Cálculo Numérico

Miguel González
mgonzalez.contacto@gmail.com
miguelgg.com

Mayo de 2019

$$P(x) = \sum_{k=0}^{N} \frac{f^{(k)}(x_0)}{k!}(x - x_0)^k + R_{N,x_0}(x)$$

Revisado en 2022
Apuntes de la asignatura impartida por Rafael Orive
en la Universidad Autónoma de Madrid en Mayo de 2019.

Acerca de este documento

Estos apuntes son una versión revisada de los de la asignatura Cálculo Numérico del grado en matemáticas, tomados en Mayo de 2019 por Miguel González. La asignatura fue impartida por Rafael Orive. A los apuntes originales se les ha añadido esta página, una imagen de portada, y breves párrafos explicativos en las zonas menos completas. Asimismo se han revisado las erratas y completado los contenidos faltantes.

Este documento es:

- Una recopilación ordenada y directa de las definiciones y resultados más importantes del tema en cuestión, al nivel de los estudios de grado.
- Una colección de demostraciones completas de dichos resultados (salvo en los casos más básicos).
- Una *guía* para revisar de manera rápida las ideas que se han adquirido previamente, o para consultar enunciados puntuales que puedan no haberse comprendido en su totalidad.

Este documento NO es:

- Un libro de texto de la asignatura.
- Una colección de ejercicios para practicar los conceptos adquiridos.
- Un listado de ejemplos para ilustrar las ideas tratadas. A pesar de ello, en ocasiones se incluyen ejemplos puntuales que puedan ser de especial interés o curiosidad, pero se intentan reducir al mínimo en virtud del primer punto de la lista anterior.

Sobre Cálculo Numérico

Esta asignatura se centra en el estudio de metodos numéricos en cálculo y álgebra lineal básica. El objetivo es encontrar algoritmos, fundados matemáticamente, que permitan resolver cálculos de manera aproximada, que serían difíciles de obtener de manera exacta. Por ejemplo, evaluar funciones o calcular integrales.

En este documento, al final de cada uno de los tres bloques lógicos, se incluye un resumen con los algoritmos desarrollados en la sección.

Requisitos previos

1. Familiaridad con la notación matemática básica.
2. Conocimientos de cálculo equivalentes a la asignatura de Cálculo I. (Funciones reales de una variable, límites, derivadas, integración).
3. Conocimientos de álgebra lineal (espacios vectoriales, aplicaciones lineales, diagonalización, ortogonalidad).

Índice

1. Interpolación

El objetivo es, dada una función f arbitraria, de la que se conoce cierta información, obtener un **polinomio** que permita aproximar dicha función, y cuyos valores son mucho más fáciles de obtener ya que solo tiene lugar suma y multiplicación.

1.1. Polinomio de Taylor

La idea es usar información sobre la función $f : D \to \mathbb{R}$ **en un punto dado** $x_0 \in D$ para generar una aproximación local polinómica. Lo que impondremos en esta aproximación $P(x)$, de grado menor o igual que N, es que $f(x_0) = p(x_0)$ y que $f^{(i)}(x_0) = p^{(i)}(x_0)$ con $1 \leq i \leq N$, es decir, **que coincida en valor y derivadas en el punto** x_0. Esto se conoce como **condiciones de Taylor**.

Teorema 1. *Dada $f : \mathbb{R} \to \mathbb{R}$, $\exists! P \in \mathbb{R}[X]_{\leq N}$ que verifique las condiciones de Taylor, es decir, que cumpla $f(x_0) = P(x_0)$ y que $f^{(i)}(x_0) = P^{(i)}(x_0)$ para $1 \leq i \leq N$. Se conoce como **polinomio de Taylor de grado menor o igual que N de f centrado en** x_0, y su expresión es:*

$$P(x) = \sum_{k=0}^{N} \frac{f^{(k)}(x_0)}{k!}(x - x_0)^k$$

Demostración. Sea P un polinomio cualquiera, expresado de la forma $P(X) = \sum_{k=0}^{N} \frac{b_k}{k!}(x - x_0)^k$ para algunos escalares $\{b_k\}_0^N$. Todo polinomio se puede expresar de esta manera, dado que $\{(x - x_0)^k\}_0^N$ es una base de $\mathbb{R}[X]_{\leq N}$ (el rango de la matriz formada por las coordenadas de estos vectores en canónica es precisamente $N + 1$ al ser triangular). Entonces, $P(x_0) = b_0$ de modo que $b_0 = f(x_0)$. Asimismo, si derivamos sucesivamente, no es difícil ver que $P^{(i)}(x) = \sum_{k=i}^{N} \frac{b_k}{(k-i)!}(x-x_0)^{k-i}$, de manera que $P^{(i)}(x_0) = b_i$, con lo que $b_i = f^{(i)}(x_0)$ para $1 \leq i \leq N$. Es decir, los valores de los escalares $\{b_k\}_0^N$ se ven forzados si imponemos las condiciones de Taylor, y el único polinomio es el descrito en el enunciado del teorema. □

Definición 1. Dadas dos funciones reales f, g definidas en un entorno de $x_0 \in \mathbb{R}$, se dice que $f = o(g)$ **cuando** x **tiende a** x_0 si y solo si $\lim_{x \to x_0} \frac{f(x)}{g(x)} = 0$.

Por otro lado, se dice que $f = O(g)$ cuando x tiende a x_0 si y solo si $\exists K, \delta > 0$ tales que $\forall x$ que cumpla $0 < |x - x_0| < \delta$, se tiene $|f(x)| < K|g(x)|$, es decir, $|\frac{f(x)}{g(x)}| < K$.

Teorema 2. *Sea P_N el polinomio de Taylor de grado menor o igual que N de f en x_0. Entonces, $f - P_N = o((x - x_0)^N)$ cuando x tiende a x_0. Es decir, cerca de x_0, el error es "más pequeño"que $(x - x_0)^N$, que es ya de por sí muy pequeño cerca de x_0. Además, es el único polinomio que satisface esta condición.*

Demostración. Como f y P_N coinciden en derivadas, y son continuos, cuando x tiende a x_0 la diferencia entre ambos, o entre sus derivadas, será nula. Sabiendo esto y aplicando l'Hôpital repetidas veces, sigue que: $\lim_{x \to x_0} \frac{f(x)-P_N(x)}{(x-x_0)^N} = \lim_{x \to x_0} \frac{f'(x)-P'_N(x)}{N(x-x_0)^{N-1}} = \cdots = \lim_{x \to x_0} \frac{f^{(k)}(x)-P_N^{(k)}(x)}{\frac{N!}{(N-k)!}(x-x_0)^{N-k}} = \cdots = \lim_{x \to x_0} \frac{f^{(N)}(x)-P_N^{(N)}(x)}{N!} = 0$, como se quería ver. Para ver que es único, supongamos que hubiese dos polinomios, P y Q, que verificasen la propiedad. Entonces, para ambos, se tendría $\lim_{x \to x_0} \frac{f(x)-P(x)}{(x-x_0)^N} = \lim_{x \to x_0} \frac{f(x)-Q(x)}{(x-x_0)^N} = 0$. Restando ambas expresiones, se tiene $\lim_{x \to x_0} \frac{Q(x)-P(x)}{(x-x_0)^N} = 0$.

Si ahora expresamos $(Q-P)(x) = \sum_0^N a_i(x-x_0)^i$, veremos que, al ser continuo, para que $\lim_{x \to x_0} \frac{\sum_0^N a_i(x-x_0)^i}{(x-x_0)^N} = 0$, debe darse $a_0 = 0$, o si no diverge. Si ahora derivamos una vez por l'Hôpital, nos damos cuenta de que $0 = \lim_{x \to x_0} \frac{\sum_0^N a_i * i(x-x_0)^{i-1}}{N(x-x_0)^{N-1}} = 0$, y ese límite solo existe si $1 * a_1 = 0 \implies a_1 = 0$. Si seguimos de esta

manera, vemos que $a_i = 0\ \forall i \in \{0, ..., N-1\}$. Finalmente, para que el límite sea 0, después de derivar suficientes veces, es fácil ver que obtenemos $0 = \lim_{x \to x_0} Ca_N$ con C una constante positiva, luego $a_N = 0$, y por tanto $Q - P = 0 \implies Q = P$. □

A continuación veremos dos expresiones explícitas para esta diferencia entre función y polinomio, que permitirán, entre otras cosas, ver cuántos términos del polinomio son necesarios para garantizar un error menor que un valor dado.

Teorema 3 (Forma de error de Lagrange)**.** *Sean $x, x_0 \in \mathbb{R}$ distintos, f una función N veces derivable en $[x, x_0]$ y $N+1$ veces derivable en (x, x_0). Sea $P_N(x)$ el polinomio de Taylor de f en x_0 de grado $\leq N$. Entonces, $\exists \xi$ con $x \gtrless \xi \gtrless x_0$ que verifica:*

$$f(x) - P_N(x) = \frac{(x - x_0)^{N+1}}{(N+1)!} f^{(N+1)}(\xi).$$

Demostración. Definimos la constante $M = \frac{f(x) - P_N(x)}{(x-x_0)^{N+1}}$, así como la función $F(t) = f(t) - P_N(t) - M(t - x_0)^{N+1}$. Observamos que, por hipótesis, $F(t)$ es $N+1$ veces derivable en (x_0, x), y N veces en $[x_0, x]$. Observamos que las sucesivas derivadas de $F(t)$ son $F^{(k)}(t) = f^{(k)}(t) - P^{(k)}(t) - M\frac{(N+1)!}{(N-k+1)!}(x - x_0)^{N-k+1}$, para $1 \leq k \leq N+1$. Observamos además que, por construcción, se tiene $F^{(k)}(x_0) = 0$, y $F(x) = 0$. Por el teorema de Rolle, $\exists \xi_1 \in (x, x_0)$ tal que $F'(\xi_1) = 0$, con lo que, una vez más por este teorema, $\exists \xi_2 \in (x_0, \xi_1)$ con $F^{(2)}(\xi_2) = 0$. Iterando este proceso, obtenemos valores de modo que $x_0 \gtrless \xi_{n+1} \gtrless \xi_n \gtrless \cdots \gtrless \xi_2 \gtrless \xi_1 \gtrless x$ tales que $F^{(k)}(\xi_k) = 0$.

Ahora, observemos que, en ξ_{N+1} se tiene $0 = f^{(N+1)}(\xi_{N+1}) - M(N+1)!$, y despejando, se llega a la expresión deseada del error, con $\xi \equiv \xi_{N+1}$ que está en (x_0, x). □

Teorema 4 (Forma de error integral)**.** *Sean $x, x_0 \in \mathbb{R}$ distintos, f una función N veces derivable en $[x, x_0]$ y $N+1$ veces derivable en (x, x_0). Sea $P_N(x)$ el polinomio de Taylor de f en x_0 de grado $\leq N$. Entonces, se verifica:*

$$f(x) - P_N(x) = \frac{1}{N!} \int_{x_0}^{x} (x - t)^N f^{(N+1)}(t) dt.$$

Demostración. Por inducción sobre el grado del polinomio de taylor, hasta N. Con $N = 0$, como la función tiene las derivadas necesarias, podemos escribir:

$$f(x) - P_0(x) = f(x) - f(x_0) = \int_{x_0}^{x} f'(s) ds = \frac{1}{0!} \int_{x_0}^{x} (x - s)^0 f'(s) ds$$

Y por tanto este caso se verifica.

Ahora vamos a comprobar que si se cumple para $k \in \{0, ..., N-1\}$, lo hace para $k+1$, y por tanto tendríamos que el caso N-ésimo que nos interesa se cumple. En efecto, si k está en ese rango, la función tiene las derivadas suficientes como para integrar por partes la expresión hipótesis:

$$f(x) - P_k(x) = \frac{1}{k!} \int_{x_0}^{x} (x - s)^k f^{(k+1)}(s) ds = \frac{1}{k!} [\frac{-(x - s)^{k+1}}{(k+1)}|_{s=x_0}^{x} - \int_{x_0}^{x} \frac{-(x - s)^{k+1}}{(k+1)} f^{(k+2)}(s) ds] =$$

$$= \frac{1}{k!} [\frac{(x - x_0)^{k+1}}{(k+1)} - \int_{x_0}^{x} \frac{-(x - s)^{k+1}}{(k+1)} f^{(k+2)}(s) ds] = \frac{(x - x_0)^{k+1}}{(k+1)!} + \frac{1}{(k+1)!} \int_{x_0}^{x} (x - s)^{k+1} f^{(k+2)}(s) ds$$

Como vemos, al integrar por partes se ha generado un término del polinomio de Taylor. Despejando, se tiene:

$$f(x) - P_{k+1}(x) = \frac{1}{(k+1)!}\int_{x_0} (x-s)^{k+1} f^{(k+2)}(s)ds$$

Que es lo que se quería. □

Estos errores no solo van a permitir determinar cuántos términos necesitamos del polinomio para estimar el valor de f en un punto con un error dado, sino que también nos van a permitir determinar si $f - P_n$ converge en algún punto cuando $n \to \infty$, es decir, si se puede obtener una serie polinómica para el valor de la función en algun punto. A continuación veremos un corolario:

Proposición 1. *Bajo las condiciones de los teoremas de error para cualquier x en un entorno de x_0 (el intervalo $(x_0 - \delta, x_0 + \delta)$ para $\delta > 0$), y si $|f^{N+1}(s)| \leq K$ con s en dicho entorno, se tiene $f(x) - P_N(x_0) = O((x - x_0)^{N+1})$ cuando x tiende a x_0.*

Demostración. Si x está en ese entorno, entonces $|\frac{f(x)-P_N(x)}{(x-x_0)^{N+1}}| = |\frac{f^{n+1}(\xi)}{(N+1)!}| \leq \frac{K}{(N+1)!} = K' > 0$, dado que ξ también estará en el entorno. □

1.2. Algoritmo de Horner

El problema inicial es evaluar, de la forma más eficiente posible, el polinomio $P(x) = \sum_0^n a_i x^i$ en el punto x_0. Observemos que, evaluando como se esperaría, se realizan n sumas y $n + (n-1) + ... + 2 + 1 = \frac{n(n+1)}{2}$ multiplicaciones, lo cual puede mejorar con el algoritmo de Horner:

Proposición 2. *Para evaluar el polinomio $P(x) = \sum_0^n a_i x^i$ en el punto x_0, podemos operar como sigue:*

1. *Definimos $q_{n-1} = a_n$.*
2. *Iterativamente, calculamos $q_{n-i-1} = q_{n-i}x_0 + a_{n-i}$ para $i = 1, 2, 3, ..., n$.*
3. *Se tiene que $q_{-1} = P(x_0)$*

Como se observa, solo han sido precisas n sumas y n productos.

Demostración. Tenemos que $q_{-1} = q_0x_0 + a_0 = (q_1x_0 + a_1)x_0 + a_0 = q_1x_0^2 + a_1x_0 + a_0 = (q_2x_0 + a_2)x_0^2 + a_1x_0 + a_0 = \cdots = (q_{n-1}x_0 + a_{n-1})x_0^{n-1} + \sum_0^{n-2} a_i x^i = a_n x_0^n + a_{n-1}x^{n-1} + \sum_0^{n-2} a_i x^i$. □

Observación 1. Tras realizar el algoritmo de Horner como anteriormente, se puede escribir:

$$P(x) = (\sum_0^{n-1} q_i x^i)(x - x_0) + q_{-1}$$

Demostración. Podemos escribir en primer lugar $P(x) = a_n x^{n-1}(x - x_0) + (a_{n-1} + a_n x_0)x^{n-1} + \sum_0^{n-2} a_i x^i$, es decir, $P(x) = q_{n-1}x^{n-1}(x - x_0) + q_{n-2}x^{n-1} + \sum_0^{n-2} a_i x^i$.

Una vez más, podemos poner: $P(x) = q_{n-1}x^{n-1}(x-x_0) + q_{n-2}x^{n-2}(x-x_0) + (q_{n-2}x_0 + a_{n-2})x^{n-2} + \sum_0^{n-3} a_i x^i = q_{n-1}x^{n-1}(x - x_0) + q_{n-2}x^{n-2}(x - x_0) + q_{n-3}x^{n-2} + \sum_0^{n-3} a_i x^i$.

Si continuamos de esta manera, llegamos a $(\sum_0^{n-1} q_i x^i)(x - x_0) + q_{-1}x^0$, lo que queríamos. □

Observación 2 (Cambio de base). Supongamos que queremos escribir $P(x) = \sum_0^n b_i(x - x_0)^i$. Entonces, podemos hallar los b_i aplicando el algoritmo de Horner sucesivas veces a los polinomios que van surgiendo en la expresión en la observación anterior. De esta manera, $b_0 = q_{-1}$, $b_1 = q'_{-1}$, etc, donde cada b_i es el q_{-1} en la i ésima iteración del algoritmo.

Esto es así porque se tiene $P(x) = (\sum_0^{n-1} q_i x^i)(x - x_0) + q_{-1} = (\sum_0^{n-2} q'_i x^i)(x - x_0) + q'_{-1})(x - x_0) + q_{-1} = (\sum_0^{n-2} q'_i x^i)(x - x_0)^2 + q'_{-1}(x - x_0) + q_{-1}$, etcétera.

1.3. Interpolación polinómica de Lagrange

El objetivo es, dados $N+1$ puntos $x_0, ..., x_n$ y sus imágenes por f, $f(x_0), ..., f(x_n)$, interpolar la función por un polinomio de grado N, es decir, obtener un polinomio de grado N que pase por los puntos $(x_i, f(x_i))$. Una primera aproximación es escribir $P_N(x) = \sum_0^n a_i x^i$, luego lo que se quiere es resolver el sistema:

$$\left\{ \sum_{i=0}^N a_i x_j^i = f(x_j) \text{ para } j \in \{0, 1, ..., n\} \right. \quad .$$

O lo que es lo mismo, el sistema $Ax = y$ donde $x^t = (a_0, ..., a_n)$, $y^t = (f(x_0), ..., f(x_n))$, y A es una **matriz de Vandermonde**:

$$A = \begin{pmatrix} 1 & x_0 & x_0^2 & \dots & x_0^N \\ 1 & x_1 & x_1^2 & \dots & x_1^N \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 1 & x_n & x_n^2 & \dots & x_n^N \end{pmatrix}.$$

De determinante, como se puede demostrar en este tipo de matrices, $det(A) = \prod_{i<j}(x_j - x_i)$ luego el sistema tiene solución, **y única (esto es, solo hay un polinomio de Lagrange de ese grado)**, siempre que $x_j \neq x_i$ si $i \neq j$, o lo que es lo mismo, si tomamos los puntos distintos.

No obstante, la resolución del problema es más cómoda en otra base:

Proposición 3. *Definimos el conjunto de polinomios $\{l_i\}_{i=0}^n$ de manera que $l_i = \prod_{j \neq i} \frac{(x - x_j)}{(x_i - x_j)}$. Estos forman base de $\mathbb{R}[X]_n$, y verifican que $l_i(x_j) = \delta_{ij}$.*

Demostración. Veamos lo segundo: $l_i(x_j) = \prod_{k \neq i} \frac{(x_j - x_k)}{(x_i - x_k)}$, y ahora si $i = j$ se tiene lo mismo en numerador y denominador, luego vale 1, y si no, aparece el término $\frac{x_j - x_j}{x_i - x_j} = 0$ y vale 0. Ahora podemos ver que son base. En efecto, si hubiese escalares α_i tales que $\sum_0^n \alpha_i l_i = 0$, entonces, bastaría con evaluar en x_j: $0 = \sum_0^n \alpha_i l_i(x_j) = \alpha_j$, para $j \in \{0, 1, ..., n\}$, luego todos los escalares son nulos y el conjunto es linealmente independiente. □

Observación 3. Con esta base, el interpolador de Lagrange es $P_N(x) = \sum_{i=0}^n f(x_i) l_i$, ya que por construcción todos los polinomios se anularán en cada x_j salvo el que acompaña a $f(x_j)$.

Sin embargo, este otro método acarrea un inconveniente. Si queremos incrementar la precisión del polinomio agregando otro punto, debemos recalcular todos los l_i, además de incorporar uno nuevo. Queremos que, al igual que con el polinomio de Taylor, se pueda aumentar el grado del polinomio simplemente añadiendo el término de grado $N+1$.

Para hacer esto, podemos ir paso a paso. Empezamos con $p_0(x) = f(x_0)$ constante, y buscamos $p_1(x) = p_0(x) + q_1(x)$. Es claro que $q_1(x_0) = 0$, para mantener el valor de lo que llevábamos, y además, $q_1(x_1) = f(x_1) - p_0(x_1)$, luego basta con hacer $q_1(x) = (x - x_0)\frac{f(x_1) - f(x_0)}{x_1 - x_0}$. A este coeficiente que acompaña a $x - x_0$ se le denota $f[x_0, x_1]$.

En general, $p_n(x) = p_{n-1}(x) + q_n(x)$, con $q_n(x) = f[x_n, ..., x_0] \cdot (x - x_0)(x - x_1)\dots(x - x_{n-1})$ donde el coeficiente asegura que $q_n(x_n) = f(x_n) - p_{n-1}(x_n)$. Como vemos, se pueden añadir términos de uno en uno. Esto se conoce como **método de Newton**, y desarrollaremos a continuación una manera de obtener los coeficientes mencionados.

1.3.1. Método de Newton

Definición 2. Con N natural, $x_0, ..., x_N$ $N+1$ puntos distintos y f definida en estos puntos. Se denomina **diferencia dividida** de f en esos puntos al coeficiente de X^N en el desarrollo en potencias de $P_N(x)$. Se denota por $f[x_0, x_1, ..., x_N]$.

El polinomio de Lagrange, por este método, se escribe entonces como:

$$P_N(x) = f[x_0]+f[x_0,x_1](x-x_0)+f[x_0,x_1,x_2](x-x_0)(x-x_1)+\cdots+f[x_0,\ldots,x_n](x-x_0)(x-x_1)\ldots(x-x_{n-1})$$

Esta construcción se basa en que el primer sumando interpole x_0, el segundo x_0, x_1, el tercero x_0, x_1, x_2, etcétera.

Para aumentar su grado basta con incorporar un término nuevo, por como está construido (los términos nuevos se anulan en los puntos que ya interpola lo anterior, de ahí que estos coeficientes sean justo las diferencias divididas: hasta el término de grado i, se trada de $P_i(x)$).

Observemos que $f[x_i] = f(x_i)$, dado que con un solo punto, el polinomio que lo interpola es siempre de grado 0 constante. Veamos como calcular en general las diferencias divididas:

Proposición 4. *Se verifica:*

$$f[x_0,...,x_n] = \frac{f[x_1,...,x_n] - f[x_0,...,x_{n-1}]}{x_n - x_0}$$

Demostración. Repitamos la construcción del polinomio P_n, pero esta vez cambiaremos el orden: construiremos con la lista $\{x_n, x_{n-1}, ..., x_1, x_0\}$. En ese caso, $P_N(x) = f[x_n] + f[x_n, x_{n-1}](x - x_n) + f[x_n, x_{n-1}, x_{n-2}](x - x_n)(x - x_{n-1}) + \cdots + f[x_n, \ldots, x_0](x - x_n)(x - x_{n-1})\ldots(x - x_1)$. Al ser único el interpolador, debe ser igual que el que ya conocíamos. En concreto, el término que acompaña a x^{n-1} debe serlo. En esta expresión, ese término es $f[x_n, ..., x_1] - f[x_n, ..., x_0](x_1 + ... + x_n)$, y en la primera, era $f[x_0, ..., x_{n-1}] - f[x_0, ..., x_n](x_0 + ... + x_{n-1})$. Ahora, igualando ambos y usando que, por definición, $f[x_n, ..., x_1] = f[x_1, ..., x_n]$ y $f[x_n, ..., x_0] = f[x_0, ..., x_n]$, se tiene la expresión de la proposición. □

Veamos que construir el polinomio de grado 0 cuesta 0 operaciones, el de grado 1 cuesta 2 restas y 1 división a partir del de grado 0, el de grado 2 a partir del de grado 1 requiere $f[x_0, x_1, x_2]$ que, por el método anterior son 2 restas y 1 división más lo que cueste $f[x_1, x_2]$, que sabemos son 2 restas y 1 división, y así sucesivamente. En general, aumentar a grado N requiere $2N$ restas y N divisiones, luego en total construir el de grado N requiere $N(N + 1)$ restas y $\frac{N(N+1)}{2}$ divisiones.

Ahora tratemos de hallar una expresión para el error que se comete al interpolar de esta manera.

Observación 4. Fijado x, si $P_n(x)$ interpola f en los puntos $\{x_i\}_0^n$, se tiene:

$$f(x) - P_n(x) = f[x_0, ..., x_n, x] \prod_{i=0}^{n} (x - x_i)$$

Demostración. Sigue directamente de cómo se construye este polinomio: sabemos que el interpolador $P_{N+1}(s)$ en $\{x_0, ..., x_n, x\}$ verifica $P_{N+1}(s) = P_N(s) + f[x_0, ..., x_n, x] \prod_{i=0}^{n}(s - x_i)$, luego basta con evaluar en x: $f(x) = P_{N+1}(x) = P_N(x) + f[x_0, ..., x_n, x] \prod_{i=0}^{n}(x - x_i)$. □

Esta observación, evidente por construcción, no resulta nada útil: hemos definido $f[x_0, ..., x_n, x]$ precisamente para que ocurriese esto, y, como es natural, para su cálculo nos hace falta $f(x)$ directamente, así que no necesitamos estimarla con ningún error.

Teorema 5. *Sean $\{x_i\}_0^n \subset \mathbb{R}$ distintos y $x \in \mathbb{R}$. Sea $a = \min\{x_0, ..., x_n, x\} < b = \{x_0, ..., x_n, x\}$, como para que todos estos puntos estén en $[a, b]$. Sea $f \in \mathcal{C}^{n+1}([a, b])$ y $P_n(x)$ su interpolador de Lagrange en $x_0, ..., x_n$. Entonces $\exists \xi \in [a, b]$ tal que:*

$$f(x) - P_n(x) = \frac{f^{(n+1)}(\xi)}{(n + 1)!} \prod_{i=0}^{n} (x - x_i)$$

Demostración. Sea $M = \frac{f(x)-P_n(x)}{\prod_0^n (x-x_i)}$ y $F(t) = f(t) - P_n(t) - M\prod_0^n (t-x_i)$. Es claro que $F \in \mathcal{C}^{n+1}([a,b])$. Por construcción, es fácil ver que $F(x) = 0 = F(x_i)\ \forall i \in \{0,...,n\}$. Sean $\{y_0,...,y_{n+1}\}$ los elementos de $\{x, x_0,...,x_n\}$ pero ordenados de manera creciente. Entonces, por el teorema de Rolle, $\exists\{\xi_i^1\}_0^n$, con cada $\xi_i^1 \in (y_i, y_{i+1})$, y por tanto en $[a,b]$, tales que $F'(\xi_i^1) = 0$.

Aplicando de nuevo el teorema de Rolle, hay $\{\xi_i^2\}_0^{n-1} \subset [a,b]$, tales que $F''(\xi_i^2) = 0$. Aplicándolo sucesivas veces de este modo, llegamos a que $\exists\xi$ tal que $F^{n+1}(\xi) = 0$.

Por otro lado, tenemos que $F^{n+1}(t) = f^{n+1}(t) + 0 - \frac{f(x)-P_n(x)}{\prod_0^n (x-x_i)}(N+1)!$. Basta con evaluar en ξ y despejar. □

1.4. Interpolación polinómica a trozos

Como la interpolación de Lagrange, por el error que hemos visto, no tiene por qué converger bien ni siquiera en intervalos pequeños, tratamos de encontrar nuevas formas de interpolar funciones. La idea de la interpolación a trozos es dividir el intervalo a interpolar y crear varios polinomios interpoladores, uno en cada división.

Definición 3. Una partición Δ_n del intervalo $[a,b]$ es un conjunto de puntos $x_0 = a < x_1 < x_2 < \ldots < x_n = b$ que lo dividen.

Definición 4. El conjunto $M_k^i(\Delta_n)$ es el conjunto de las funciones $f \in \mathcal{C}^k([a,b])$ tales que f restringida a cada intervalo (x_{j-1}, j) de la partición ($j \in \{1,2,\ldots,n\}$) es un polinomio de grado $\leq i$.

Es decir, son funciones que *a trozos* son polinomios. Para las interpolaciones lineales, nos interesa M_0^1, que son las funciones continuas que en cada trozo de la partición son lineales (rectas).

Teorema 6. *Sea f una función real. Dada $\Delta_n \subset [a,b]$ y $f(x_0),\ldots,f(x_n)$ los valores de f en la partición, $\exists! S \in M_0^1(\Delta_n)$ tal que $S(x_i) = f(x_i)\ \forall i \in \{0,\ldots,n\}$.*

Buscamos una función $S(x)$ que sea lineal en cada trozo de la partición y pase por esos puntos. En concreto, en el trozo i-ésimo, $[x_i, x_{i+1}]$, ha de ser una recta que pase por $(x_i, f(x_i))$, $(x_{i+1}, f(x_{i+1}))$. Sabemos que el polinomio interpolador de lagrange $p_i(x) = f(x_i) + \frac{f(x_{i+1})-f(x_i)}{x_{i+1}-x_i}(x-x_i)$ cumple eso y es único. Por tanto, basta con que:

$$S(x) = \left\{ p_i(x) = f(x_i) + \tfrac{f(x_{i+1})-f(x_i)}{x_{i+1}-x_i}(x-x_i) \quad \text{si } x \in [x_i, x_{i+1}] \quad (i \in \{0,\ldots,n-1\}) \right.$$

Y como cada trozo es único, lo es S. □

Definición 5. Dada f con valores conocidos en $x_0,\ldots,x_n$ de la partición Δ_n, se conoce como $S(x)$ el **polinomio interpolante lineal a trozos** de f en $x_0,\ldots,x_n$ a la función tal que:

$$S(x) = f(x_{i-1}) + f[x_{i-1}, x_i](x - x_{i-1}) \quad (x \in [x_{i-1}, x_i])$$

Con $i \in \{1,\ldots,n\}$. Como vemos, queda definida a lo largo de toda la partición, en cada intervalo es el interpolante lineal de Lagrange y por tanto vale igual que f en los puntos de la partición.

Vamos a ver que esta forma de interpolar funciones tiene varias ventajas. La primera es que es fácil de calcular, y la segunda es que conforme disminuye la norma de la partición (es decir, aumentamos los puntos), el interpolante converge siempre a la función (dadas algunas condiciones acerca de su segunda derivada). Veámoslo:

Teorema 7. *Supongamos que $f \in \mathcal{C}^2([a,b])$, y tenemos $\Delta_n = \{x_i\}_0^n$ partición de $[a,b]$. Además, supongamos que $\exists K \in \mathbb{R}$ con $|f''(x)| \leq K\ \forall x \in [a,b]$. Si definimos $h_i = x_i - x_{i-1}$ y la **norma o diámetro** de la partición por $h = \max\limits_{i\in\{1\ldots n\}} \{h_i\}$, tenemos que:*

$$|f(x) - S(x)| \leq \frac{K}{8}h^2$$

Y como vemos la convergencia es ***cuadrática*** *con el diámetro de la partición. Con esta expresión podemos expresar, además, que:* $f(x) - S(x) = O(h^2)$*. Cuando la norma converja a* 0 *lo hará el error también.*

Demostración. Como f es $\mathcal{C}^2$, tenemos que, $\forall x \in [x_{i-1}, x_i]$:

$$|f(x) - S(x)| = \frac{|f''(\xi_i)|}{2!}|x - x_{i-1}| \cdot |x - x_i|$$

Con $\xi_i \in (x_{i-1}, x_i)$. Esto sigue de aplicar el error a cada interpolador de Lagrange que compone la función. Ahora, sabemos que $\frac{|f''(\xi_i)|}{2!}|x - x_{i-1}| \cdot |x - x_i| \leq \frac{K}{2}|x - x_{i-1}| \cdot |x - x_i|$. Como $x \in [x_{i-1}, x_i]$, se tiene: $|x - x_{i-1}| \cdot |x - x_i| = -(x - x_{i-1}) \cdot (x - x_i)$, que tras derivar, alcanza su máximo en $x_{i-\frac{1}{2}} \equiv \frac{x_i + x_{i-1}}{2}$. Evaluando, dicho máximo es $(\frac{x_i - x_{i-1}}{2})^2$, de tal modo que:

$$|f(x) - S(x)| \leq \frac{K}{8}(x_i - x_{i-1})^2 = \frac{K}{8}h_i^2$$

Como queríamos ver. Ahora, esté donde esté x, se podrá acotar por el máximo de las longitudes de cada intervalo. □

1.4.1. Interpolación cuadrática

Ahora vamos a estudiar como interpolar en $M_0^2(\Delta_n)$. La idea es similar a la interpolación lineal, pero está claro que ahora hay infinitos polinomios cuadráticos en cada intervalo de la partición, con lo que necesitamos otro punto más para interpolar. Cogeremos habitualmente $x_{i-\frac{1}{2}} \equiv \frac{x_i + x_{i-1}}{2}$.

Definición 6. Dada f con valores conocidos en $x_0, \ldots, x_n$ de la partición Δ_n de $[a, b]$, así como en los puntos medios $x_{i-\frac{1}{2}}$ $\forall i \in [1, \ldots, n]$, se conoce **polinomio interpolante cuadrático a trozos** de f en $x_0, \ldots, x_n$ a la función $S(x)$ tal que:

$$S(x) = f(x_{i-1}) + f[x_{i-1}, x_i](x - x_{i-1}) + f[x_{i-1}, x_i, x_{i-\frac{1}{2}}](x - x_{i-1})(x - x_i) \quad (x \in [x_{i-1}, x_i])$$

Con $i \in \{1, \ldots, n\}$. Como vemos, queda definida a lo largo de toda la partición, en cada intervalo es el interpolante cuadrático de Lagrange y por tanto vale igual que f en los puntos de la partición. Asimismo, es el único elemento de $M_0^2(\Delta_n)$ que interpola esos puntos, dado que cada uno de los trozos es único con esas características, como sabemos.

Tiene la desventaja de que cada trozo es ligeramente más costoso de calcular, y hace falta información de la función en más puntos cada vez. No obstante, tiene cotas de error mucho más impresionantes que el lineal a trozos y, por supuesto, que el de Lagrange.

Teorema 8. *Supongamos que* $f \in \mathcal{C}^3([a, b])$*, y tenemos* $\Delta_n = \{x_i\}_0^n$ *partición de* $[a, b]$*. Además, supongamos que* $\exists K \in \mathbb{R}$ *con* $|f^{(3)}(x)| \leq K$ $\forall x \in [a, b]$*. Si definimos* $h_i = x_i - x_{i-1}$ *y la* ***norma o diámetro*** *de la partición por* $h = \max_{i \in \{1 \ldots n\}} \{h_i\}$*, tenemos que:*

$$|f(x) - S(x)| \leq \frac{K\sqrt{3}}{216}h^3$$

Y como vemos la convergencia es ***cúbica*** *con el diámetro de la partición. Con esta expresión podemos expresar, además, que:* $f(x) - S(x) = O(h^3)$*. Cuando la norma converja a* 0 *lo hará el error también.*

Demostración. Como f es $\mathcal{C}^3$, tenemos que, $\forall x \in [x_{i-1}, x_i]$:

$$|f(x) - S(x)| = \frac{|f^{(3)}(\xi_i)|}{3!}|x - x_{i-1}| \cdot |x - x_i| \cdot |x - x_{i-\frac{1}{2}}|$$

Con $\xi_i \in (x_{i-1}, x_i)$. Esto sigue de aplicar el error a cada interpolador de Lagrange que compone la función. Como ya sabemos acotar la derivada, por hipótesis, vamos a tratar de acotar $|x - x_{i-1}| \cdot |x - x_i| \cdot |x - x_{i-\frac{1}{2}}|$. Si derivamos el polinomio dentro del valor absoluto, obtenemos dos puntos de extremo: $\tilde{x} = x_{i-\frac{1}{2}} \pm \frac{h_i}{2\sqrt{3}}$. Si sustituimos en el polinomio y tomamos valor absoluto, sigue que: $|x - x_{i-1}| \cdot |x - x_i| \cdot |x - x_{i-\frac{1}{2}}| \leq \frac{\sqrt{3}h_i^3}{36}$, con lo que se tiene lo que se quería. Ahora, esté donde esté x, se podrá acotar por el máximo de las longitudes de cada intervalo. □

1.4.2. Interpolación por Splines

Hemos visto que las aproximaciones a trozos son muy buenas, y fáciles de calcular. Para las de grado mayor a 1, además, hemos tenido que tomar los puntos medios de la partición para definir unívocamente la curva. No obstante, podemos, en lugar de eso, exigir que el interpolador resultante sea diferenciable: esto se conoce como spline.

Definición 7 (Spline cuadrático)**.** Dada f y la partición Δ_n de $[a, b]$, su **spline interpolador cuadrático** $S \in \mathcal{M}_1^2([a, b])$ es una función, C^1, tal que $S(x_i) = f(x_i)\ \forall i \in \{0, ..., n\}$.

Intuitivamente, esto fuerza cada uno de los trozos a *enganchar bien*, pero siempre queda libre la elección de cómo poner el primer trozo.

Proposición 5. *Dada f y una partición, existe un spline interpolador cuadrático (no es único).*

Demostración. El spline viene dado a trozos por $S(x) = a_i x^2 + b_i x + c_i$, para $x \in [x_{i-1}, x_i]$ con $i \in \{1, \ldots, n\}$. Por tanto, tenemos las siguientes restricciones que imponer:

1. $f(x_i) = a_i x_i^2 + b_i x_i + c_i$ (Valor de f y continuidad, una por cada trozo)
2. $f(x_{i-1}) = a_i x_{i-1}^2 + b_i x_{i-1} + c_i$ (Valor de f y continuidad, una por cada trozo)
3. $S(x_i^-) = 2a_i x_i + b_i = 2a_{i+1} x_i + b_{i+1} = S(x_i^+)$ para $i \in \{1, \ldots, n-1\}$ (existencia y continuidad de la derivada en puntos no extremos)

Esto son $3n - 1$ ecuaciones, para $3n$ incógnitas. Existe y, para determinar qué solución forzar, se pueden imponer otras características (como que la derivada en x_0 y x_n coincidan, o fijar la derivada incial a un valor dado, es decir, proporcionar una pendiente de comienzo). □

Una aproximación mucho mejor es la cúbica:

Definición 8 (Spline cúbico)**.** Dada f y la partición Δ_n de $[a, b]$, su **spline interpolador cúbico** $S \in \mathcal{M}_2^3([a, b])$ es una función, C^2, tal que $S(x_i) = f(x_i)\ \forall i \in \{0, ..., n\}$.

Por un argumento similar al anterior, incluyendo las derivadas segundas en los puntos no extremos de la partición, obtenemos que existe. Además, obtenemos $4n - 2$ ecuaciones para $4n$ incógnitas, de tal modo que se suele optar por agregar alguna de estas condiciones adicionales:

1. **Condición natural.** Consiste en imponer que $S''(x_0) = S''(x_n) = 0$.
2. **Respecto a la derivada.** Consiste en fijar el valor de $S'(x_0)$ y de $S'(x_n)$ a un valor prefijado (no necesariamente igual). Es decir, exigir una pendiente de comienzo y una pendiente de fin.

3. **Periodicidad.** Si la función vale lo mismo en x_0 y en x_n, se puede imponer que el Spline sea periódico, es decir, que $S'(x_0) = S'(x_n)$ y $S''(x_0) = S''(x_n)$. Esto permitiría recomenzar el spline en su final, y lo resultante seguiría siendo un spline (útil para interpolar curvas periódicas).

Método simple para el cálculo de Splines Cúbicos.

Si bien se pueden resolver las $4n$ ecuaciones planteadas anteriormente, se puede plantear otro sistema equivalente y algo más sencillo de resolver computacionalmente, al disponer de menos ecuaciones. Supondremos que adoptamos la condición natural, y sea S el spline de f en Δ_n partición de $[a, b]$. Denotamos $S''(x_i) = M_i$, con lo que $M_0 = M_n = 0$. También denotamos $y_i = f(x_i)$ los puntos por los que pasa el spline.

1. **Los M_i permiten hallar el spline.** Lo que debemos observar, en primer lugar, es que $S''(x)$ es el interpolante lineal a trozos que pasa por los M_i. Efectivamente, al derivar 2 veces las cúbicas, obtenemos rectas, continuas al ser $S \in \mathcal{C}^2$, y pasan por M_i en cada x_i por definición. Es decir:

$$S''(x) = M_{i-1}\frac{x_i - x}{x_i - x_{i-1}} + M_i\frac{x - x_{i-1}}{x_i - x_{i-1}} \qquad (x \in [x_{i-1}, x_i])$$

Por lo tanto:

$$S'(x) = -M_{i-1}\frac{(x_i - x)^2}{2(x_i - x_{i-1})} + M_i\frac{(x - x_{i-1})^2}{2(x_i - x_{i-1})} + A_i \qquad (x \in [x_{i-1}, x_i])$$

Con cada $A_i \in \mathbb{R}$. Finalmente:

$$S(x) = +M_{i-1}\frac{(x_i - x)^3}{6(x_i - x_{i-1})} + M_i\frac{(x - x_{i-1})^3}{6(x_i - x_{i-1})} + A_i(x - x_{i-1}) + B_i \qquad (x \in [x_{i-1}, x_i])$$

Hemos ido organizando las constantes cuidadosamente, evitando que aparezcan términos de x aislados. Ahora tenemos dependencia de A_i y B_i, pero si imponemos, en cada uno de estos intervalos, $S(x_{i-1}) = y_{i-1}$, y despejamos, sigue que:

$$B_i = y_{i-1} - \frac{M_{i-1}}{6}(x_{i-1} - x_i)^2$$

Y si ahora sustituimos este valor conocido, e imponemos $S(x_i) = y_i$, tras despejar:

$$A_i = \frac{y_i - y_{i-1}}{x_i - x_{i-1}} + \frac{M_{i-1} - M_i}{6}(x_i - x_{i-1})$$

Lo que nos da una expresión completa de S en términos de datos conocidos y de los M_i, sus derivadas segundas.

2. **Obtención de los M_i.** Ahora, basta con referirse a la ecuación de S'. Una vez sustituido el valor de A_i, evaluaremos cada punto interior x_i en el trozo de la izquierda y en el de la derecha. Esto da lugar a:

$$\begin{cases} S'(x_i^+) = -\frac{M_i(x_{i+1} - x_i)}{2} + \frac{y_{i+1} - y_i}{x_{i+1} - x_i} + \frac{M_i - M_{i+1}}{6}(x_{i+1} - x_i) \\ S'(x_i^-) = +M_i\frac{(x_i - x_{i-1})}{2} + \frac{y_i - y_{i-1}}{x_i - x_{i-1}} + \frac{M_{i-1} - M_i}{6}(x_i - x_{i-1}) \end{cases}$$

Para $i \in \{1, \ldots, n-1\}$. Igualando ambas expresiones, dado que sabemos que la derivada es continua, se tiene, finalmente:

$$\frac{x_{i+1}-x_i}{6}M_{i+1}+\frac{x_i-x_{i-1}}{6}M_{i-1}+\frac{x_{i+1}-x_{i-1}}{3}M_i=\frac{y_{i+1}-y_i}{x_{i+1}-x_i}-\frac{y_i-y_{i-1}}{x_i-x_{i-1}} \qquad (i\in\{1,\dots,n-1\})$$

Estas son las $n-1$ **ecuaciones del Spline**. Resolviendo estas, con $M_0=M_n=0$, obtenemos una única solución, y sustituyéndolas en la expresión de 1 obtenemos la ecuación del spline. Si asumimos que los puntos de la partición son equidistantes de distancia h, las ecuaciones quedan:

$$\frac{M_{i-1}}{6}+\frac{2M_i}{3}+\frac{M_{i+1}}{6}=\frac{1}{h}\left(\frac{y_{i+1}-y_i}{h}-\frac{y_i-y_{i-1}}{h}\right) \qquad (i\in\{1,\dots,n-1\})$$

Es un sistema de $n-1$ ecuaciones con $n-1$ incógnitas (habiendo impuesto $M_0=M_n=0$), de única solución al ser su matriz:

$$A=\begin{pmatrix} \frac{2}{3} & \frac{1}{6} & 0 & \dots & 0 \\ \frac{1}{6} & \frac{2}{3} & \frac{1}{6} & \dots & 0 \\ 0 & \frac{1}{6} & \frac{2}{3} & \dots & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & 0 & \dots & \frac{2}{3} \end{pmatrix}$$

Finalmente, un resultado:

Teorema 9. *Dada f en $\mathcal{C}^4([a,b])$, y Δ_n una partición de $[a,b]$ con h su norma. Si S es el spline cúbico hallado anteriormente (o con otra de las condiciones adicionales que se explicaron), se tiene que $\exists c_i>0$ tales que:*

$$|f^{(i)}(x)-S^{(i)}(x)|\le c_i\cdot K\cdot h^{4-i}$$

Con $\frac{h}{x_j-x_{j-1}}\le K\ \forall j\in\{1,\dots,N\}$, para cualquier $i\in\{0,1,2\}$.

Es decir, el spline aproxima muy bien tanto la función como sus derivadas, hasta la segunda.

1.5. Polinomios de Chebyshev

Volviendo al error cometido por la interpolación en forma de Lagrange, sabemos que si $f\in\mathcal{C}^{n+1}([a,b])$, y $P_n(x)$ es el interpolador de lagrange en $\{x_i\}_0^n\subset[a,b]$, además de que $|f^{n+1}(x)|\le K$ en $[a,b]$, entonces:

$$|f(x)-P_n(x)|=\frac{K}{(n+1)!}|(x-x_0)\dots(x-x_n)|$$

Nos planteamos ahora cómo deberíamos elegir los x_i para **minimizar el error** en todo el intervalo $[a,b]$, es decir, cuáles son los x_i tales que el máximo, con x variando, de $|(x-x_0)\dots(x-x_n)|$, es mínimo.

En primer lugar, para simplificar los cálculos, con independencia del intervalo que estemos tratando, vamos a reescalarlo linealmente para convertirlo en $[-1,1]$. No es difícil calcular que la única transformación lineal (afín) que lleva $[a,b]$ en $[-1,1]$, sin revertir orientación, es: $y=\frac{2}{b-a}(x-\frac{a+b}{2})$.

Ahora, reescribiremos la expresión que queremos minimizar en términos de elementos de $[-1,1]$, obteniéndolos con esa transformación:

$$\left|\,(x-x_0)\dots(x-x_n)\,\right|=$$

$$=\left(\frac{b-a}{2}\right)^{n+1}\left|\left(\frac{2}{b-a}\left(x-\frac{a+b}{2}-\left(x_0-\frac{a+b}{2}\right)\right)\right)\dots\left(\frac{2}{b-a}\left(x-\frac{a+b}{2}-\left(x_n-\frac{a+b}{2}\right)\right)\right)\right|=$$

$$= \left(\frac{b-a}{2}\right)^{n+1} \left| (y-y_0)\dots(y-y_n) \right|$$

Donde y y los y_i son los que corresponden a x, x_i a través de la transformación. Es decir, **las distancias se pueden calcular con sus correspondientes en $[-1,1]$, y luego multiplicar el resultado por ese factor de escala.**

Por tanto, ahora el problema se reduce a ver qué valores minimizan esa expresión en $[-1,1]$, es decir, encontrar los $\{y_i\}$ para los que se alcanza $\min\limits_{y_0,\dots,y_n} \{ \max\limits_{y\in[-1,1]} \{ | (y-y_0)\dots(y-y_n) | \} \}$. La solución está en los polinomios de Chebyshev, que minimizan esta norma en todos los polinomios de $[-1,1]$ y por tanto sus raíces serán los y_i.

Definición 9. Los **polinomios de Chebyshev** se definen por $T_0(x) = 1$, $T_1(x) = x$, y $T_n(x) = 2xT_{n-1}(x) - T_{n-2}(x)$ con $n \geq 2$.

Esta expresión recursiva no es demasiado útil, sin embargo, se tiene:

Proposición 6. *En el intervalo $[-1,1]$, se tiene $T_n(x) = \cos(n \cdot \arccos(x))$.*

Demostración. Por inducción sobre n. Hacen falta un par de casos base: $\cos(0) = 1 = T_0(x)$, y $\cos\arccos(x) = x = T_1(x)$. Ahora, suponiendo que se tiene para todos los $k < n$, con $n \geq 2$, entonces, con ayuda de algunas expresiones trigonométricas:

$$\cos(n\arccos(x)) = \cos(2\arccos(x) + (n-2)\arccos(x)) =$$

$$= \cos(2\arccos(x))\cos((n-2)\arccos(x)) - \sin(2\arccos(x))\sin((n-2)\arccos(x)) =$$

$$= (2\cos^2(\arccos(x)) - 1)\cos((n-2)\arccos(x)) - 2\sin(\arccos(x))\cos(\arccos(x))\sin((n-2)\arccos(x)) =$$

$$= (2\cos(\arccos(x)))[\cos(\arccos(x))\cos((n-2)\arccos(x)) - \sin(\arccos(x))\sin((n-2)\arccos(x))] -$$

$$- \cos((n-2)\arccos(x)) = 2x(\cos((n-1)\arccos(x))) - \cos((n-2)\arccos(x)) = 2xT_{n-1}(x) - T_{n-2}(x)$$

. □

Proposición 7. *Los ceros de $T_n(x)$ en el intervalo $[-1,1]$, se alcanzan en los puntos $x = \cos(\frac{\pi+2k\pi}{2n})$, con $k \in \{0,\dots,n-1\}$. Por tanto, todos sus ceros son esos.*

Demostración. Con la forma cerrada de la proposición anterior, los polinomios se anulan en $n \cdot \arccos(x) = \frac{\pi}{2} + k\pi$. Es decir, en los x tales que $x = \cos\frac{\pi+2k\pi}{2n}$. Para $k \in \{0,\dots,n-1\}$, se obtienen ángulos diferentes entre $\frac{\pi}{2n}$ y $\frac{(2n-1)\pi}{2n}$. Si k es mayor, los ángulos que se obtienen dan lugar al mismo coseno (además, en caso contrario, habría más de n raíces). □

Ahora veamos que estos polinomios no tienen una expresión parecida al que nosotros queríamos minimizar, que era mónico. Por ello, necesitaremos conocer el coeficiente de mayor grado de los polinomios para así normalizarlos.

Proposición 8. *El coeficiente de grado n en $T_n(x)$ es 2^{n-1} para $n \geq 1$.*

Demostración. Por inducción. En $T_1(x)$ es $1 = 2^0$. Suponiendo que se tiene para $1 \leq k < n$, veamos que, como $T_n(x) = 2xT_{n-1}(x) - T_{n-2}(x)$, el único coeficiente de grado n que aparece en el desarrollo es 2 veces el coeficiente de grado $n-1$ de T_{n-1}, luego es $2 \cdot 2^{n-2} = 2^{n-1}$, como queríamos. □

Ahora, lo que nos queda ver es que $\tilde{T_n} = \frac{T_n}{2^{n-1}}$ es el polinomio mónico de grado n con menor máximo en $[-1,1]$, y tendremos la solución al problema, ya que tiene las mismas raíces que T_n.

Observación 5. Los extremos de $T_n(x)$, y por tanto de $\tilde{T_n}(x)$, se alcanzan en los puntos de la forma $x = \cos(\frac{k\pi}{n})$ con $k \in \{0,\dots,n\}$.

Demostración. Basta con derivar: $T_n'(x) = -\sin(n \arc\cos(x)) \frac{-N}{\sqrt{1-x^2}}$. El segundo factor no se anula, solo el primero, cuando $n \arc\cos(x) = k\pi$, y por tanto $x = \cos\frac{k\pi}{n}$. A partir de $k = n+1$ se repiten los valores al ser función par el coseno (estaríamos sumando incrementos de π, dando lugar al ángulo opuesto). □

Observación 6. Los extremos de $T_n(x)$, ordenados de menor a mayor, son: $\tilde{x_i} = \cos(\frac{(n-i)\pi}{n})$, para $i \in \{0, \dots, n\}$ y además $T_n(\tilde{x_i}) = (-1)^{n-i}$.

Esto es así porque $\cos(n \arc\cos(\cos(\frac{(n-i)\pi}{n}))) = \cos((n-i)\pi)$.

Es decir, van oscilando entre -1 y 1. Asimismo, tenemos que $|\tilde{T_n}(x)| \leq \frac{1}{2^{n-1}}$ $\forall x \in [-1, 1]$. Ya solo queda un último paso:

Proposición 9. *Sea Π_n el conjunto de todos los polinomios mónicos. Entonces, fijado $P_n \in \Pi_n$, se tiene que $\max_{x\in[-1,1]} |P_n(x)| \geq \frac{1}{2^{n-1}}$. Por tanto, el que menor máximo tiene en $[-1, 1]$, mónico, es $\tilde{T}_n(x)$.*

Demostración. Supongamos que existiese $P \in \Pi_n$ con $\max_{x\in[-1,1]} |P_n(x)| < \frac{1}{2^{n-1}}$. En ese caso, sea $Q = \tilde{T}_n - P$. Observemos que Q es de grado $n-1$, como mucho, ya que ambos son mónicos. Sean $\{\tilde{x_i}\}_0^n$ los puntos $\tilde{x_i} = \cos(\frac{(n-i)\pi}{n})$ donde $\tilde{T}_n$ alcanza sus extremos. Entonces, está claro que, si $n-k$ es par: $Q(\tilde{x_k}) = \frac{1}{2^{n-1}} - P(\tilde{x_k}) \geq \frac{1}{2^{n-1}} - |P(\tilde{x_k})| > \frac{1}{2^{n-1}} - \frac{1}{2^{n-1}} = 0$. Por otra parte, si $n-k$ es impar: $Q(\tilde{x_k}) = -\frac{1}{2^{n-1}} - P(\tilde{x_k}) < -\frac{1}{2^{n-1}} + \frac{1}{2^{n-1}} = 0$. Es decir, Q cambia de signo en cada $\tilde{x_i}$.

Ahora, por el teorema de valores intermedios, $\exists x_i \in (\tilde{x_i}, \tilde{x}_{i+1})$, con $i \in \{0, \dots, n-1\}$, tales que $Q(x_i) = 0$. Es decir, Q tiene n raíces, lo que en un polinomio de grado, a lo sumo, $n-1$, solo se da si $Q = 0$, es decir, $P_n = \tilde{T}_n$. Como sabemos que $\max_{x\in[-1,1]} |\tilde{T_n}(x)| = \frac{1}{2^{n-1}}$, esto contradice el supuesto y no existe tal P_n. □

Por tanto, **los nodos óptimos de interpolación (Nodos de Chebyshev), en [-1,1], son las raíces del polinomio $\tilde{T}_{n+1}$, es decir, $y_i = \cos\frac{\pi+2i\pi}{2(n+1)}$, con $i \in \{0, \dots, n\}$.**

Teorema 10. *Sabiendo esto, en cualquier intervalo $[a, b]$, deshaciendo la transformación, los nodos óptimos son los $x_i = \frac{b-a}{2} y_i + \frac{a+b}{2} = \frac{b-a}{2} \cos\frac{\pi+2i\pi}{2n} + \frac{a+b}{2}$, para $i \in \{0, \dots, n\}$. Interpolando en esos nodos, se consigue la cota de error mínima:*

$$|f(x) - P_n(x)| \leq \frac{K_{n+1}}{(n+1)!} \frac{1}{2^n} \frac{(b-a)^{n+1}}{2^{n+1}} = \frac{K_{n+1}}{(n+1)!} \frac{(b-a)^{n+1}}{2^{2n+1}}$$

1.6. Interpolación de Hermite

Ahora nos planteamos si podemos generalizar la interpolación de Lagrange. En concreto, dados puntos $x_0, \dots, x_n$, y una función f, vamos a pedir un polinomio H que coincida no solo en valor, si no en derivadas:

Definición 10. Dados $x_0, \dots, x_n \in \mathbb{R}$, se define el **polinomio interpolador de Hermite de grado 2n+1** por $H_{2n+1}(x)$, que verifica:

1. $f(x_i) = H_{2n+1}(x_i)$
2. $f'(x_i) = H'_{2n+1}(x_i)$

Para $i \in \{0, \dots, n\}$. Se puede probar que es único, al tener $2n+2$ incógnitas y $2n+2$ ecuaciones independientes.

Para construir este polinomio, con una idea similar a la de Lagrange, disponemos de lo siguiente:

Definición 11. Fijados f y $x_0, \ldots, x_n$, siendo $l_i(x)$ los polinomios indicadores de Lagrange (de grado n, que valen 1 en x_i y 0 en los demás x_j), definimos los **polinomios de Hermite**:

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

$$K_i(x) = l_i^2(x)(x - x_i)$$

para $i \in \{0, \ldots, n\}$. Observemos que son de grado $2n + 1$.

Estos polinomios actúan de indicadores, pero de forma más potente:

Proposición 10. *Se verifica que $H_i(x_i) = 1$, $H_i(x_j) = 0$ con $i \neq j$, y $H_i'(x_j) = 0$ con cualquier $j \in \{0, \ldots, n\}$. Asimismo, se tiene que $K_i'(x_i) = 1$, $K_i'(x_j) = 0$ con $i \neq j$, y $K_i(x_j) = 0$ con cualquier $j \in \{0, \ldots, n\}$.*

Es decir, estos polinomios sirven de indicadores en los puntos o en las derivadas, y por tanto, se tiene:

$$H_{2n+1}(x) = \sum_{j=0}^{n} f(x_j)H_j(x) + f'(x_j)K_j(x)$$

Demostración. Está claro que $H_i(x_j) = l_i(x_j)^2(1-2l_i'(x_i)(x_j-x_i))$, que vale 0 al valerlo $l_i(x_j)$ si $j \neq i$, y, si no, vale $1 \cdot (1-0)$. Además, derivando, sigue que $H_i'(x) = 2l_i(x)l_i'(x)(1-2l_i'(x)(x-x_i)) - 2l_i'(x)l_i^2(x)$. Evaluando en x_j con $j \neq i$ se anula al hacerlo los l_i, y en x_i se tiene: $H_i'(x_i) = 2l_i'(x_i) - 2l_i'(x_i) = 0$. Para los K, veamos que $K_i(x_j) = l^2(x_j)(x_j - x_i)$, y una parte se anula si $j \neq i$ y la otra si $i = j$. La derivada, en x_j, es $K_i(x_j) = 2l_i(x_j)l_i'(x_j)(x_j - x_i) + l_i^2(x_j) = 0 + l_i^2(x_j)$, que vale 0 o 1 según el caso. □

Se tiene la siguiente cota de error:

Teorema 11. *Sean $x, x_0, \ldots, x_n \in [a, b]$ y $f \in \mathcal{C}^{2n+2}([a, b])$. Sea H el interpolador de hermite de grado $2n + 1$. Entonces, $\exists \xi \in (a, b)$ tal que:*

$$f(x) - H(x) = \frac{f^{(2n+2)}(\xi)}{(2n+2)!}(x - x_0)^2(x - x_1)^2 \ldots (x - x_n)^2$$

Y, mediante un método similar al de las diferencias dividas, se tiene:

Observación 7 (Algoritmo eficiente para el cálculo del interpolador de Hermite). Se puede obtener así:

1. Definimos $z \in \mathbb{R}^{2n+2}$ dado por $z_{2i} = z_{2i+1} = x_i$ para $i \in \{0, \ldots, n\}$, así como $Q_{i0} \in \mathbb{R}^{2n+2}$ dado por $Q_{2i,0} = Q_{2i+1,0} = f(x_i)$ para $i \in \{0, \ldots, n\}$.

2. Ahora, definimos $Q_{i,1} \in \mathbb{R}^{2n+2}$ dado por $Q_{2i+1,1} = f'(x_i)$, y $Q_{2i,1} = \frac{Q_{2i,0} - Q_{2i-1,0}}{z_{2i} - z_{2i-1}}$, para $i \in \{0, \ldots, n\}$.

3. Recursivamente, se define $Q_{i,j} = \frac{Q_{i,j-1} - Q_{i-1,j-1}}{z_i - z_{i-j}}$, para $j \in \{2, \ldots, 2n + 1\}$, e $i \in \{j, \ldots, 2n + 1\}$.

4. Finalmente, se tiene que:

$$H(x) = Q_{0,0} + Q_{1,1}(x - x_0) + Q_{2,2}(x - x_0)^2 + Q_{3,3}(x - x_0)^2(x - x_1) + \cdots +$$
$$+ Q_{2n+1,2n+1}(x - x_0)^2(x - x_1)^2 \ldots (x - x_{n-1})^2(x - x_n)$$

A. Algunos algoritmos de interpolación.

Observación 8 (Algoritmo para interpolación de Lagrange. Método de Newton.). Tenemos f y nodos $x_0, \ldots, x_n$.

1. Dispondremos de una matriz Q, de $n+1 \times n+1$ con las diferencias divididas.
2. Inicializamos la primera fila con los valores $f(x_k)$. Es decir, $Q_{0,j} = f(x_j)$.
3. Para cada fila siguiente, se tiene, si $j \in \{0, n-i\}$:

$$Q_{i,j} = \frac{Q_{i-1,j+1} - Q_{i-1,j}}{x_{j+i} - x_j}$$

   El elemento que queda en cada $Q_{k,1}$ es la diferencia dividida correspondiente.

Observación 9 (Algoritmo para el Spline cuadrático). Se da una función f y nodos $x_0 \ldots x_n$ de interpolación. El spline cuadrático se puede hallar como sigue.

1. Dispondremos de una colección $Q \subset (\mathbb{R}^3)^n$ de n 3-tuplas con los coeficientes de cada polinomio. Esta colección puede ser sobreescrita en cada paso, luego se puede implementar como una 3-tupla directamente.
2. Q se inicializa como sigue:

$$Q_1 = \begin{pmatrix} x_0^2 & x_0 & 1 \\ x_{1/2}^2 & x_{1/2} & 1 \\ x_1^2 & x_1 & 1 \end{pmatrix}^{-1} \begin{pmatrix} f(x_0) \\ f(x_{1/2}) \\ f(x_1)) \end{pmatrix}$$

   Realmente, la segunda línea de la matriz se corresponde con una condición arbitraria (en este caso que pase por el punto medio).
3. En este momento, Q tiene los coeficientes de la primera parábola. Podemos denotarlo por Q_1. Si necesitamos los coeficientes de otro intervalo, nos podemos valer de:

$$Q_k = \begin{pmatrix} x_k^2 & x_k & 1 \\ x_{k+1}^2 & x_{k+1} & 1 \\ 2x_k & 1 & 0 \end{pmatrix}^{-1} \begin{pmatrix} f(x_k) \\ f(x_{k+1}) \\ 2Q_{k-1,0}x_k + Q_{k-1,1} \end{pmatrix}$$

   De este modo los coeficientes de la k-ésima parábola son Q_k.

2. Integración numérica

En este capítulo se discutirá el cálculo de integrales por métodos numéricos. Antes, mostraremos brevemente la facilidad de calcular numéricamente valores de derivadas.

2.1. Derivación numérica

Queremos, dada una función f, calcular su derivada en x_0 con poco error. Para ello, nos podemos valer de las herramientas que proporciona el Teorema de Taylor. Una primera estimación es:

Proposición 11. *Si $h > 0$, $f'(x_0) = \frac{f(x_0+h)-f(x_0)}{h} + O(h)$*

Demostración. Sabemos, por Taylor, que $f(x_0+h) = f(x_0)+f'(x_0)(x_0+h-x_0)+O((x_0+h-x_0)^2) = f(x_0) + f'(x_0)h + O(h^2)$, luego se tiene, despejando. □

Una que converge más rápido es:

Proposición 12. *Si $h > 0$, $f'(x_0) = \frac{f(x_0+h)-f(x_0-h)}{2h} + O(h^2)$*

Demostración. $f(x_0 + h) = f(x_0) + f'(x_0)h + f''(x_0)\frac{h^2}{2} + O(h^3)$, y, por otro lado, $f(x_0 - h) = f(x_0) - f'(x_0)h + f''(x_0)\frac{h^2}{2} + O(h^3)$. Si restamos ambas expresiones y despejamos la derivada, se tiene lo que se quería. □

Esta técnica se puede obtener para conseguir numéricamente la derivada segunda, también.

Proposición 13. *Si $h > 0$, $f''(x_0) = \frac{f(x_0+h)+f(x_0-h)-2f(x_0)}{h^2} + O(h^2)$.*

Demostración. Es análoga a la anterior, pero con el polinomio hasta grado 4 y sumando en lugar de restar. □

2.2. Reglas de cuadratura

Como hemos visto, obtener derivadas numéricamente solo conociendo el valor de la función es bien sencillo, aplicando el teorema de Taylor. Ahora, fijados $a < b$, y dada una función f de la que conocemos algunos valores, nos interesa evaluar numéricamente la integral:

$$I(f) = \int_a^b f(x)dx$$

Esto es necesario ya que, en muchas ocasiones, f no tiene antiderivada como funciones elementales, o bien es sumamente difícil llegar hasta ella. También, es frecuente que f en sí misma no sea una función elemental, habiéndose obtenido de datos de la realidad, y solamente se sepa su valor en algunos puntos.

Una burda aproximación inicial podría ser estimar mediante reglas rectangulares: $I^R(f) = f(a)(b-a)$, $I^{R_f}(f) = f(b)(b-a)$, $I^{PM}(f) = f(\frac{a+b}{2})(b-a)$. Estas aproximaciones iniciales se pueden generalizar del siguiente modo:

Definición 12. Una **regla de cuadratura** de n nodos es una aplicación que, dada una función f, trata de estimar su integral de la siguiente manera:

$$I_n(f) = \alpha_0 f(x_0) + \alpha_1 f(x_1) + \cdots + \alpha_n f(x_n)$$

I está caracterizada por los **nodos de cuadratura** $x_0 \dots x_n \in [a, b]$ y los **pesos de cuadratura** $\alpha_0, \dots, \alpha_n$.

Definición 13. El **grado** de una regla de cuadratura I_n es el entero no negativo M que verifica que, $\forall p$ polinomio con $\deg(p) \leq M$, se tiene $I(p) = I_n(p)$, y que $\exists p$ con $\deg(p) = M+1$ tal que $I(p) \neq I_n(p)$.

Es decir, es el mayor grado de polinomios que integra correctamente.

Por ejemplo, la regla $I^R(f) = f(a)(b-a)$, tiene grado 0: solo integra bien las constantes. Vamos a ver que la regla del punto medio, $I^{PM} f(\frac{a+b}{2})(b-a)$, aunque solo tenga un nodo, es de grado 1. Efectivamente, para cualquier polinomio $p(x) = mx+n$, se tiene $f(p) = (m\frac{a+b}{2}+n)(b-a)$, y la integral de tal polinomio es $\frac{mb^2-ma^2}{2} + nb - na$, que coinciden sacando el factor común $(b-a)$. Es sencillo comprobar que falla para polinomios de grado 2. (Por ejemplo, x^2 en $[-1,1]$, tendría por esta regla integral 0, y sabemos que no es cierto.).

Si los polinomios de interpolación estiman las funciones, cabe esperar que su integral sea parecida. Esto da lugar a la regla:

Definición 14 (Regla de cuadratura por interpolación de Lagrange)**.** Dados los $x_0, \ldots, x_n \in [a,b]$, la regla de interpolación de Lagrange es:

$$I_n(x) = \left(\int_a^b l_0(x)dx\right) f(x_0) + \cdots + \left(\int_a^b l_n(x)dx\right) f(x_n)$$

Donde $l_i(x)$ es el polinomio que se usaba en interpolación de Lagrange (valiendo 0 en los x_j con $j \neq i$, y 1 en x_i).

Si los puntos x_i están equiespaciados, se conoce como **regla de Newton-Cotes**.

Por linealidad de la integral, esta regla de cuadratura consiste efectivamente en integrar el interpolador de Lagrange. Es sencilla de aplicar dado que no es difícil integrar polinomios. Por ejemplo, tomando $x_0 = a$ y $x_1 = b$, se convierte en lo que se denomina **cuadratura por trapecios**, I^T.

Teorema 12. *La regla I_n de cuadratura, por interpolación de Lagrange, es de grado $\geq n$.*

Demostración. Si p es un polinomio de grado $\leq n$, por unicidad del interpolador de Lagrange, se tiene $p_n = p$, de modo que $I(p) = I(p_n) = I_n(p)$. □

Proposición 14. *Cualquier regla de cuadratura $I_n(f)$ es una aplicación lineal.*

Demostración. $I(\lambda f + \mu g) = \sum_{i=0}^n \alpha_i(\lambda f + \mu g)(x_i) = \sum_{i=0}^n \alpha_i(\lambda f(x_i) + \mu g(x_i)) = \lambda \sum_{i=0}^n \alpha_i f(x_i) + \mu \sum_{i=0}^n \alpha_i g(x_i) = \lambda I_n(f) + \mu I_n(g)$. □

Proposición 15. *Dada una base B de un espacio de polinomios $\mathbb{R}_{\leq n}[X]$, si cualquier regla de cuadratura $I_n(f)$ integra correctamente todos los elementos de B, entonces es de grado $\geq n$.*

Demostración. Si ponemos $B = \{p_0, \ldots, p_n\}$, entonces dado p de grado $\leq n$, se tiene $p = \sum_{i=0}^n \lambda_i p_i$. Se tiene que $I(p) = \sum_{i=0}^n \lambda_i I(p_i) = \sum_{i=0}^n \lambda_i I_n(p_i) = I_n(\sum_{i=0}^n \lambda_i p_i) = I_n(p)$. Hemos usado la linealidad de la integral y la hipótesis, así como la proposición anterior. □

Esto nos permite obtener unas ecuaciones que ha de verificar una regla de cuadratura:

Observación 10 (Coeficientes indeterminados). Dada $I_n(f)$ una regla de cuadratura en n nodos, si queremos que tenga grado $\geq M$, deberá ser correcta en los polinomios $\{1, X, X^2, \ldots, X^m\}$, dado que forman base del espacio que se quiere. Por tanto, debe darse:

$$\begin{cases} I_n(1) = \alpha_0 + \alpha_1 + \cdots + \alpha_n = b - a \\ I_n(x) = \alpha_0 x_0 + \alpha_1 x_1 + \cdots + \alpha_n x_n = \frac{b^2 - a^2}{2} \\ I_n(x^2) = \alpha_0 x_0^2 + \alpha_1 x_1^2 + \cdots + \alpha_n x_n^2 = \frac{b^3 - a^3}{3} \\ \vdots \\ I_n(x^m) = \alpha_0 x_0^m + \alpha_1 x_1^m + \cdots + \alpha_n x_n^m = \frac{b^{m+1} - a^{m+1}}{m+1} \end{cases}$$

Estas ecuaciones permiten comprobar el grado de una regla, que será el número de ecuaciones, partiendo de la primera, que se verifican. Asimismo, resolviendo el sistema para un m dado, se pueden obtener reglas del grado deseado.

Proposición 16. *La regla de cuadratura I_n de $n+1$ nodos distintos dos a dos, y de grado $\geq n$, es única.*

Demostración. Sea I_n una tal regla de cuadratura. Como sabemos, sus pesos han de verificar el sistema anterior, que consta de $n+1$ ecuaciones con $n+1$ incógnitas, y además su matriz de coeficientes es de Vandermonde, luego si los x_i son distintos dos a dos, su determinante es no nulo y existe una solución única.

Corolario. La única regla de cuadratura de $n+1$ nodos y grado $\geq n$ es la de interpolación de Lagrange.

A continuación veremos la **regla de Simpson.** Se trata de una regla de cuadratura por interpolación de Lagrange en 3 nodos, que curiosamente alcanza a tener grado 3 (sabemos que debía tener por lo menos 2).

Definición 15. La **regla de Simpson** está dada por:

$$I^S(f) = \frac{b-a}{6} f(a) + \frac{2(b-a)}{3} f\left(\frac{a+b}{2}\right) + \frac{b-a}{6} f(b)$$

Se consigue por interpolación de Lagrange en los nodos a, b y su punto medio.

Es fácil comprobar, con las ecuaciones previas, que es de grado 3. Por conveniencia, además, se puede comprobar en una versión modificada de esas ecuaciones, con la base $\{1, x - \frac{a+b}{2}, x^2, (x - \frac{a+b}{2})^3\}$, dado que se anulan algunos términos. Esta idea de emplear otras bases puede ser útil al verificar reglas.

Observación 11 (Un método alternativo para obtener los coeficientes.). Otra idea para obtener los coeficientes de cuadratura, hasta grado N de precisión, es usar el desarrollo de Taylor de la función f a cuadrar.

1. Tomamos el desarrollo de Taylor de f en $c = \frac{a+b}{2}$ el punto medio del recinto de integración. Lo integramos para obtener el desarrollo de su integral, obteniendo una expresión en base de potencias de $(x - c)$:

$$\int_a^b f dx \approx \sum_{k=0}^{n} \frac{f^{(2k)}(c)}{(2k)!} \left(\frac{b-a}{2}\right)^{2k+1} \frac{2}{2k+1}$$

   Cabe observar que al ser c el punto medio, solo quedan las potencias de grado impar, al ser $(x-c)^k$ simétrico respecto de c si k no es par.

2. Ahora, con la misma idea que anteriormente, tomamos la regla de cuadratura como $\sum_0^n \alpha_i f(x_i)$ y evaluamos cada f en x_i, también mediante su desarrollo en serie:

$$I_n(f) \approx \sum_{i=0}^{n} \frac{f^{(k)}(c)}{k!}(x_i - c)^k$$

3. Los coeficientes se pueden obtener igualando los primeros N términos de cada expansión (dividiendo por la derivada correspondiente y por el denominador $k!$, para evitarlos):

$$\sum_{i=0}^{n} \alpha_i (x_i - c)^k = \left(\frac{b-a}{2}\right)^{k+1} \frac{2}{k+1}$$

Si k es par, o, si es impar:

$$\sum_{i=0}^{n} \alpha_i (x_i - c)^k = 0$$

La ventaja de este sistema es que la mitad de las ecuaciones son homogéneas.

2.3. Cuadratura Gaussiana

La idea de la regla Gaussiana $I_n(f) = \sum_{i=0}^n \alpha_i f(x_i)$ es encontrar, fijado n, los α_i e x_i que maximizan el grado de la regla. En particular, encontrar los valores que hacen que dicho grado sea $2n+1$, el máximo posible. ($2n+2$ ecuaciones, $2n+2$ incógnitas). Para obtener la regla como tal, también se impondrá que ningún α_i sea nulo y que no haya dos x_i iguales. (Si no, degenera en menos puntos.)

En cierto sentido, este problema se asemeja al de interpolación de Chebyshev, y la primera idea para abordarlo es similar:

Proposición 17. *Realizado el cambio lineal de escala* $t = \frac{x - \frac{a+b}{2}}{\frac{b-a}{2}}$, *que traslada linealmente* $[a,b]$ *en* $[-1,1]$, *se tiene:*

$$\int_a^b f(x)dx = \int_{-1}^{1} f\left(\frac{b-a}{2}t + \frac{a+b}{2}\right)\frac{b-a}{2}dt$$

El problema se reduce, entonces, a encontrar la regla $I_n(f) = \sum_{i=0}^n \alpha_i f(x_i)$ *que maximiza el grado, si* f *es una función de* $[-1,1]$.

Por tanto, una vez encontrados los α_i' ***y*** x_i' ***para*** $[-1,1]$***, se podrán cambiar a*** $[a,b]$ ***deshaciendo el cambio:***

1. $x_i = \frac{b-a}{2}x_i' + \frac{a+b}{2}$
2. $\alpha_i = \frac{b-a}{2}\alpha_i'$

Simplemente hemos tenido en cuenta el aspecto de la integral que hemos cuadrado en $[-1,1]$.

Este cambio es muy conveniente, ya que podemos hacer las comprobaciones con las funciones $\{1, x, \ldots, x^n\}$ y la mitad de las ecuaciones (polinomios impares) se volverán homogéneas, al estar el intervalo centrado en 0. Ahora basta con hallarlas para ese intervalo:

Observación 12 (Cuadraturas Gaussianas de órdenes bajos.). Estas han sido obtenidas para $[-1,1]$ con el método de coeficientes indeterminados, resolviendo los sistemas:

1. Para grado 1, se tiene $\alpha_0 = 2$, $x_0 = 0$. (La regla del punto medio)
2. Para dos puntos, es decir, grado 3, se tiene $\alpha_0 = \alpha_1 = 1$, $x_0 = \frac{1}{\sqrt{3}}$, $x_1 = -\frac{1}{\sqrt{3}}$.
3. Para tres puntos (grado 5), $\alpha_0 = \alpha_2 = \frac{5}{9}$, $\alpha_1 = \frac{8}{9}$, $x_0 = -\sqrt{\frac{3}{5}}$, $x_2 = \sqrt{\frac{3}{5}}$, $x_1 = 0$.

Grados superiores dan lugar a nodos y coeficientes más complicados.

2.4. Polinomios de Legendre

Como hemos visto anteriormente, hallar los nodos y pesos para una cuadratura Gaussiana I_n se puede complicar inmensamente en cuanto n no es pequeño, dado que hay que resolver un sistema no lineal. La solución al problema viene de mano de los **polinomios de Legendre:**

Definición 16. Se definen los **polinomios de Legendre** $\{p_k\}_{k=0}^{\infty}$ a través de la relación recursiva:

$$\begin{cases} p_0(x) = 1 \\ p_1(x) = x \\ p_k(x) = (x - B_k)p_{k-1}(x) - C_k p_{k-2}(x) \quad (k \geq 2) \end{cases}$$

Donde $B_k = \frac{\int_{-1}^{1} x(p_{k-1}(x))^2 dx}{\int_{-1}^{1}(p_{k-1}(x))^2 dx}$, y $C_k = \frac{\int_{-1}^{1} x p_{k-2}(x) p_{k-1}(x) dx}{\int_{-1}^{1}(p_{k-2}(x))^2 dx}$.

Así, se tiene que cada polinomio $p_k(x)$ es de grado k, y por tanto, $\{p_k\}_0^n$ es base de los polinomios hasta grado n.

Es importante el siguiente resultado:

Proposición 18. *Los polinomios de Legendre son ortogonales en* $[-1, 1]$*, es decir,* $\int_{-1}^{1} p_k(x)p_j(x)dx = 0$ *siempre que* $j \neq k$*.*

Demostración. Por inducción. Es fácil ver que $\{p_0, p_1, p_2\}$ son ortogonales, calculando a mano las 3 integrales necesarias. Ahora, supongamos que lo son los polinomios $\{p_0, \ldots, p_{k-1}\}$, con $k \geq 3$. Hay que demostrar que p_k lo es individualmente a cada uno de ellos.

En primer lugar, $\int_{-1}^{1} p_k p_{k-1} dx = \int_{-1}^{1} x p_{k-1}^2 dx - B_k \int_{-1}^{1} p_{k-1}^2 - C_k \int_{-1}^{1} p_{k-2} p_{k-1} dx = \int_{-1}^{1} x p_{k-1}^2 dx - B_k \int_{-1}^{1} p_{k-1}^2$ por hipótesis. Ahora, esta integral vale 0 si y solo si $B_k = \frac{\int_{-1}^{1} x(p_{k-1}(x))^2 dx}{\int_{-1}^{1}(p_{k-1}(x))^2 dx}$, lo que se tiene por definición.

En segundo lugar, $\int_{-1}^{1} p_k p_{k-2} dx = \int_{-1}^{1} x p_{k-1} p_{k-2} - B_k \int_{-1}^{1} p_{k-1} p_{k-2} dx - C_k \int_{-1}^{1} p_{k-2}^2 dx = \int_{-1}^{1} x p_{k-1} p_{k-2} - C_k \int_{-1}^{1} p_{k-2}^2 dx$ por hipótesis, y una vez más se anula por construcción de C_k.

Finalmente, si $j < k-2$, tenemos $\int_{-1}^{1} p_k p_j dx = \int_{-1}^{1} x p_{k-1} p_j dx - B_k \int_{-1}^{1} p_{k-1} pj dx - C_k \int_{-1}^{1} p_{k-2} p_j dx = \int_{-1}^{1} x p_{k-1} p_j dx$, por hipótesis. Ahora, observemos que $xp_j(x)$ es de grado, a lo sumo, $k-2$, por tanto, puede ser expresado como combinación lineal $xp_j(x) = \sum_{i=0}^{j} \alpha_i p_i(x)$ en la base de polinomios de Legendre, y entonces, $\int_{-1}^{1} x p_{k-1} p_j dx = \int_{-1}^{1} p_{k-1} \sum_{i=0}^{j} \alpha_i p_i(x) dx = \sum_{i=0}^{j} \alpha_i \int_{-1}^{1} p_i p_k = 0$ por hipótesis. □

Haremos uso del siguiente lema:

Lema 1. *Si* $\{p_k(x)\}_{k=0}^{n}$ *son ortogonales en* $[a, b]$*, cada uno de grado* k*, entonces* $p_k(x)$ *tiene exactamente* k *raíces distintas en* $[a, b]$*.*

Por tanto, los polinomios de Legendre tienen todas sus raíces distintas en $[-1, 1]$.

Teorema 13. *Los nodos $\{x_i\}_{i=0}^n$ de la cuadratura Gaussiana $I_n(f)$ (de grado $2n+1$), en el intervalo $[-1,1]$ son las raíces del polinomio de Legendre $p_{n+1}(x)$. Los pesos $\{\alpha_i\}_{i=0}^n$ se pueden hallar por interpolación polinómica en esos nodos: $\alpha_i = \int_{-1}^1 l_i(x)dx$.*

Demostración. Sea P un polinomio de grado menor o igual que $2n+1$. Por división euclídea, $P = Qp_{n+1} + R$, donde Q, R tienen grado menor o igual que n. Sean $\{x_i\}_0^n$ las raíces de p_{n+1}. Entonces, $P(x_i) = R(x_i)$ por construcción. Si tomamos la regla $I_n(f) = \sum_{i=0}^n \alpha_i f(x_i)$, con los α_i descritos en el teorema, sabemos que tiene grado n por lo menos, por tanto, $\int_{-1}^1 Rdx = \sum_{i=0}^n \alpha_i R(x_i) = \sum_{i=0}^n \alpha_i P(x_i) = I_n(P)$.

Por otra parte, se tiene que $\int_{-1}^1 Pdx = \int_{-1}^1 Qp_{n+1}dx + \int_{-1}^1 Rdx = \int_{-1}^1 Rdx$. al ser $\int_{-1}^1 Qp_{n+1}dx = 0$, lo que se ve claro si ponemos $Q = \sum_{i=0}^n \beta_i p_i(x)$. Por tanto, finalmente, $I_n(P) = \int_{-1}^1 Pdx$, y como P era arbitrario, deducimos que esta regla es Gaussiana y por tanto los nodos y los pesos son los descritos en el teorema. □

Por tanto, con este método hallamos fácilmente nodos y pesos, y para un intervalo $[a,b]$ generalizado, basta con aplicar el cambio de variable conocido.

2.5. Errores de cuadratura

A continuación queremos estimar que error se comete con cada regla. No hay una forma generalizada para el error, pero se puede hallar individualmente con algunas técnicas.

Teorema 14 (Valor Medio Generalizado)**.** *Sean $g, \alpha : [a,b] \to \mathbb{R}$, con $\alpha(x) > 0$, continuas. Entonces, $\exists \xi \in [a,b]$ tal que:*

$$\int_a^b g(x)\alpha(x)dx = g(\xi)\int_a^b \alpha(x)dx$$

(Obsérvese que para $\alpha = 1$ y $g(x) = f'(x)$ se tiene el caso particular del teorema de valor medio más habitual)

Demostración. Al ser g continua en un compacto, alcanza $g_0 = \min_{x\in[a,b]} g(x)$ y $G_0 = \max_{x\in[a,b]} g(x)$ en $[a,b]$. Como $\alpha(x) > 0$, se puede multiplicar la desigualdad para obtener $g_0\alpha(x) \le \alpha(x)g(x) \le G_0\alpha(x)$. Integrando, resulta que $g_0 \int_a^b \alpha(x)dx \le \int_a^b \alpha(x)g(x)dx \le \int_a^b G_0\alpha(x)dx$, con lo que $g_0 \le \frac{\int_a^b \alpha(x)g(x)dx}{\int_a^b \alpha(x)dx} \le G_0$. Por el teorema de valores intermedios, al ser g continua, hay $\xi \in (a,b)$ donde $g(\xi) = \frac{\int_a^b \alpha(x)g(x)dx}{\int_a^b \alpha(x)dx}$. □

Proposición 19 (Cota de error: Regla del punto medio)**.** *Si $f \in \mathcal{C}^2([a,b])$, se tiene:*

$$E = \int_a^b f(x)dx - (b-a)f(\frac{a+b}{2}) = \frac{(b-a)^3}{24}f''(\xi)$$

Para $\xi \in [a,b]$.

Demostración. Sea $c = \frac{a+b}{2}$. Observemos que el error que estamos buscando se puede poner como: $E = \int_a^b (f(x) - f(c))dx = \int_a^b (f(x) - f(c)) - f'(c)(x-c)dx$. Para el ultimo paso usamos que $(x-c)$ integra 0 al ser impar por c.

Con el fin de utilizar el teorema anterior, definimos $g(x) := \frac{f(x)-f(c)-f'(c)(x-c)}{(x-c)^2}$ fuera de c y $g(x) = \frac{f''(c)}{2}$ en c. Esta función es continua dado que si calculamos su límite en c, coincide con el valor que le hemos dado. Ahora el error se reduce a hallar $\int_a^b g(x)(x-c)^2dx$. Por el teorema de taylor, $\exists \xi \in [a,b]$ tal que $f(x) - f(c) - f'(c)(x-c) = \frac{f''(\xi)(x-c)^2}{2}$, luego, si $x \neq c$, dividendo, queda que $g(x) = \frac{f''(\xi)}{2}$. Por otra parte, si $x = c$, es trivial que $g(x) = \frac{f''(\xi)}{2}$ para $\xi = c$.

Así, finalmente, por el teorema anterior, tenemos que $E = g(\eta)\int_a^b (x-c)^2 dx$, con $\eta \in [a,b]$, y por el resultado que hemos obtenido en el párrafo anterior, se tiene $E = \frac{f''(\xi)}{2}\int_a^b (x-c)^2 dx = \frac{(b-a)^3}{24} f''(\xi)$. □

Proposición 20 (Cota de error: Regla del trapecio). *Si $f \in \mathcal{C}^2([a,b])$, se tiene:*

$$E = \int_a^b f(x)dx - I^t(f) = -\frac{(b-a)^3}{12} f^{(2)}(\xi)$$

Para $\xi \in [a,b]$.

Proposición 21 (Cota de error: Regla de Simpson). *Si $f \in \mathcal{C}^4([a,b])$, se tiene:*

$$E = \int_a^b f(x)dx - I^S(f) = \int_a^b f(x)dx - \frac{b-a}{6}(f(a)+f(b)) - \frac{2(b-a)}{3} f(\frac{b+a}{2}) = -\frac{(b-a)^5}{2880} f^{(4)}(\xi)$$

Para $\xi \in [a,b]$.

A fin de demostrar estas proposiciones, así como dar cotas de error para muchas otras reglas, vamos a ver un ejemplo de como se llevaría a cabo. Supongamos que $I_n(f)$ es la regla de Newton-Cotes (interpolación de Lagrange equiespaciada) en $n+1$ puntos. Entonces, su error es $\int_a^b f(x)dx - I_n(f) = \int_a^b f(x)dx - \sum_{i=0}^n (\int_a^b l_i(x)dx) f(x_i) = \int_a^b (f(x) - \sum_{i=0}^n l_i(x) f(x_i)dx) = \int_a^b \frac{f^{(n+1)}(\xi(x))}{(n+1)!} \prod_{k=0}^n (x-x_j)dx$. Por el teorema de Valor Medio Generalizado, para ciertas condiciones (quizás multiplicando por -1 algunos de los elementos del productorio para que sea positivo), se tiene:

$$\int_a^b f(x)dx - I_n(f) = \frac{f^{(n+1)}(\xi)}{(n+1)!} \int_a^b \prod_{i=0}^n (x-x_i)dx$$

Para $\xi \in [a,b]$.

Si nos aprovechamos de esta cota de error, podemos enunciar el siguiente teorema:

Teorema 15 (Cotas de error para cuadraturas Newton-Cotes). *Sea $x_0 = a$, $h = \frac{b-a}{n}$, $x_k = x_0 + hk$ para $k \in \{1,\dots,n\}$. Sea $I_n(f)$ la regla de Newton-Cotes en esos puntos. Entonces, $\exists \xi \in [a,b]$ tal que:*

$$\int_a^b f(x)dx - I_n(f) = \begin{cases} \frac{h^{n+2} f^{(n+1)}(\xi)}{(n+1)!} \int_0^n t(t-1)\dots(t-n)dt & \text{si } n \text{ es impar} \\ \\ \frac{h^{n+3} f^{(n+2)}(\xi)}{(n+2)!} \int_0^n t^2(t-1)\dots(t-n)dt & \text{si } n \text{ es par} \end{cases}$$

Debe ocurrir que $f \in \mathcal{C}^k([a,b])$, con k el orden de la derivada que estamos calculando. De aquí surgen las cotas anteriores y se pueden deducir las de órdenes mayores.

Teorema 16 (Cotas de error para cuadraturas Gaussianas). *Sea $[a,b]$ un intervalo, $\{x_i\}_0^n$ los nodos gaussianos en ese intervalo, y $\{\alpha_i\}_0^n$ los pesos gaussianos. Sea $f \in \mathcal{C}^{2n+2}([a,b])$. Entonces, $\exists \xi \in [a,b]$ tal que:*

$$\int_a^b f dx - I_n(f) = \frac{f^{(2n+2)}(\xi)}{(2n+2)!} \int_a^b (x-x_0)^2 \dots (x-x_n)^2 dx = \frac{f^{(2n+2)}(\xi)}{(2n+2)!} \left(\frac{b-a}{2}\right)^{2n+3} \int_{-1}^1 (x-y_0)^2 \dots (x-y_n)^2 dx$$

Donde los y_k son los equivalentes de los nodos gaussianos, en $[-1,1]$.

Demostración. Usaremos $Q(x)$, el interpolante de Hermite de grado $2n+1$ de f en los x_i. Lo que tenemos que observar es que, al ser de grado $2n+1$, se tiene: $\int_a^b Q dx = I_n(Q) = I_n(f)$ al coincidir en los puntos. Por tanto, el error se reduce a calcular $E = \int_a^b (f-Q)dx$. Si utilizamos la expresión del error de interpolación de Hermite: $E = \int_a^b \frac{f^{(2n+2)}(\xi(x))}{(2n+2)!}(x-x_0)^2 \dots (x-x_n)^2 dx$, y como el producto de cuadrados es positivo y el integrando es continuo, usando el teorema de valor medio generalizado, se tiene el resultado del teorema. □

2.6. Reglas compuestas

Como vemos, las cotas anteriores son malas cuanto mayor es el intervalo. Una solución es mejorar la regla de cuadratura, pero otra, mucho más sensata, es reducir los intervalos:

Definición 17. Sea $\Delta = \{x_0, \dots, x_n\}$ con $x_i < x_j$ si $i < j$, $x_0 = a$, $x_n = b$, una partición de $[a,b]$. Dada una regla de cuadratura $I_[a,b]$ en el intervalo $[a,b]$, se puede definir su **regla compuesta** teniendo en cuenta que $\int_a^b f dx = \sum_{i=1}^n \int_{x_{i-1}}^{x_i} f dx$, y por tanto, si estimamos cada una de estas sub-integrales con la regla, tenemos:

$$I^c_{[a,b]}(f) = \sum_{i=1}^n I_{[x_{i-1},x_i]}(f)$$

Por ejemplo, la integral en $[a,b]$ se puede estimar con la regla del punto medio compuesta $I^{PMC}(f) = \sum_{i=1}^n (x_i - x_{i-1}) f(\frac{x_i + x_{i-1}}{2})$.

Para relacionar el error de una regla simple con el de una compuesta, demostraremos el siguiente *teorema del valor medio discreto:*

Teorema 17. *Sea $g : [a,b] \to \mathbb{R}$ continua, $\alpha_i \geq 0$ y $\xi_i \in [a,b]$. Se tiene que $\exists \xi \in [a,b]$ con:*

$$\sum_{i=1}^n \alpha_i g(\xi_i) = g(\xi) \sum_{i=1}^n \alpha_i$$

Demostración. Es muy parecida al teorema del valor medio generalizado. Si todos los ξ_i son iguales, el teorema sigue trivialmente. Si no, sea $G(x) = \sum_{i=1}^n \alpha_i (g(\xi_i) - g(x))$, continua al serlo g. Sea $g = \min_{x\in[a,b]} g(x)$ y $G = \max_{x\in[a,b]} g(x)$. Al ser continua en intervalo cerrado, $g, G \in [a,b]$. Ahora, veamos que $G(g) > 0$, dado que por construcción, todos los términos son no negativos y al menos uno no es nulo al no ser todos los ξ_i iguales. Análogamente $G(G) < 0$, luego $\exists \xi \in I(g,G) \subset (a,b)$ tal que $G(\xi) = 0$ y se tiene lo que se quería despejando. □

Proposición 22. *Si Δ es equiespaciada con separación h, el error de la regla de punto medio compuesto es:*

$$\int_a^b f dx - I^{PMC}(f) = \frac{f''(\xi)(b-a)}{24} h^2$$

Y por tanto disminuye cuadráticamente con la longitud de la partición.

Demostración. $\int_a^b f dx - I^{PMC}(f) = \sum_{i=1}^n \int_{x_{i-1}}^{x_i} f dx - I^{PM}_{[x_{i-1},x_i]}(f) = \sum_{i=1}^n \frac{f''(\xi_i)h^3}{24} = \frac{f''(\xi)h^3}{24} \sum_{i=1}^n 1 = \frac{f''(\xi)h^3 n}{24} = \frac{f''(\xi)(b-a)}{24} h^2$. Hemos usado el teorema anterior para unificar el ξ. □

En general, esta estrategia sirve para cualquier regla compuesta derivada a partir de una regla simple de la que se conoce el error.

3. Resolución de ecuaciones no lineales

El objetivo ahora es, dada una función real $f : D \subset \mathbb{R} \to \mathbb{R}$, hallar $x_0 \in D$ tal que $f(x_0) = 0$, es decir, obtener o estimar una raíz de f.

3.1. Método de bisección

Para este método supondremos que f es continua en su dominio. Sabemos, por el teorema de valores intermedios, que para todo $[a, b] \subset D$ con $f(a)f(b) < 0$, hay una raíz en $[a, b]$. El método entonces puede llevarse a cabo como sigue:

Definición 18. Los pasos del método de bisección son:

1. Tomar $x_0, y_0 \in [a, b] \subset D$ con $f(x_0)f(y_0) < 0$.
2. Definir $c_0 = \frac{x_0+y_0}{2}$ el punto medio. Ahora:
   Si $f(c_0) = 0$ hemos acabado.
   Si $f(x_0)f(c_0) < 0$, sean $x_1 := x_0$ e $y_1 := c_0$.
   Si $f(x_0)f(c_0) > 0$, sean $x_1 := c_0$ e $y_1 := y_0$.
3. Ahora tenemos un nuevo subintervalo en cuyos extremos la función tiene extremos opuestos, con $(y_1 - x_1) = \frac{(y_0-x_0)}{2}$.
4. En general, si tenemos x_k e y_k, definimos $c_k = \frac{x_k+y_k}{2}$ el punto medio. Ahora:
   Si $f(c_k) = 0$ hemos acabado.
   Si $f(x_k)f(c_k) < 0$, sean $x_{k+1} := x_k$ e $y_{k+1} := c_k$.
   Si $f(x_k)f(c_k) > 0$, sean $x_{k+1} := c_k$ e $y_{k+1} := y_k$.
5. Tenemos el subintervalo $[x_{k+1}, y_{k+1}]$, con una raíz, tal que $(y_{k+1} - x_{k+1}) = \frac{y_k-x_k}{2} = \frac{y_0-x_0}{2^{k+1}}$.

Repitiendo tantas veces como se desee, hallamos un intervalo de longitud tan pequeña como se quiera que contiene a la raíz. Tomando un elemento de ese intervalo $[x_n, y_n]$, estimamos la raíz con error menor que $\frac{y_0-x_0}{2^n}$.

Este método asegura encontrar una raíz con el error mencionado anteriormente, pero únicamente una. Además, la función debe ser continua, y el método es lento (se converge a la raíz considerablemente lento comparado con otros métodos). Lo bueno es que la raíz está acotada.

3.2. Método de la secante

La idea es reducir el encontrar la raíz de f a encontrarla en una función lineal $ax + b = 0$. Para ello, usaremos el polinomio interpolador.

Definición 19. Los pasos del método de la secante son:

1. Tomar $x_0 \in D$. Si $f(x_0) = 0$, hemos acabado.
2. Tomar $x_1 \in D$. Si $f(x_1) = 0$, hemos acabado. Si $f(x_1) = f(x_0)$, habrá que tomar otro x_1.
3. Se construye el interpolador de f en esos puntos, $p(x) = f(x_0) + \frac{f(x_1)-f(x_0)}{x_1-x_0}(x - x_0)$. Se toma su raíz: $x_2 := x_0 - \frac{f(x_0)(x_1-x_0)}{f(x_1)-f(x_0)}$. Para que tenga raíz es necesario, como dijimos, que $f(x_1) = f(x_0)$.

4. En general, teniendo x_{k-1} y x_k, se obtiene $x_{k+1} = x_{k-1} - \frac{f(x_{k-1})(x_k - x_{k-1})}{f(x_k) - f(x_{k-1})}$. Si $f(x_k) = f(x_{k-1})$ no tiene raíz, y se puede tomar, por ejemplo x_{k-2} en lugar de x_{k-1} para el cálculo.

5. Tras suficientes n iteraciones, x_n es próximo a la raíz de f.

Es más rápido convergiendo que el de bisección, pero tiene algunos problemas, por ejemplo el ya mencionado (que $f(x_k) = f(x_{k-1})$), o que algún x_k durante el proceso caiga fuera del dominio de f.

Proposición 23 (Convergencia del método de la secante.)**.** *Si realizamos el método de la secante en el intervalo $[a, b]$ con $f(a)f(b) < 0$, de tal modo que $\exists \alpha \in [a, b]$ con $f(\alpha) = 0$, y el método de la secante converge, se tiene que $\lim_{n\to\infty} x_n = \alpha$. Si $f \in \mathcal{C}^2([a, b])$, y denotamos $e_n = |x_n - \alpha|$ en este caso, se tiene que:*

$$e_{n+1} = \frac{f''(\eta)}{2f'(\xi)} e_n e_{n-1}$$

Para $\eta, \xi \in [a, b]$, y por tanto, se tiene:

$$\lim_{n\to\infty} \frac{e_{n+1}}{e_n} = 0$$

*Es decir, tiene una convergencia **superlineal**.*

Demostración. Si converge, lo hace a una raíz dado que $\lim_{n\to\infty} x_n = \lim_{n\to\infty} x_n - \frac{f(\lim_{n\to\infty} x_n)}{\lim_{n\to\infty} f[x_n, x_{n-1}]} \implies 0 = \frac{f(\lim_{n\to\infty} x_n)}{\lim_{n\to\infty} f[x_n, x_{n-1}]} \implies f(\lim_{n\to\infty} x_n) = 0$. Para obtener la igualdad del error, veamos que $x_{n+1} - \alpha = x_n - \alpha - \frac{f(x_n) - \alpha}{f[x_n, x_{n-1}]} = (x_n - \alpha)(1 - \frac{f(x_n) - f(\alpha)}{(x_n - \alpha) f[x_n, x_{n-1}]}) = (x_n - \alpha)(1 - \frac{f[x_n, \alpha]}{f[x_n, x_{n-1}]}) = (x_n - \alpha)\frac{f[x_n, x_{n-1}] - f[x_n, \alpha]}{f[x_n, x_{n-1}]} = (x_n - \alpha)(x_{n-1} - \alpha)\frac{f[x_n, x_{n-1}, \alpha]}{f[x_n, x_{n-1}]}$. Por tanto, se tiene que $e_{n+1} = e_n e_{n-1} |\frac{f[x_n, x_{n-1}, \alpha]}{f[x_n, x_{n-1}]}| = e_n e_{n-1} \frac{f''(\eta)}{2f'(\xi)}$, como se quería. En el último paso usamos la expresión de las diferencias divididas que surge de combinar la observación 4 con el Teorema 5. □

3.3. Métodos de punto fijo. Método de Newton.

Se basan en el siguiente algoritmo para encontrar un punto fijo ($x \in D$ con $f(x) = x$) de una función:

Definición 20 (Algoritmo para hallar un punto fijo.)**.** Dada $g : D \subset \mathbb{R} \to D' \subset D$, y $x_0 \in D$, se define la sucesión $\{x_n\}_0^\infty$ por $x_{n+1} = g(x_n)$. Así, cada término se obtiene iterando la función sobre el anterior.

Proposición 24. *Si g es continua, y la sucesión $\{x_n\}$ del método anterior tiene límite, es decir, si $\exists \lim_{n\to\infty} x_n = P$, entonces $g(P) = P$.*

Demostración. Tenemos que $g(P) = g(\lim_{n\to\infty} x_n) = \lim_{n\to\infty} g(x_n) = \lim_{n\to\infty} x_{n+1} = P$, donde hemos usado la continuidad de g. □

Definición 21 (Método de Newton para encontrar raíces.)**.** Este método para hallar una raíz de g consiste en hallar un punto fijo de $\phi(x) = x - \frac{g(x)}{g'(x)}$, es decir, calcular los sucesivos términos de la sucesión $x_{n+1} = x_n - \frac{g(x_n)}{g'(x_n)}$, partiendo de un $x_0 \in D$.

Teorema 18. *Si $\{x_n\}$ es la sucesión del método anterior, con $f \in \mathcal{C}^1([a, b])$, con $f'(x) \neq 0$ en $[a, b]$ y tal que $\exists \lim_{n\to\infty} x_n = \alpha$, entonces $f(\alpha) = 0$, es decir, converge a una raíz.*

Demostración. $\alpha = \lim_{n\to\infty} x_{n+1} = \lim_{n\to\infty} x_n - \frac{g(x_n)}{g'(x_n)} = \lim_{n\to\infty}(x_n) - \frac{f(\lim_{n\to\infty} x_n)}{f'(\lim_{n\to\infty} x_n)} = \alpha - \frac{f(\alpha)}{f'(\alpha)}$, luego $0 = \frac{f(\alpha)}{f'(\alpha)}$, con lo que $f(\alpha) = 0$. □

El siguiente resultado ilustra cómo debe ser x_0 para que el método converja.

Teorema 19. *Supongamos que se tiene $g : D \to \mathbb{R}$, $x_0 \in D$ tales que $\exists a < b \in \mathbb{R}$ con:*

1. $g \in \mathcal{C}^1([a,b])$
2. $|g'(x)| \leq K \ \forall x \in [a,b]$
3. $g(x) \in [a,b] \ \forall x \in [a,b]$
4. $x_0 \in [a,b]$

Entonces, $\exists P \in [a,b]$ tal que $g(P) = P$, y además:

1. *Si $K < 1$, entonces P es el único con esta propiedad, y la iteración $x_{n+1} = g(x_n)$ converge a P. Se denomina **punto fijo atractor.***
2. *Si $|g'(P)| > 1$, y $x_0 \neq P$, entonces $x_{n+1} = g(x_n)$ no converge a P en ningún caso (por muy cerca que esté). Se denomina **punto fijo repulsivo.***

Demostración. Sabemos que existe P dado que $f(x) := x - g(x)$ tiene signo distinto en $[a,b]$ a causa de la propiedad 3, y es continua por 1. Ahora, si $K < 1$, entonces $|x_{n+1} - P| = |g(x_n) - g(P)| = |g'(\xi)||x_n - P| \leq K|x_n - P|$ por el teorema de Valor Medio. Repitiendo este paso, se obtiene que $|x_{n+1} - P| \leq K^{n+1}|x_0 - P|$. Ahora, $\lim_{n\to\infty} |x_{n+1} - P| = |x_0 - P| \lim_{n\to\infty} K^{n+1} = 0$ al ser $K < 1$, luego $\lim_{n\to\infty} x_{n+1} = P$. Para ver que es único el punto fijo, veamos que si hubiese otro, P', cumpliría lo mismo, y además $\lim_{n\to\infty} |P - P'| = |P - x_n + x_n - P'| \leq |P - x_n| + |P' - x_n| = 0 + 0 = 0$, luego $P = P'$.

Ahora, si $|g'(P)| > 1$, por continuidad $\exists \delta$ tal que $\forall x \in (P-\delta, P+\delta)$, se tiene $|g'(x)| > 1$. Supongamos ahora que $\lim_{n\to\infty} x_n = P$. En ese caso, habría algún x_n con $|x_n - P| < \delta$, y entonces $|x_{n+1} - P| = |g(x_n) - g(P)| = |g'(\xi)||(x_n - P)| \geq |x_n - P|$, dado que $\xi \in [x_n, P] \subset (P-\delta, P+\delta)$, y lo que hemos obtenido es que el término siguiente de la sucesión está mas lejos de P que este, impidiendo que converja. □

Los siguientes dos corolarios del teorema anterior son útiles para ver cómo de rápido converge el método:

Proposición 25. *Sea α el punto atractor de g en las condiciones del teorema anterior. Sea $e_n = x_n - \alpha$ el error en el paso n del método. Entonces, $\lim_{n\to\infty} \frac{e_{n+1}}{e_n} = g'(\alpha)$. Como sabemos, $-1 \leq g'(\alpha) \leq 1$, de modo que el método convergerá más rápido cuanto más pequeño en valor absoluto sea $g'(\alpha)$.*

Demostración. $e_{n+1} = (g(x_n) - g(\alpha)) = g'(\xi_n)(x_n - \alpha) = g'(\xi_n)e_n$, con $x_n \gtrless \xi_n \gtrless \alpha$. Por el teorema de compresión, $\lim_{n\to\infty} \xi_n = \alpha$, luego $\lim_{n\to\infty} \frac{e_{n+1}}{e_n} = \lim_{n\to\infty} g'(\xi_n) = g'(\alpha)$ por continuidad. □

Proposición 26. *Sea α el punto atractor de g en las condiciones del teorema anterior, con $g'(\alpha) = 0$ y $g \in \mathcal{C}^2([a,b])$. Sea $e_n = x_n - \alpha$ el error en el paso n del método. Entonces, $\lim_{n\to\infty} \frac{e_{n+1}}{e_n^2} = \frac{g''(\alpha)}{2}$. Es decir, el método converge cuadráticamente en cada paso.*

Demostración. $e_{n+1} = g(x_n) - g(\alpha) = g(\alpha) + g'(\alpha)(x_n - \alpha) + \frac{g''(\alpha)}{2}(x_n - \alpha)^2 + o((x_n - \alpha)^2) - g(\alpha) = \frac{g''(\alpha)}{2}(x_n - \alpha)^2 + o((x_n - \alpha)^2)$ por hipótesis. Ahora, si dividimos por $(e_n)^2$, se tiene $\frac{e_{n+1}}{(e_n)^2} = \frac{g''(\alpha)}{2} + \frac{o((x_n-\alpha)^2)}{(x_n-\alpha)^2}$, luego en el límite, por definición de o-pequeña, se tiene lo que se quería. □

Cabe observar que el método anterior se puede extender (convergencia cúbica, cuádrica...) siempre que g esté en la clase correspondiente y que las derivadas se anulen.

Veamos que el método de Newton cumple esto último:

Observación 13. El método de Newton para la raíz converge rápido porque se trata de un punto fijo atractor de $g(x) = x - \frac{f(x)}{f'(x)}$, cuya derivada es $g'(x) = 1 - \frac{f'(x)^2 - f(x)f''(x)}{f'(x)^2}$, y si α es el atractor, que sabemos que es raíz de f, entonces $g'(\alpha) = 1 - 1 = 0$.

3.4. Otros métodos

Definición 22 (Método *Regula Falsi*)**.** La idea es combinar secante y bisección:

1. Comenzamos con $a_0, b_0 \in \mathbb{R}$ con $f(a_0)f(b_0) < 0$. Tomamos $c_0 = a_0 - \frac{f(a_0)}{f[a_0,b_0]}$ la raíz de la secante en esos dos puntos.
2. Si $f(c_0)f(a_0) < 0$, definimos $a_1 := a_0$ y $b_1 := c_0$. Si $f(c_0)f(a_0) > 0$, ponemos $a_1 := c_0$ y $b_1 := b_0$. Si $f(c_0) = 0$ hemos acabado.
3. Repetimos el proceso con a_1 y b_1, obteniendo la raíz de la secante y decidiendo los dos puntos siguientes mediante el criterio seguido en bisección.

Definición 23 (Método Whittaker)**.** Consiste en fijar una pendiente arbitraria para las rectas que se toman.

Se comienza con $x_0 \in [a, b]$, y se itera la sucesión $x_{n+1} = x_n - \frac{f(x_n)}{\lambda}$ para $\lambda \neq 0$. Se tiene que converge bien si $sig(\lambda) = sig(f')$, $sig(x_0) = sig(f'')$ y $|\lambda| \geq |f'(x_0)|$.

B. Algunos algoritmos de cuadratura y ecuaciones no lineales.

Observación 14 (Cuadratura de trapecio compuesta). Dada $f \in \mathcal{C}^2([a,b])$, $a < b$ y $tol > 0$ una tolerancia para el error, se puede cuadrar como el trapecio así:

1. En primer lugar hallaremos cuantos subintervalos (n) son necesarios. Recordemos que el error es $E = \frac{(b-a)h^2}{12}|f^(2)(\xi)|$. Como $h = (b-a)/n$, se puede despejar con el maximo de la derivada o incrementar n y ver en cada incremento si E = $\frac{(b-a)h^2}{12}M < tol$. Cuando se logre esto, esa será la cota de error. M es la cota de la derivada, que, si no se conoce, se puede tomar un número elevado de muestras en $[a,b]$, evaluar la derivada allí obteniendo el conjunto D y tomar $M = max(|D|)$.
2. Teniendo n, se define $h = \frac{b-a}{n}$. Se inicializa $I = 0$. Ahora, se hace $I \leq I + \frac{h}{2}f(a+ih) + \frac{h}{2}f(a+(i-1)h)$, con $i \in \{1,\ldots,n\}$. De esta manera sumamos la regla en cada intervalo. La multiplicación por $\frac{h}{2}$ puede hacerse al final.

Observación 15 (Cuadratura de Simpson compuesta). Dada $f \in \mathcal{C}^4([a,b])$, $a < b$ y $tol > 0$ una tolerancia para el error, se puede cuadrar con Simpson así:

1. En primer lugar hallaremos cuantos subintervalos (n) son necesarios. Recordemos que el error es $E = \frac{(b-a)h^4}{2880}|f^(4)(\xi)|$. Como $h = (b-a)/n$, se puede despejar con el maximo de la derivada o incrementar n y ver en cada incremento si E = $\frac{(b-a)h^4}{2880}M < tol$. Cuando se logre esto, esa será la cota de error. M es la cota de la derivada, que, si no se conoce, se puede tomar un número elevado de muestras en $[a,b]$, evaluar la derivada allí obteniendo el conjunto D y tomar $M = max(|D|)$.
2. Teniendo n, se define $h = \frac{b-a}{n}$. Se inicializa $I = 0$. Ahora, se hace $I \leq I + \frac{h}{6}f(a+ih) + \frac{h}{6}f(a+(i-1)h) + \frac{4h}{6}f(a+(i-1)h+\frac{h}{2})$, con $i \in \{1,\ldots,n\}$. De esta manera sumamos la regla en cada intervalo. La multiplicación por $\frac{h}{6}$ puede hacerse al final, sin olvidarse del 4 en el punto medio.

Observación 16 (Método de bisección). Dada f y $a < b$, con $f(a)f(b) < 0$, así como $tol > 0$ una tolerancia para el error:

1. Se asigna $x_n := \frac{a+b}{2}$, $err = \frac{b-a}{2}$ y $n := 1$ el número de iteraciones. Ahora, mientras que $err \geq tol$, si $f(a)f(x_n) > 0$, se hace $a := x_n$. Si $f(a)f(x_n) < 0$, en cambio, $b := x_n$. Finalmente, se recalcula $x_n := \frac{a+b}{2}$, $err := \frac{err}{2}$ y se aumenta n.
2. Finalmente, se tiene que x_n aproxima la raíz con un error acotado por err en n iteraciones.

Observación 17 (Método de Newton). Dada f y x_0 un punto inicial, así como $tol > 0$ una tolerancia para el error:

1. Se comienza con $n := 0$, $err := \infty$, y $x_n = x_0$. Mientras $err \geq tol$ y $n < 1000$ (para evitar bloqueos si no converge), se asigna $x_0 := x_n$, ahora se calcula $x_n := x_0 - \frac{f(x_0)}{f'(x_0)}$, se hace $err := |x_n - x_0|$ y se incrementa n.
2. Finalmente, se tiene que x_n aproxima la raíz con un error acotado por err en n iteraciones.

Observación 18 (Método de iteraciones de punto fijo). Dada f, un punto inicial x_0 y $tol > 0$ una tolerancia para el error:

1. Se comienza con $err := \infty$ y $n := 0$.
2. Mientras $err \geq tol$ y $n \leq 1000$ (para evitar bloqueos si no converge), se hace $x_n := f(x_0)$, se calcula $err = |x_n - x_0|$, se guarda $x_0 := x_n$ y se incrementa n.
3. Finalmente, se tiene que x_n aproxima el punto fijo con un error acotado por err en n iteraciones.

Este método permite aplicar el método de newton si se pasa la f adecuada.

4. Resolución de sistemas lineales

En esta sección se trata la resolución del sistema $Ax = b$ con $A \in \mathcal{M}_d$ invertible, $x, b \in \mathbb{R}^d$.

4.1. Eliminación de Gauss

El método de eliminación consiste en aplicar transformaciones elementales a las filas de A y b para reducir A a una matriz triangular superior, manteniendo las soluciones del sistema. El algoritmo es como sigue:

Observación 19. Algoritmo de eliminación de Gauss:

Para i de 1 a $d-1$:
  Para j de $i+1$ a d:
  $l_{ji} := \frac{a_{ji}}{a_{ii}}$
  Para k de $i+1$ a d: $a_{jk} := a_{jk} - l_{ji}a_{ik}$
  $b_j := b_j - l_{ji}b_i$

Como vemos, para cada fila *pivote*, de arriba abajo, por cada una de las filas que tiene debajo, calculamos el **multiplicador** (factor que hay que aplicar al pivote para eliminar los coeficientes debajo de él), y después aplicamos la operación $fila = fila - multiplicador \cdot filapivote$ a toda esa fila, incluido en b.

A continuación, para resolver el sistema, basta con aplicar un algoritmo de resolución regresiva, es decir, comenzar por la fila inferior e ir obteniendo soluciones hacia arriba. Este algoritmo se detallará en profundidad en la siguiente subsección.

En cada paso de i, la nueva matriz $A^{(i+1)}$ se obtiene aplicando una matriz M_i, de forma:

$$M_i = \begin{pmatrix} 1 & 0 & \dots & 0 & \dots & 0 \\ 0 & 1 & \dots & 0 & \dots & 0 \\ \vdots & \vdots & \ddots & \vdots & \dots & \vdots \\ 0 & 0 & \dots & 1 & \dots & 0 \\ 0 & 0 & \dots & -l_{i+1,i} & \dots & 0 \\ 0 & 0 & \dots & -l_{i+2,i} & \dots & 0 \\ \vdots & \vdots & \vdots & \vdots & \dots & \vdots \\ 0 & 0 & \dots & -l_{d,i} & \dots & 1 \end{pmatrix}$$

Con cada $l_{ji} = \frac{a^{(i)}_{ji}}{a^{(i)}_{ii}}$ el multiplicador del paso correspondiente. No es difícil ver que:

$$M_i^{-1} = \begin{pmatrix} 1 & 0 & \dots & 0 & \dots & 0 \\ 0 & 1 & \dots & 0 & \dots & 0 \\ \vdots & \vdots & \ddots & \vdots & \dots & \vdots \\ 0 & 0 & \dots & 1 & \dots & 0 \\ 0 & 0 & \dots & l_{i+1,i} & \dots & 0 \\ 0 & 0 & \dots & l_{i+2,i} & \dots & 0 \\ \vdots & \vdots & \vdots & \vdots & \dots & \vdots \\ 0 & 0 & \dots & l_{d,i} & \dots & 1 \end{pmatrix}$$

En efecto, tenemos que $M_i = I - l_i e_i^t$ donde e_i es el vector canónico i-ésimo, y l_i es el vector con los multiplicadores del paso i. Entonces, queremos ver que $M^{-1} = I + l_i e_i^t$. Si multiplicamos: $(I - l_i e_i^t)(I + l_i e_i^t) = I - (l_i e_i^t)^2 = I$, dado que $(l_i e_i^t)^2 = 0$, en virtud del resultado siguiente:

Proposición 27. *Se tiene que* $(l_k e_k^t)(l_m e_m^t) = 0$ *siempre que* $k \leq m$.

Demostración. Si $A = (a_i j)_{i,j}$ es ese producto, está claro que $a_{ij} = \sum_{s=1}^{d} (l_k e_k^t)_{i,s} (l_m e_m^t)_{s,j} = \sum_{s=1}^{d} (l_k)_i (e_k)_s (l_m)_s (e_m)_j = (l_k)_i (l_m)_k (e_m)_j = 0$ dado que $(l_m)_k = 0$ al ser $k \leq m$. □

Esto da lugar al siguiente método:

4.2. Factorización *LU*

Como hemos visto, tenemos un método, a través de eliminación Gaussiana, de obtener una expresión del tipo:

$$M_{d-1} M_{d-2} \dots M_2 M_1 A = U$$

Con U una triangular superior y las M_k fácilmente invertibles y triangulares inferiores. Por tanto:

$$A = (M_{d-1} M_{d-2} \dots M_2 M_1)^{-1} U = M_1^{-1} M_2^{-1} \dots M_{d-1}^{-1} U$$

Si denotamos $L = M_1^{-1} M_2^{-1} \dots M_{d-1}^{-1}$, podemos ver que:

$$L = \begin{pmatrix} 1 & 0 & \dots & 0 & \dots & 0 \\ l_{2,1} & 1 & \dots & 0 & \dots & 0 \\ \vdots & \vdots & \ddots & \vdots & \dots & \vdots \\ l_{i,1} & l_{i,2} & \dots & 1 & \dots & 0 \\ l_{i+1,i} & l_{i+1,2} & \dots & l_{i+1,i} & \dots & 0 \\ l_{i+2,i} & l_{i+2,2} & \dots & l_{i+2,i} & \dots & 0 \\ \vdots & \vdots & \vdots & \vdots & \dots & \vdots \\ l_{d,i} & l_{d,2} & \dots & l_{d,i} & \dots & 1 \end{pmatrix}$$

Es decir, la matriz L es triangular inferior. Se tiene entonces una factorización $A = LU$ fácil de obtener, y que nos ayudará a resolver el sistema $Ax = b$ de manera inmediata:

Observación 20 (Sustitución progresiva). Dada una matriz L triangular inferior, se puede resolver de inmediato el sistema $Lx = b$ como sigue:

Para i de 1 a d:
$x_i := \frac{b_i - \sum_{k=1}^{i-1} a_{ik} x_k}{a_{ii}}$

Observación 21 (Sustitución regresiva). Dada una matriz U triangular superior, se puede resolver de inmediato el sistema $Ux = b$ como sigue:

Para i de d a 1:
$x_i := \frac{b_i - \sum_{k=i+1}^{d} a_{ik} x_k}{a_{ii}}$

Observación 22 (Resolución de un sistema en el que $A = LU$). Para resolver un sistema $LUx = b$:

1. Resolver el sistema $Lc = b$ mediante sustitución progresiva.
2. Resolver el sistema $Ux = c$ mediante sustitución regresiva.
3. Se tiene entonces que $Ax = LUx = Lc = b$, como se quería.

Proposición 28. *Dada* $A \in \mathcal{M}_d(\mathbb{R})$ *con todos sus menores angulares invertibles, entonces el método descrito anteriormente (tanto factorización LU como resolución) funciona en todos los pasos.*

Demostración. En primer lugar, $a_{11} \neq 0$ por hipótesis, luego se puede efectuar el primer paso de la descomposición. Para el segundo paso, consideramos la submatriz $A_2 = \begin{pmatrix} a_{11} & a_{12} \\ 0 & \tilde{a}_{22} \end{pmatrix}$ que queda tras el primer paso, en la esquina superior izquierda. Como es equivalente al segundo menor angular de A por una transformación M_1 biyectiva, entonces A_2 es invertible también, y por tanto $\tilde{a}_{22} \neq 0$ y podemos hacer el siguiente paso. Si continuamos con el mismo argumento tenemos el mismo resultado en todos los pasos. □

4.2.1. Método del pivotaje parcial

Es posible que en ocasiones no podamos efectuar el método de Gauss debido a que el *pivote* es nulo. No obstante, si A es invertible, habrá alguna fila debajo del pivote cuyo elemento no sea nulo, y bastará con intercambiar su fila con la del pivote:

Definición 24. En el **método de pivotaje parcial**, tras cada paso del método de Gauss se efectúa una permutación entre la fila pivote que corresponde y aquella fila con el elemento pivote de mayor módulo. El método queda:

Para i de 1 a $d-1$:

Intercambiar en A la fila i por la fila $k \geq i$ tal que $|a_{kk}| = \max_{j \geq i} |a_{jj}|$

Para j de $i+1$ a d:

$l_{ji} := \frac{a_{ji}}{a_{ii}}$

Para k de $i+1$ a d: $a_{jk} := a_{jk} - l_{ji} a_{ik}$

$b_j := b_j - l_{ji} b_i$

Al igual que cada iteración del bucle principal correspondía por multiplicar por M_i una matriz adecuada con los multiplicadores, se tiene que ahora cada iteración es multiplicar a $A^{(i)}$ por la izquierda por $M_i P_i$, donde P_i es una matriz elemental permutadora de las filas i y k.

Observación 23. Tras el método de pivotaje parcial, se tiene una descomposición $PA = LU$ donde P es una matriz elemental permutadora de filas, L es triangular inferior con 1s en la diagonal y U es triangular superior.

Razón. Tras efectuar el método, tenemos $M_{d-1} P_{d-1} M_{d-2} P_{d-2} \dots M_2 P_2 M_1 P_1 A = U$. Podemos reagrupar las matrices de la izquierda teniendo en cuenta que las permutaciones son autoinversas:

$U = M_{d-1}(P_{d-1} M_{d-2} P_{d-1}) P_{d-1} P_{d-2} \dots M_2 P_2 M_1 P_1 A$, y así sucesivamente iríamos recogiendo a la izquierda del todo matrices del tipo $(\tilde{P} M_i \tilde{P})$ con $\tilde{P}$ permutadores. Estas matrices tendrán la misma forma que las M_i, de modo que podemos invertirlas como sabemos para obtener la expresión deseada.

4.3. Método de mínimos cuadrados

A continuación nos planteamos como podríamos resolver un sistema de más ecuaciones que incógnitas de manera satisfactoria. Dicho de otro modo, si $A \in \mathcal{M}_{m \times n}$, con $m > n$, y planteamos el sistema $Ax = b$ con $(A|b)$ de rango $n+1$, buscamos qué *solución* x conviene.

Definición 25 (Método de mínimos cuadrados)**.** Dado $x \in \mathbb{R}^n$, se define su **residuo** respecto a un sistema $Ax = b$ como el descrito anteriormente, por $r(x) = Ax - b$. El método de mínimos cuadrados busca **minimizar la norma euclídea de** r.

Es decir, buscamos $\tilde{x} \in \mathbb{R}^n$ tal que $\|\tilde{x}\| = \min_{x \in \mathbb{R}_0} \{\|r(x)\|\}$.

Esta norma al ser una forma cuadrática podrá minimizarse. Por ejemplo, en caso del sistema de dos ecuaciones y una incógnita $a_1x = b_1$ y $a_2x = b_2$, resulta que, derivando, $\tilde{x} = \frac{b_1a_1+b_2a_2}{a_1^2a_2^2}$.

Si bien este método puede encontrar soluciones derivando, existe una forma definitivamente útil para obtener la $\tilde{x}$ buscada:

Proposición 29. *Dado el sistema $Ax = b$ con las características explicadas anteriormente, la solución al problema de mínimos cuadrados es el x_0 tal que $A^tAx_0 = A^tb$.*

Demostración. x_0 verifica que $0 = A^t(Ax_0 - b)$, lo que quiere decir que $Ax_0 - b$ es ortogonal a cada columna de A, y por tanto a cada combinación de estas (ImA). Sabiendo esto, se tiene que $Ax - b = Ax_0 - b + A(x - x_0)$, luego $\|Ax - b\|^2 = \|Ax_0 - b + A(x - x_0)\|^2 = (Ax_0 - b, Ax_0 - b) + 2(Ax_0 - b, A(x - x_0)) + (A(x-x_0), A(x-x_0)) = (Ax_0-b, Ax_0-b)+(A(x-x_0), A(x-x_0)) \geq (Ax_0-b, Ax_0-b) = \|Ax_0 - b\|^2$. Hemos utilizado que $(Ax_0 - b, A(x - x_0)) = 0$, por la ortogonalidad discutida anteriormente. □

Obsérvese que se trata de un sistema con la matriz de coeficientes de $n \times n$ y la columna de independientes de $n \times 1$, es decir, con mismo número de incógnitas que ecuaciones. Este, por tanto, podría ser resuelto con LU u otros métodos que veremos posteriormente.

4.4. Descomposición QR

Hemos visto con LU una manera de descomponer una matriz en otras dos más sencillas que dan lugar a dos sistemas de inmediata resolución. El método QR sigue la misma estrategia, y además tendrá utilidad para el método de mínimos cuadrados, ya que puede hacerse en matrices $m \times n$, $m \geq n$.

En lugar de descomponer A en una triangular superior y otra inferior, vamos a poner $A = QR$ con Q de orden $m \times n$, R de orden $n \times n$, tales que $Q^tQ = I_n$ (es una matriz cuyas columnas son **vectores ortonormales**), y R es triangular superior.

Antes de entrar en detalles sobre la obtención de dichas matrices, veamos algunas de sus utilidades:

Observación 24 (Resolución de sistemas QR). En caso de que A sea cuadrada, entonces resolver $Ax = QRx = b$ es inmediato como sigue:

1. Se halla el c tal que $Qc = b$. Como, si Q es cuadrada, será ortonormal, es inmediato que $c = Q^tb$.
2. Se resuelve el sistema $Rx = c$ por sustitución regresiva.

Observación 25 (Resolución de mínimos cuadrados). Supongamos ahora que $m > n$ y estamos queriendo obtener la solución de mínimos cuadrados de $Ax = b$. La factorización QR también aporta utilidad aquí:

1. Debemos resolver el sistema $A^tAx = A^tb$, es decir $R^tQ^tQRx = R^tQ^tb$, y por ortogonalidad, $R^tRx = R^tQ^tb$. Obsérvese que R^t ya es triangular inferior, y R superior, luego nos viene dada de inmediato la LU.
2. Se halla c tal que $R^tc = R^tQ^tb$ mediante sustitución progresiva.
3. Se halla x tal que $Rx = c$ mediante sustitución regresiva.

4.4.1. Método de Gram-Schmidt

A continuación veremos un primer método para obtener la descomposición:

Definición 26 (Gram-Schmidt para QR)**.** Queremos obtener QR con las columnas de Q ortonormales y R triangular superior. Para obtener vectores ortonormales dados otros vectores (las columnas de A), existe un método conocido como de **Gram-Schmidt**.

1. Se comienza con $R = (0)_{i,j} \in \mathcal{M}_n$, y $Q = A$.

   Para cada i de 1 a n:

   Para cada j de 1 a i:

   Se toma $r_{ji} := (a_i, q_j)$ Aquí a_i y q_i denotan la columna i de A y Q, respectivamente.

   Ahora se hace $q_i := a_i - \sum_{k=1}^{i-1} r_{ki} q_k$.

   Finalmente se hace $q_i := \frac{q_i}{\|q_i\|}$

Es importante que el orden sea el establecido arriba, dado que la matriz Q va actualizándose en mitad del algoritmo e influye en los valores que se calculan para R.

Ir quitando a cada vector de A las componentes en los espacios de los demás vectores de A, e irlos normalizando, garantiza que las columnas de Q son ortonormales. Además, si se comprueba con cuidado, tal y como se ha construido R, se obtiene triangular superior y con $A = QR$, como se quería.

4.4.2. Método de Householder

Otro método para obtener matrices QR utiliza los conocidos como **reflectores de Householder**, que son simetrías respecto de un plano (de dimensión $n-1$), que permiten transformar la matriz A en una triangular superior. Como las simetrías son ortogonales, tendremos la descomposición QR. Además, como las simetrías mantienen la norma de sus vectores, la Q también será isometría (cosa no garantizada en Gram-Schmidt) y permitirá realizar métodos iterativos en un futuro (para calcular autovalores) sin que los vectores se vean muy reducidos o muy aumentados.

Definición 27 (Reflector de Householder)**.** Un **reflector de Householder** P es una simetría cuya única condición es que lleva el vector $x \in \mathbb{R}^d$ en el vector $y \in \mathbb{R}^d$. Para ello, debe tenerse que $\|x\| = \|y\|$, o de otro modo no existe tal simetría.

Para construirlo, simplemente basta con observar que una simetría respecto del ortogonal al vector $x - y$ (respecto de un plano $(d-1)$-dimensional) satisface lo que se quiere. Es decir:

$$P = I - 2\frac{(x-y)(x-y)^t}{\|x-y\|^2}$$

Estamos restando a la identidad, dos veces la proyección sobre $\langle x - y \rangle$, lo que da la simetría. Se puede comprobar de manera rutinaria que el resultado es una matriz simétrica, con $P^2 = I_d$, y tal que $Px = y$.

Definición 28 (Factorización QR de Householder)**.** El método irá transformando cada columna a_i de A hasta obtener una triangular superior.

1. Se construye el primer reflector, P_1 de tal modo que $P_1 a_1 = r_1$, donde:

$$r_1 = \begin{pmatrix} -signo(a_{1,1})\,\|a_1\| \\ 0 \\ \vdots \\ 0 \end{pmatrix}$$

   Es decir, estamos llevando la primera columna de A a un vector que tiene ceros salvo en la primera componente. Esta componente, para que exista simetría, es $\|a_1\|$. El signo se añade para conveniencia de métodos iterativos, con el fin de que no diverjan.

2. Ahora, se tiene que:

$$P_1 A = \begin{pmatrix} -signo(a_{1,1})\,\|a_1\| & a_{12} & \dots & a_{2d} \\ 0 & a_{22} & \dots & a_{2d} \\ \vdots & \vdots & \ddots & \vdots \\ 0 & a_{d1} & & \\ 0 & a_{d2} & \dots & a_{dd} \end{pmatrix}$$

Nos centramos en la submatriz:

$$A^{(1)} = \begin{pmatrix} a_{22} & \dots & a_{2d} \\ \vdots & \ddots & \vdots \\ a_{d1} & & \\ a_{d2} & \dots & a_{dd} \end{pmatrix}$$

Podemos considerar de nuevo un reflector de Householder, $\tilde{P}_2 \in \mathcal{M}_{d-1}$ en esta matriz. Lo obtenemos como antes, que lleve $a_1^{(1)}$ a una columna:

$$r_2 = \begin{pmatrix} -signo(a_{1,1}^{(1)}) \left\| a_1^{(1)} \right\| \\ 0 \\ \vdots \\ 0 \end{pmatrix}$$

3. El reflector obtenido de esta manera, para que sea de tamaño $d \times d$ y podamos aplicarlo a la matriz entera, pero que no nos cambie nada salvo la submatriz, es:

$$P_2 = \left(\begin{array}{c|c} Id_1 & 0 \\ \hline 0 & \tilde{P}_2 \end{array} \right)$$

4. Repetimos el método en cada submatriz que se va generando al ir aplicando las simetrías para obtener los $d-1$ reflectores. En cada paso, teniendo el reflector de la submatriz, el reflector para A se obtiene así:

$$P_k = \left(\begin{array}{c|c} Id_{k-1} & 0 \\ \hline 0 & \tilde{P}_k \end{array} \right)$$

5. Se tiene, por como se ha construido todo, que $P_{d-1}P_{d-2}\dots P_1 A = R$ con R triangular superior. Basta con tomar $Q = (P_{d-1}P_{d-2}\dots P_1)^t$.

Cabe destacar que este método, si la matriz original es $m \times n$, con $m \geq n$, se realiza con un reflector por columna ($n-1$ reflectores) y da lugar a una descomposición en la que Q es $m \times m$ y R es $m \times n$, en lugar de los tamaños de las Q, R que hemos mencionado anteriormente.

4.5. Métodos iterativos

En esta sección vamos a proponer métodos iterativos que obtendrán aproximaciones a la solución de un sistema $Ax = b$ con $A \in \mathcal{M}_{n\times n}(\mathbb{R})$. Estudiaremos cuándo convergen a la solución del sistema.

La idea general es separar A en dos matrices, N_1 y N_2, de tal modo que $A = N_1 - N_2$ y por tanto $x = N_1^{-1}N_2x + N_1^{-1}b$. De este modo, hemos convertido el sistema en otro equivalente:

$$x = Mx + c \qquad M \in \mathcal{M}_n, \quad c \in \mathbb{R}^n$$

Y podremos resolverlo iterando un valor inicial x_0 a través de $g(x) = Mx + c$ como hacíamos en métodos de punto fijo para resolver ecuaciones no lineales.

Definición 29 (Método de Jacobi)**.** Este método se aplica a matrices A tal que $D = diag(A)$ (matriz que contiene la diagonal de A únicamente) es invertible. Es decir, matrices con $a_{ii} \neq 0$, $\forall i \in \{1, \dots, d\}$.

En este caso, se tiene $A = D+(A-D)$, con lo que $Ax = b$ se reescribe como $x = -D^{-1}(A-D)x+D^{-1}b$, y el método consiste en iterar la sucesión:

$$\begin{cases} x_0 \in \mathbb{R}^d & \text{a priori arbitrario, y} \\ x_{k+1} = -D^{-1}(A - D)x_k + D^{-1}b & \forall k \geq 1 \end{cases}$$

Observemos que en este método, si $x_k^{(n)}$ denota la k-ésima componente de la n-ésima iteración, y $A = (a_{ij})_{i,j}$, podemos expresar la siguiente relación:

$$x_k^{(n+1)} = c_k - \sum_{j\neq k} \frac{a_{kj}}{a_{kk}} x_j^{(n)}$$

Donde c es el vector que sumamos ($D^{-1}b$). Lo que se observa es que, si vamos calculando en orden las componentes, podríamos utilizar, para calcular $x_k^{(n+1)}$, las componentes $x_j^{(n+1)}$ con $j < k$ que ya hemos calculado, en lugar de utilizar las $x_j^{(n)}$, dado que conforme iteramos más, nos acercamos más a la solución, y por tanto debemos esperar que usando las componentes nuevas el método avance más rápido.

Analíticamente, esto se puede escribir:

$$x_k^{(n+1)} = c_k - \sum_{j=1}^{k-1} \frac{a_{kj}}{a_{kk}} x_j^{(n+1)} - \sum_{j=k+1}^{n} \frac{a_{kj}}{a_{kk}} x_j^{(n+1)}$$

Y matricialmente, da lugar al método de Gauss-Seidel:

Definición 30 (Método de Gauss-Seidel)**.** En este método, sea L_* la matriz triangular inferior con las entradas de A (diagonal incluida), y sea $U = A - L_*$ (Triángulo superior de A sin la diagonal). Descomponemos $A = L_* + U$.

En este caso, $Ax = b$ se reescribe como $x = -L_*^{-1}Ux+L_*^{-1}b$, y el método consiste en iterar la sucesión:

$$\begin{cases} x_0 \in \mathbb{R}^d & \text{a priori arbitrario, y} \\ x_{k+1} = -L_*^{-1}Ux_k + L_*^{-1}b & \forall k \geq 1 \end{cases}$$

Para estudiar cuándo converge un método iterativo, y con que velocidad, definimos:

Definición 31. Dada $M \in \mathcal{M}_n$, el **espectro** de M es $\sigma(M) = \{\lambda \in \mathbb{R} : \exists v \in \mathbb{R}^n, v \neq 0 : Mv = \lambda v\}$, es decir, el conjunto de autovalores de M.

El **radio espectral** de M es $\rho(M) = \max\limits_{\lambda\in\sigma(M)} |\lambda|$.

Teorema 20. *El método iterativo dado por $x_{k+1} = Mx_k + c$, para el sistema de ecuaciones $x = Mx + c$, converge para todo x_0 inicial* ***si y solo si*** *$\rho(M) < 1$.*

Demostración. Sea $e_k = x_k - x$ el error que comete el iterante en el paso k. Como $x_{k+1} = Mx_k + c$, y $x = Mx + c$, restando se obtiene que $e_{k+1} = Me_k$. Por lo tanto, de manera recursiva, $e_{k+1} = M^{k+1}e_0$ y entonces se tiene convergencia si y solo si $\lim_{k\to\infty} M^k = 0$. Sea J la forma de Jordan de M, y P la matriz de paso tal que $M = P^{-1}JP$. Entonces, $\lim_{k\to\infty} M^k = \lim_{k\to\infty} P^{-1}J^kP = 0$ si y solo si M tiene los autovalores de módulo ≤ 1, dado que cada bloque de Jordan converge a 0 en ese caso. (Si es mayor que 1 diverge, y si es justo 1 no converge a 0 sino que permanece constante). □

Proposición 30. *Si A es* ***estrictamente diagonalmente dominante por filas****, es decir, si $|a_{ii}| > \sum_{j\neq i} |a_{ij}|$ $\forall i \in \{1, \ldots, n\}$, convergen Jacobi y Gauss-Seidel, y $1 > \rho(M_j) > \rho(M_{GS})$, es decir, en este caso, Gauss-Seidel lo hace más rápido.*

Definición 32 (Razón de convergencia)**.** Se define la **razón de convergencia** de un método iterativo $x_{k+1} = Mx_k + c$ como $R = -\log_{10}\rho(M)$. Esta indica lo rápido que converge un método. Para que $e_n = \frac{e_0}{10^m}$, es decir, para reducir el error inicial 10^m veces, hay que hacer $n = \frac{m}{R}$ iteraciones.

5. Cálculo numérico de autovalores y autovectores

En esta sección veremos métodos para obtener autovalores y autovectores de forma numérica.

Definición 33 (Método de la potencia)**.** Sea $A \in \mathcal{M}_d$, simétrica, por tanto con d autovalores $\{\lambda_i\}_1^d$ con autovectores $\{u_i\}_1^d$, que forman base ortonormal, tales que:

$$|\lambda_1| > |\lambda_2| \geq |\lambda_3| \geq \cdots \geq |\lambda_d|$$

A λ_1 se le denomina **autovalor dominante**, y para este método, como se indica, debe ser simple. El método permite aproximar λ_1 y u_1, y es el siguiente (en adelante $x_{i,j}$ denota la j-ésima componente del vector x_i):

1. Se toma $x_0 \in \mathbb{R}^d$ cualquiera. Sea $\alpha_0 = x_{0,k}$ tal que $|x_{0,k}| = \max_{1\leq j\leq d} |x_{0,j}|$. En la norma $\|\|_\infty$, se tiene que $\|x\|_\infty = \max_{1\leq j\leq d} |x_{0,j}|$, luego lo que estamos tomando es la coordenada que da lugar a esa norma (aunque con su signo). Ahora ponemos $z_0 = \frac{x_0}{\alpha_0}$ (lo normalizamos en $\|\|_\infty$, con la coordenada dominante valiendo 1).
2. Tomamos $x_1 = Az_0$, y hacemos $\alpha_1 = x_{1,k}$ con $|x_{1,k}| = \max_{1\leq j\leq d} |x_{1,j}|$. Definimos $z_1 = \frac{x_1}{\alpha_1}$.
3. En general, si tenemos z_k, entonces $x_{k+1} = Az_k$, $\alpha_{k+1} = |x_{k+1,t}$ con $|x_{k+1,t}| = \max_{1\leq j\leq d} |x_{k+1,j}|$, y $z_{k+1} = \frac{x_{k+1}}{\alpha_{k+1}}$.
4. Se tiene que $\lim_{k\to\infty} z_k = z$, con z paralelo a u_1, y $\lim_{k\to\infty} \alpha_k = \lambda_1$, con probabilidad 1 (aunque no siempre).

Demostración de la convergencia. Dada la base ortonormal de autovectores de A, los normalizamos en $\|\|_\infty$, con el mismo procedimiento que el algoritmo (para que la coordenada dominante sea 1 y no -1), dando lugar a una base ortogonal (respecto de la norma euclídea) de autovectores $\{v_1, v_2, \ldots, v_d\}$. Entonces, si $z_0 = \sum_1^d a_j v_j$, se tiene $x_1 = Az_0 = \sum_1^d a_j Av_j = \sum_1^d a_j \lambda_j v_j$. En ese caso, si $\alpha_1 \neq 0$, lo cual ocurrirá si y solo si z_0 no es autovector de 0 (ocurre con probabilidad 1), podemos poner $z_1 = \frac{\lambda_1}{\alpha_1}(a_1 v_1 + \sum_2^d \frac{\lambda_i a_i}{\lambda_1} v_i)$.

En ese caso, $Az_1 = \frac{\lambda_1}{\alpha_1}(a_1\lambda_1 v_1 + \sum_2^d \frac{\lambda_i^2 a_i}{\lambda_1} v_i)$, al aplicar A otra vez, y normalizando de nuevo: $z_2 = \frac{\lambda_1^2}{\alpha_1\alpha_2}(a_1 v_1 + \sum_2^d \frac{\lambda_i^2 a_i}{\lambda_1^2} v_i)$.

En general, podremos escribir $z_k = \frac{\lambda_1^k}{\prod_1^k \alpha_i}(a_1 v_1 + \sum_2^d \frac{\lambda_i^k a_i}{\lambda_1^k} v_i)$. Como λ_1 es el de mayor módulo, entonces $\lim_{k\to\infty} \frac{\lambda_j^k}{\lambda_1^k} = 0$ si $j > 1$, y por ello $\lim_{k\to\infty} z_k = \lim_{k\to\infty} \frac{\lambda_1^k a_1}{\prod_1^k \alpha_i} v_1$.

A continuación debemos observar que z_k siempre está normalizado en $\|\|_\infty$, con coordenada dominante 1, luego debe converger a un vector normalizado en $\|\|_\infty$ con coordenada dominante 1. Como v_1 ya tiene tal propiedad, no queda otra que $\lim_{k\to\infty} \frac{\lambda_1^k a_1}{\prod_1^k \alpha_i} = 1$.

Finalmente, si se da este caso, tenemos que $\lim_{k\to\infty} z_k = v_1$. Ahora sea $\eta_k = \frac{\lambda_1^k a_1}{\prod_1^k \alpha_i}$. Una sencilla manipulación permite ver que $\eta_k = \eta_{k-1}\frac{\lambda_1}{\alpha_k}$. Tomando límites, queda que $1 = \lim_{k\to\infty} \frac{\lambda_1}{\alpha_k}$, con lo que $\lim_{k\to\infty} \alpha_k = 1$. □

Este método puede modificarse ligeramente para encontrar otro autovalor que no sea el dominante. Para ello, construiremos una matriz alternativa en la que ese autovalor sí sea dominante:

Definición 34 (Método de la potencia inversa)**.** Si λ_j es un autovalor que queremos estimar, simple y no dominante, tomamos $\lambda \in \mathbb{R}$ tal que $|\lambda - \lambda_j| < \min_{i\neq j} |\lambda - \lambda_i|$, es decir, más próximo a dicho autovalor que a cualquier otro. A continuación construimos la matriz $B = (A - \lambda I)^{-1}$.

Si aplicamos el método de la potencia a esta matriz B, se obtiene como autovalor el número $\frac{1}{\lambda_j - \lambda}$, y como autovector se obtendrá un $z \in \mathbb{R}^d$ que es autovector de λ_j en A. (Y de $\frac{1}{\lambda_j-\lambda}$ en B).

Demostración. En primer lugar, veamos que si y solo si β es autovalor de B, entonces $(A-\lambda I)^{-1}v = \beta v$. Por tanto, $v = (A - \lambda I)\beta v$, de tal modo que $Av = (\frac{1}{\beta} + \lambda)v$. Es decir, $(\frac{1}{\beta} + \lambda) = \lambda_i$ para algún autovalor de A, λ_i. Entonces $\beta = \frac{1}{\lambda_i - \lambda}$, y con el mismo autovector que λ_i en A. Como λ_j es simple y λ se escogió de esa manera, entonces el autovalor $\frac{1}{\lambda_j-\lambda}$ de B es dominante y el método de la potencia nos lo proporciona. Como tiene el mismo autovector que λ_j en A, se tiene lo que se quería. □

Definición 35 (Método QR para autovalores y autovectores)**.** El siguiente método iterativo permite estimar los autovalores de $A \in \mathcal{M}_d$ con su factorización QR. Para evitar que el tamaño de los elementos de las iteraciones se dispare o se reduzca mucho, conviene que la QR obtenida sea mediante el método de Householder, que mantiene las normas de las columnas en todo momento.

1. Se toma $A := A_0$ y se factoriza de tal modo que $A_0 = Q_0 R_0$.
2. Se define $A_1 = R_0 Q_0$. Cabe observar que como $R_0 = Q_0^t A_0$, entonces $A_1 = Q_0^t A_0 Q_0$, es decir, es A_0 cambiada de base, y por ello **tiene los mismos autovalores**.
3. Iterativamente, si se tiene A_k, entonces se factoriza $A_k = Q_k R_k$, y se toma $A_{k+1} = R_k Q_k$. En todo momento A_k tiene los mismos autovalores que la original.
4. Si converge, $\lim_{k\to\infty} A_k = D$, con D diagonal. Por lo tanto, la diagonal de D estará poblada por los autovalores de A. Además, el cambio de base que lleva A en D, según hemos visto antes, está dado por la matriz $\lim_{k\to\infty} Q_0 Q_1 \dots Q_k$, luego las columnas de esta matriz son los autovectores.

C. Algunos algoritmos de sistemas lineales y cálculo de autovectores.

Observación 26 (Descomposición LU). Dada $A \in \mathcal{M}_d$, con todos los menores angulares invertibles, se pueden obtener matrices L, U tales que $A = LU$, L triangular inferior con 1s en la diagonal y U triangular superior, así:

1. Se comienza asignando $L := Id_d$ y $U := A$. En L iremos almacenando los multiplicadores, y en U iremos aplicándolos.
2. Para cada i de 1 a $d-1$ (cada fila-pivote):

   Para cada j de $i+1$ a d (cada fila por debajo del pivote):

   Se calcula el multiplicador: $L_{ji} := \frac{U_{ji}}{U_{ii}}$.

   Para cada k de i a d (cada columna):

   Se hace $U_{jk} := U_{jk} - L_{ji}U_{ik}$

El algoritmo finaliza con las U y L especificadas.

Observación 27 (Resolución de sistema lineal sin invertir, mediante LU). Dada $A \in \mathcal{M}_d$, con todos los menores angulares invertibles, y $b \in \mathbb{R}^d$, podemos encontrar $x \in \mathbb{R}^d$ tal que $Ax = b$ así:

1. Se obtienen L y U aplicando el algoritmo anterior sobre A.
2. Se inicializan los vectores $c, x \in \mathbb{R}^d$ a los valores que se desee (por ejemplo, todo ceros, o un rango del 1 al d).
3. Se va a resolver $Lc = b$. Para ello, aplicamos sustitución progresiva como sigue:
4. Para cada i de 1 a d (cada ecuación):

   Se toma $c_i := b_i$.

   Para cada j de 1 a $d-1$ (cada incógnita ya resuelta):

   Se hace $c_i := c_i - L_{ij}c_j$

   Se divide $c_i := \frac{c_i}{L_{ii}}$
5. Se va a resolver $Ux = c$. Para ello, aplicamos sustitución regresiva como sigue:
6. Para cada i de d a 1 (cada ecuación, empezando por abajo):

   Se toma $x_i := c_i$.

   Para cada j de $i+1$ a d (cada incógnita ya resuelta):

   Se hace $x_i := x_i - U_{ij}x_j$

   Se divide $x_i := \frac{x_i}{U_{ii}}$

El algoritmo finaliza con el x especificado.

Observación 28 (Resolución de sistema lineal por método de Jacobi). Dada $A \in \mathcal{M}_d$ invertible, con toda la diagonal no nula, $b \in \mathbb{R}^d$, y una tolerancia t para el error podemos encontrar $x_n \in \mathbb{R}^d$ tal que $Ax_n = b$ con un error ϵ menor que t, por el método de Jacobi, en n pasos, así:

1. Se calcula P la diagonal de A. Se calcula P^{-1} invirtiendo cada escalar no nulo de P.

2. Se obtiene la matriz de iteración $M := P^{-1}(P - A)$. Se obtiene el vector de iteración $c := P^{-1}b$.

3. Se obtiene $\rho(M)$ como el máximo de los valores absolutos de los autovalores de M. Se ajusta $\epsilon := \rho(M)$. Se ajusta $x_n := 0 \in \mathbb{R}^d$ y $n := 0$.

4. Mientras que $\epsilon \geq t$:

   Se itera: $x_n := Mx_n + c$

   Se hace $\epsilon := \epsilon \cdot \rho(M)$

   Se incrementa $n := n + 1$.

Cabe observar que el método solo funciona si $\rho(M) < 1$, por lo que puede convenir utilizar una condición de parada alternativa en el bucle principal.

Observación 29 (Resolución de sistema lineal por método de Gauss-Seidel). Dada $A \in \mathcal{M}_d$ invertible, con toda la diagonal no nula, $b \in \mathbb{R}^d$, y una tolerancia t para el error podemos encontrar $x_n \in \mathbb{R}^d$ tal que $Ax_n = b$ con un error ϵ menor que t, por el método de Gauss-seidel, en n pasos, así:

1. Se calcula P el triángulo inferior (diagonal inclusive) de A.

2. Se obtiene la matriz de iteración $M := P^{-1}(P - A)$. Se obtiene el vector de iteración $c := P^{-1}b$.

3. Se obtiene $\rho(M)$ como el máximo de los valores absolutos de los autovalores de M. Se ajusta $\epsilon := \rho(M)$. Se ajusta $x_n := 0 \in \mathbb{R}^d$ y $n := 0$.

4. Mientras que $\epsilon \geq t$:

   Se itera: $x_n := Mx_n + c$

   Se hace $\epsilon := \epsilon \cdot \rho(M)$

   Se incrementa $n := n + 1$.

Cabe observar que el método solo funciona si $\rho(M) < 1$, por lo que puede convenir utilizar una condición de parada alternativa en el bucle principal.

Observación 30 (Descomposición QR mediante Gram-Schmidt). Dada $A \in \mathcal{M}_d$, invertible, se pueden obtener matrices Q, R tales que $A = QR$, Q ortonormal y R triangular superior, así:

1. Se comienza asignando $Q := A$, y $R := 0 \in \mathcal{M}_d$.

2. Para i de 1 a d (por cada columna):

   Para cada j de 1 a $i - 1$ (por cada columna ya calculada):

   Se asigna $R_{ji} := (a_i, q_j)$, donde a_i es la columna i de A, y q_j la columna j de Q.

   Se resta $q_i := q_i - R_{ji}q_j$

   Ahora se calcula $R_{ii} := \|q_i\|$.

   Finalmente se divide: $q_i := \frac{q_i}{R_{ii}}$

El algoritmo finaliza con las Q y R especificadas.

Observación 31 (Descomposición QR mediante Householder). En las mismas condiciones que el algoritmo anterior, se puede descomponer en Q, R por Householder así (Denotamos $A^{(a\to b, c\to d)}$ la submatriz de A que se obtiene tomando las filas de la a a la b y las columnas de la c a la d):

1. Se comienza con $R := A$, $Q := I_d$.

2. Para i de 1 a $d-1$ (por cada columna a reflejar):

   Se asigna $\tilde{Q} = I_d$.

   Se toma $x := A^{(i \to d, i \to i)}$ las entradas de la primera columna de la submatriz correspondiente.

   Se toma $\alpha := -signo(x_1)\,\|x\|$.

   Se hace $x := x - \alpha e_1$, con $e_1 = Id_d^{(1 \to (n-i+1), 1 \to 1)}$ el vector canónico.

   Se hace $x := \frac{x}{\|x\|}$.

   Se halla el reflector, con $\tilde{Q}^{(i \to n, i \to n)} = I_{n-i+1} - 2(xx^t)$.

   Se actualizan $Q := \tilde{Q}Q$ y $R := \tilde{Q}R$.

3. Finalmente, se traspone $Q := Q^t$.

El algoritmo finaliza con las Q y R especificadas.

Observación 32 (Método iterativo para obtener autovalores con QR). Dada una matriz A, factorizable QR por alguno de los algoritmos anteriores (para este algoritmo, conviene Householder), se pueden obtener sus autovalores, con una error ϵ menor que una tolerancia t prefijada, así:

1. Se asigna $\epsilon := \infty$.

2. Mientras que $\epsilon \geq t$:

   Se factoriza A en Q y R mediante alguno de los algoritmos anteriores.

   Se reasigna $A := RQ$.

   Se calcula L el triángulo inferior **sin diagonal** de A.

   Se asigna $\epsilon := \text{máx}_{i,j}\,|L_{ij}|$ el máximo de los valores absolutos de elementos en L.

El algoritmo finaliza con las aproximaciones a los autovalores en la diagonal de A.