec{f}^k = S^k \Delta \vec{x}^k = \left(S^{k-1} + \vec{u} (\Delta \vec{x}^k)^T \right) \Delta \vec{x}^k = -\vec{f}^{k-1} + \vec{u} \|\Delta \vec{x}^k\|^2,$$

de donde aislamos

$$\vec{u} \|\Delta \vec{x}^k\|^2 = \Delta \vec{f}^k + \vec{f}^{k-1} = \vec{f}^k \implies \vec{u} = \frac{\vec{f}^k}{\|\Delta \vec{x}^k\|^2}.$$

En resumen,

- Dado $\vec{x}^0 \in \mathbb{R}^n$, $S^0 \in \mathbb{R}^{n \times n}$, ($S_0 = \text{Id}$ ó $S_0 = J(\vec{x}^0)$ aproximación lineal),
- Calcular $\vec{f}^0 = \vec{f}(\vec{x}^0)$,
- Para $k = 0, 1, \dots$,
 - Resolver $S^k \Delta \vec{x}^{k+1} = -\vec{f}^k \implies \Delta \vec{x}^{k+1}$,
 - $\vec{x}^{k+1} = \vec{x}^k + \Delta \vec{x}^{k+1}$,
 - Calcular $\vec{f}^{k+1} = \vec{f}(\vec{x}^{k+1})$,
 - $S^{k+1} = S^k + \vec{f}^{k+1} \left(\frac{\Delta \vec{x}^k}{\|\Delta \vec{x}^k\|^2} \right)^T$.

Tema 3

Aproximación funcional

Tenemos los siguientes datos:

- $f(x)$ analíticamente o evaluable,
- $\{f_i\}_{i=0}^n$ valor de f en $n+1$ puntos $\{x_i\}_{i=0}^n$,
- $\{f_i\}_{i=0}^n, \{f'_i\}_{i=0}^n, \dots$,
- Espai d'aproximació V (espai vectorial).

El objetivo es determinar $p \in V$ t.q. $p(x) \approx f(x)$ (V es el tipo de aproximación y \approx el criterio de aproximación). Tanto si los datos son discretos como si son continuos, podemos decidir diferentes criterios y tipos de aproximación.

Tipos de aproximación:

- Polinómica: $f(x) \approx p_n(x) \in \mathcal{P}^n$, n fixat.
- Trigonométrica: $f(x) \approx \sum_{k=1}^n a_k \sin(nkl) + b_k \cos(nkl)$.
- Splines: Polinomios a trozos.

Criterios de aproximación:

- Interpolación: $p(x_i) = f(x_i)$, $i = 0, \dots, n$.
- Mínimos cuadrados: $p = \arg \min_{q \in V} \|f - q\|^2$, por ejemplo, $\|f - p\|^2 = \sum_{i=0}^n (f_i - p(x_i))^2$.
- Min-max.

3.1. Interpolación polinómica

Teorema 3.1.1. Dados $n+1$ puntos $\{x_i\}_{i=0}^n$ diferentes y $n+1$ valores $\{f_i\}_{i=0}^n$, existe un único polinomio $p_n(x)$ de grado $\leq n$ tal que $p_n(x_i) = f_i$, $i = 0, \dots, n$.

Demostración. Consideramos $p_n(x) = a_0 + a_1x + \dots + a_nx^n$ (en la base natural) e imponemos $f_i = p_n(x_i) = a_0 + a_1x_i + \dots + a_nx_i^n$, $i = 0, \dots, n$.

$$\underbrace{\begin{pmatrix} 1 & x_0 & \dots & x_0^n \\ \vdots & \vdots & & \vdots \\ 1 & x_n & \dots & x_n^n \end{pmatrix}}_A \begin{pmatrix} a_0 \\ \vdots \\ a_n \end{pmatrix} = \begin{pmatrix} f_0 \\ \vdots \\ f_n \end{pmatrix}.$$

A es una matriz de Vandermonde, $\det(A) = \prod_{1 \leq i < j \leq n} (x_i - x_j) \neq 0$ si $\{x_i\}$ son no repetidos y por tanto existe una única solución al sistema. \square

Observación 3.1.2. En la práctica, el determinante no sirve para saber si una matriz es singular. Hay que mirar el rango, el numero de condición (κ_2), etc.

Ejemplo 3.1.3. Si tenemos $\det(A) = 0, 1$, entonces $\det(0, 1 \cdot A) = 10^{-(n+1)}$.

Observación 3.1.4. Las matrices de Vandermonde están muy mal condicionadas

| n | $\kappa_2(A)$ |
|-----|---------------------|
| 2 | 13 |
| 3 | 154 |
| 5 | $5,7 \cdot 10^4$ |
| 10 | $4,5 \cdot 10^{12}$ |

$\kappa_2(A) = \|A\| \cdot \|A^{-1}\|$, si A es simétrica,

$$\kappa_2(A) = \frac{|\lambda_{\max}|}{|\lambda_{\min}|}.$$

Si $\kappa_2(A) \gg 1 \implies$ mal condicionada.

Mal condicionamiento:

- 1) Los errores de redondeo se propagan mucho. Los métodos iterativos convergen muy lentamente.
- 2) Aunque hiciéramos aritmética exacta:

$$\|\vec{r}_x\| \leq \underbrace{\kappa_2(A)}_{10^{12} \cdot \frac{1}{2} 10^{-16} = \frac{1}{2} 10^{-4} \rightarrow 3 \text{ cifras significativas correctas}} \|\vec{r}_b\| \text{ donde } \vec{x} \text{ es solución de } A\vec{x} = \vec{b}$$

Así pues, intentaremos no trabajar con matrices de Vandermonde. Escogeremos una base $\{L_j(x)\}_{j=0}^n$ (base de Lagranje) tal que, $\forall \{f_i\}_{i=0}^n$,

$$f_i = f(x_i) \approx p_n(x_i) = \sum_{j=0}^n f_j L_j(x_i).$$

Es decir, tenemos que imponer

$$L_j(x_i) = \delta_{ij} = \begin{cases} 1 & i = j, \\ 0 & i \neq j. \end{cases}$$

Definición 3.1.5.

$$f(x) \approx p_n(x) = \sum_{j=0}^n f_j L_j(x), \text{ donde } L_j = \frac{\prod_{k \neq j} (x - x_k)}{\prod_{k \neq j} (x_j - x_k)} \in \mathcal{P}^n.$$

Ejemplo 3.1.6. $f(x) \approx p_2(x) = f_0 L_0(x) + f_1 L_1(x) + f_2 L_2(x)$

$$L_0(x) = \frac{(x - 0, 5)(x - 1)}{(0 - 0, 5)(0 - 1)} = 2(x - 0, 5)(x - 1),$$

$$L_1(x) = \frac{(x - 0)(x - 1)}{(0, 5 - 0)(0, 5 - 1)} = (-4)x(x - 1),$$

$$L_2(x) = 2x(x - 0, 5).$$

Teorema 3.1.7. *Teorema del residuo de Lagrange.*

Dada $f \in \mathcal{C}^{n+1}$ i $\{x_i\}_{i=0}^n$, el polinomio interpolador $p_n(x) \in \mathcal{P}^n$ tal que $p_n(x_i) = f(x_i)$, $i = 0, \dots, n$. Tiene error

$$f(x) = p_n(x) + \frac{f^{(n+1)}(\mu(x))}{(n+1)!} L(x), \text{ donde } \begin{cases} L(x) = (x - x_0) \dots (x - x_n), \\ \mu(x) \in [x_0, x_n; x]. \end{cases}$$

Demostración. Observamos que es cierto cuando $x = x_i$ (uno de los puntos base):

$$\underbrace{f(x_i) - p_n(x_i)}_{=0} - \frac{f^{(n+1)}(\mu)}{(n+1)!} \underbrace{L(x_i)}_{=0} = 0 - 0 = 0.$$

Consideramos ahora $\hat{x} \neq x_i, \forall i$ (fijado) y $g(x) = f(x) - p_n(x) - k(\hat{x})L(x)$, donde escogemos k tal que $g(\hat{x}) = 0$ ($k(\hat{x}) = \frac{f(\hat{x}) - p(\hat{x})}{L(\hat{x})}$).

Entonces $g(x)$ tiene $n+2$ ceros diferentes:

$$\begin{cases} \hat{x} \text{ es un cero,} \\ \{x_i\}_{i=0}^n \text{ son } n \text{ ceros.} \end{cases}$$

Por tanto $g'(x)$ tiene almenos $n+1$ ceros (por el teorema de Rolle), $g''(x)$ tiene almenos n ceros, \dots , $g^{(n+1)}(x)$ tiene almenos 1 cero. Llamemosle $\mu(\hat{x}) = \mu$,

$$g^{(n+1)}(\mu) = f^{(n+1)}(\mu) - k(\hat{x})(n+1)! = 0.$$

Sabiendo que $k(\hat{x}) = \frac{f(\hat{x}) - p(\hat{x})}{L(\hat{x})}$, tenemos que

$$\begin{aligned} f^{(n+1)}(\mu) - \frac{f(\hat{x}) - p(\hat{x})}{L(\hat{x})} (n+1)! = 0 &\implies \dots \implies \\ &\implies f(\hat{x}) = p_n(\hat{x}) + \frac{f^{(n+1)}(\mu(\hat{x}))}{(n+1)!} L(\hat{x}), \quad \forall \hat{x}. \end{aligned}$$

□

Observación 3.1.8. *La paradoja de Runge.* Uno de los problemas que la aproximación funcional presenta es la “paradoja de Runge”. Sea $f(x) = \frac{1}{1+25x^2}$, la interpolación con $n+1$ puntos equiespaciados empeora cuando n crece. Si n es muy grande (hay muchos datos), es necesario hacer un cambio de criterio (mínimos cuadrados) o cambiar de tipo (Splines).

3.2. Interpolación de Splines

Proposición 3.2.1. Dados unos puntos base $\{x_i\}_{i=0}^n$, un grado m y una continuidad \mathcal{C}^q , consideramos

$$E_S = \left\{ S(x) \in \mathcal{C}^q([x_0, x_n]) \text{ t. q. } S|_{[x_i, x_{i+1}]} \in \mathcal{P}^m \right\},$$

$$S_i(x) = S(x)|_{[x_i, x_{i+1}]}.$$

Entonces E_S es un espacio vectorial (tiene una base, etc.).

3.2.1. Splines lineals ($m = 1$) \mathcal{C}^0

$$S_i(x) = a_i(x - x_i) + b_i, \text{ ¿cuánto vale } \dim(E_S)?$$

$$\begin{array}{rcl} \# \text{coeficientes} & = & 2n \text{ (2 coeficientes por } n \text{ intervalos)} \\ - \# \text{condiciones} & = & n - 1 \text{ (continuidad de } S \text{ en cada punto interior)} \\ \hline \dim(E_S) & = & n + 1 \text{ (parámetros libres)}. \end{array}$$

Los $n+1$ parámetros libres están determinados unequivocamente por los $n+1$ valores de la función.

Cálculo del Spline:

$$S_i(x) = a_i(x - x_i) + b_i$$

donde $a_i = \frac{f_{i+1} - f_i}{x_{i+1} - x_i}$, $b_i = f_i$.

De todas las bases de Splines lineales \mathcal{C}^0 , una interesante es

$$\left\{ \underbrace{\Phi_i}_{\in E_S} \right\}_{i=0}^n \text{ t. q. } S(x) = \sum_{i=0}^n f_i \Phi_i(x).$$

Las funciones de la base deberán cumplir

$$\Phi_i(x_j) = \delta_{ij}$$

ya que

$$f_j = S(x_j) = \sum_{i=0}^n f_i \Phi_i(x_j), \quad \forall \{f_i\}_{i=0}^n$$

Observación 3.2.2. Podemos usar la base para hacer un ajuste por mínimos cuadrados.

3.2.2. Splines cuadráticos ($m = 2$) \mathcal{C}^1

$$S_i(x) = a_i(x - x_i)^2 + b_i(x - x_i) + c_i, \text{ ¿cuánto vale } \dim(E_S)?$$

$$\begin{array}{rcl} \# \text{coeficientes} & = & 3n \\ - \# \text{condiciones} & = & 2(n - 1) \\ \hline \dim(E_S) & = & n + 2. \end{array}$$

Los $n+2$ parámetros quedan determinados por los $n+1$ valores de f y $S'(x_0) = f'_0$.

3.2.3. Splines cúbicos ($m = 3$) \mathcal{C}^1

$S_i(x) = a_i(x - x_i)^3 + b_i(x - x_i)^2 + c_i(x - x_i) + d_i$, ¿cuánto vale $\dim(E_S)$?

$$\begin{array}{rcl} \# \text{coeficientes} & = & 4n \\ - \# \text{condiciones} & = & 2(n-1) \\ \hline \dim(E_S) & = & 2n+2. \end{array}$$

Para determinar un Spline necesitamos $2(n+1)$ datos: $\{f_i\}_{i=0}^n$ y $\{f'_i\}_{i=0}^n$.

Observación 3.2.3. Si solo tenemos $\{f_i\}_{i=0}^n$, podemos calcular aproximaciones

$$f'_i = \frac{f_{i+1} - f_{i-1}}{2h} \approx f'(x_i)$$

si los puntos son equiespaciados. Si los puntos no son equiespaciados hay que deducir una forma diferente.

Cálculo de una Spline:

$$\begin{aligned} f_i &= S_i(x_i) = d_i, \\ f_{i+1} &= S_i(x_{i+1}) = a_i h_i^3 + b_i h_i^2 + c_i h_i + d_i, \\ f'_i &= S'_i(x_i) = c_i, \\ f'_{i+1} &= S'_i(x_{i+1}) = 3a_i h_i^2 + 2b_i h_i + c_i, \end{aligned}$$

con $h_i = x_{i+1} - x_i$.

Si aislamos, obtenemos

$$\begin{aligned} a_i &= \frac{h_i(f'_i + f'_{i+1}) - 2t_i}{h_i^3}, \\ b_i &= \frac{3t_i - h_i(2f'_i + f'_{i+1})}{h_i^2}, \\ c_i &= f'_i, \\ d_i &= f_i, \end{aligned}$$

con $h_i = x_{i+1} - x_i$ y $t_i = f_{i+1} - f_i$.

La base que nos interesa para los Splines cúbicos \mathcal{C}^1 es la base

$$\{\Phi_i\}_{i=0}^n \cup \{\tilde{\Phi}_i\}_{i=0}^n \text{ t. q. } S(x) = \sum_{i=0}^n f_i \Phi_i(x) + \sum_{i=0}^n f'_i \tilde{\Phi}_i(x).$$

Tenemos que

$$\begin{aligned} \left. \begin{aligned} f_j &= S(x_j) = \sum_{i=0}^n f_i \Phi_i(x_j) + \sum_{i=0}^n f'_i \tilde{\Phi}_i(x_j) \\ f'_j &= S'(x_j) = \sum_{i=0}^n f_i \Phi'_i(x_j) + \sum_{i=0}^n f'_i \tilde{\Phi}'_i(x_j) \end{aligned} \right\} \Rightarrow \\ \Rightarrow \left\{ \begin{aligned} \Phi_i(x_j) &= \delta_{ij}, \quad \tilde{\Phi}_i(x_j) = 0 \quad \forall i, j \\ \Phi'_i(x_j) &= 0 \quad \forall i, j, \quad \tilde{\Phi}'_i(x_j) = \delta_{ij}. \end{aligned} \right. \end{aligned}$$

Por lo tanto, las funciones de la base deberán cumplir

$$\begin{aligned}\Phi_i &= \left\{ f_i = 1, f_j = 0 \quad \forall j \neq i, f'_j = 0 \quad \forall j \right\}, \\ \tilde{\Phi}_i &= \left\{ f_i = 0 \quad \forall i, f'_i = 1, f'_j = 0 \quad \forall j \neq i \right\}.\end{aligned}$$

Observación 3.2.4. Es una base local, lo cual es bueno para mínimos cuadrados, EDPs, una matriz con muchos valores nulos, etc. ya que hay una dependencia local de $S(x)$ con los datos.

3.2.4. Splines cúbicos ($m = 3$) \mathcal{C}^2

$$S_i(x) = a_i(x - x_i)^3 + b_i(x - x_i)^2 + c_i(x - x_i) + d_i, \text{ ¿cuánto vale } \dim(E_S)?$$

$$\begin{array}{rcl}\# \text{coeficientes} & = & 4n \\ - \# \text{condiciones} & = & 3(n-1) \text{ (continuidad de } S, S' \text{ y } S'' \text{ en } \{x_i\}_{i=1}^{n-1}) \\ \hline \dim(E_S) & = & n+3.\end{array}$$

Tenemos $n+1$ valores de la función, y hay que añadir 2 condiciones adicionales. Estas pueden ser:

- *Spline natural* (el más “suave” posible): $f''_0 = 0, f''_n = 0$.
- $S'(x_0) = f'_0, S'(x_n) = f'_n$.

Formulación con derivadas: expresamos el Spline en función de $\{f_i\}_{i=0}^n$ (conocido) y de las derivadas $\{S'_i\}_{i=0}^n$ (no conocido).

$$\begin{aligned}a_i &= a_i(f_i, f_{i+1}, S'_i, S'_{i+1}) \\ b_i &= b_i(f_i, f_{i+1}, S'_i, S'_{i+1}) \\ c_i &= c_i(f_i, f_{i+1}, S'_i, S'_{i+1}) \\ d_i &= d_i(f_i, f_{i+1}, S'_i, S'_{i+1})\end{aligned}$$

Con la formula del Spline cúbico \mathcal{C}^1 tenemos continuidad de $S(x)$ y de $S'(x)$. Escogeremos $\{S'_i\}_{i=0}^n$ para tener continuidad de $S''(x)$:

$$S''_{i-1}(x_i, S'_{i-1}, S'_i) = S''_i(x_i, S'_i, S'_{i+1}), \quad i = 1, \dots, n-1$$

Es decir,

$$\begin{pmatrix} \lambda_1 & 2 & \mu_1 & & & \\ & \lambda_2 & 2 & \mu_2 & & \\ & & \ddots & \ddots & \ddots & \\ & & & \lambda_{n-1} & 2 & \mu_{n-1} \end{pmatrix} \begin{pmatrix} S'_0 \\ \vdots \\ \vdots \\ S'_n \end{pmatrix} = \begin{pmatrix} e_i \\ \vdots \\ \vdots \\ e_{n-1} \end{pmatrix}, \quad \begin{cases} \lambda_i = \frac{h_i}{h_{i-1}+h_i} \\ \mu_i = \frac{h_{i-1}}{h_{i-1}+h_i} \\ e_i = \frac{3}{h_{i-1}+h_i} \left(\frac{h_i}{h_{i-1}} t_{i-1} + \frac{h_{i-1}}{h_i} t_i \right). \end{cases}$$

Para tener un SCD nos faltan dos datos adicionales:

- Podemos imponer

$$\begin{aligned} S'(x_0) &= f'_0 = S'_0, \\ S'(x_n) &= f'_n = S'_n. \end{aligned}$$

- También podemos imponer $S''(x_0) = S''(x_n) = 0$ para obtener un Spline natural (curvatura inicial y final nula).

$$\begin{aligned} \text{i) } 0 &= S''(x_0) = S''_0(x_0) = 6a_0(x_0 - x_0) + 2b_0 = b_0(S'_0, S'_1) = 0, \\ 3t_0 - h_0(2S'_0 + S'_1) &= 0. \end{aligned}$$

$$\text{ii) } 0 = S''_{n-1}(x_n) = a_{n-1} \underbrace{(x_n - x_{n-1})}_{h_{n-1}} + b_{n-1}.$$

Nos queda una ecuación en función de S'_{n-1} y S'_n . Añadimos 2 ecuaciones y nos queda un sistema $(n+1) \times (n+1)$.

Cálculo del Spline:

- Planteamos y resolvemos el sistema lineal $\rightarrow \{S'_i\}_{i=0}^n$.
- Con la fórmula del Spline cúbico \mathcal{C}^1 calculamos a_i, b_i, c_i, d_i en cada intervalo (con $f'_i = S'_i$).

Veamos cual es la base del Spline natural ($S''_0 = S''_n = 0$). Tenemos que $\dim(E_S) = n+1$,

$$S(x) = \sum_{i=0}^n f_i \Phi_i(x).$$

Queremos $\Phi_i(x)$ tal que sean cúbicas a trozos, \mathcal{C}^2 , $S''_0 = S''_n = 0$, $\Phi_i(x_j) = \delta_{ij}$. La base no será local pero tendrá amortiguamiento rápido. La podemos dibujar tomando

$$\begin{cases} f_j = 0 & j \neq i, \\ f_i = 1. \end{cases}$$

Observación 3.2.5. *Formulación con curvaturas.* Fijamos $S''_0 = S''_n = 0$ y resolvemos el sistema

$$\begin{pmatrix} \lambda_1 & 2 & \mu_1 & & & \\ & \lambda_2 & 2 & \mu_2 & & \\ & & \ddots & \ddots & \ddots & \\ & & & \lambda_{n-1} & 2 & \mu_{n-1} \end{pmatrix} \begin{pmatrix} S''_0 \\ \vdots \\ S''_n \end{pmatrix} = \begin{pmatrix} e_i \\ \vdots \\ e_{n-1} \end{pmatrix}, \quad \begin{cases} \lambda_i = \frac{h_i}{h_{i-1}+h_i} \\ \mu_i = \frac{h_{i-1}}{h_{i-1}+h_i} \\ e_i = \frac{6}{h_{i-1}+h_i} \left(\frac{t_i}{h_i} + \frac{t_{i-1}}{h_{i-1}} \right). \end{cases}$$

Teorema 3.2.6. *Teorema de la suavidad del Spline natural.*

De todas las funciones \mathcal{C}^2 que pasan por (x_i, f_i) , $i = 0, \dots, n$, el Spline natural es la que minimiza la “rugosidad”:

$$I(g) := \int_{x_0}^{x_n} [g''(x)]^2 dx$$

El Spline es la interpolación más suave.

Demostración. Sean g una función \mathcal{C}^2 que pasa por los puntos (x_i, f_i) y S el Spline natural.

$$I(g) - I(S) = \int_{x_0}^{x_n} \left[(g'')^2 - (S'')^2 \right] dx = \underbrace{\int_{x_0}^{x_n} (g'' - S'')^2 dx}_{>0, =0 \iff g=S} + 2 \underbrace{\int_{x_0}^{x_n} S'' (g'' - S'') dx}_{:=A}.$$

Tenemos que A es

$$\begin{aligned} A &= \int_{x_0}^{x_n} S'' (g'' - S'') dx = \sum_{i=0}^{n-1} \left(- \int_{x_i}^{x_{i+1}} S''' (g' - S') dx + \left[S'' (g' - S') \right]_{x_i}^{x_{i+1}} \right) = \\ &= \sum_{i=0}^{n-1} -S'''_i [g - S]_{x_i}^{x_{i+1}} + S''_i (g' - S')_{x_i}^{x_{i+1}}. \end{aligned}$$

Por lo tanto $I(g) \geq I(S)$ y la igualdad se da solo si $g = S$. \square

Observación 3.2.7. Vemos que $I(g) \geq I(S) + [S''(g' - S')]_{x_0}^{x_n}$ para S Spline \mathcal{C}^2 cúbico. Si los datos son $\{f_i\}_{i=0}^n$ y $\{S'_0, S'_n\}$, el Spline \mathcal{C}^2 cúbico es el más suave posible en estas condiciones.

3.3. Mínimos cuadrados

Observación 3.3.1. Si tenemos muchos datos puede no ser necesario que la aproximación pase exactamente por los puntos, especialmente experimentales.

Tenemos lo siguiente:

- Datos: $\{f_i\}_{i=0}^n$ o bien $f(x)$.
- Espacio de aproximación V : Polinomios de grado menor que m , funciones lineales a trozos, etc.
- Norma (producto escalar):

- Datos discretos: $\langle u, v \rangle = \sum_{i=0}^n u(x_i) v(x_i)$, $\|u\| = \sqrt{\langle u, u \rangle} = \sqrt{\sum_{i=0}^n (u(x_i))^2}$.
- Datos continuos: $\langle u, v \rangle = \int_a^b u(x) v(x) dx$, $\|u\| = \sqrt{\langle u, u \rangle} = \sqrt{\int_a^b (u(x))^2 dx}$.

Índice alfabético

convergencia

de orden p , [3](#)

lineal, [3](#)

superlineal, [3](#)

error

absoluto, [3](#)

relativo, [6](#)

factor asintótico de convergencia, [5](#)

interpolación de Lagrange, [13](#)

método

consistente, [2](#)

convergente, [3](#)