Resumen teórico de la materia Métodos Numéricos[†]

Javier Gonzalo Silveira
Santiago Miguel Palladino
Facundo Matías Carreiro

1. Aritmética de la computadora

La computadora usa aritmética de dígitos finitos. Esto implica que todo número representable tiene un número de dígitos fijo y finito. La representación de un número depende de la base elegida β, la cantidad de dígitos de la mantisa t, y los límites l y u del exponente,

$$x^* = \pm\,(0, d_1 d_2 \dots d_t) \times \beta^e$$

donde $0 \leq d_i < \beta$, el exponente e cumple $l \leq e \leq u$ y el dígito $d_1 \neq 0$ (esta última condición le da el nombre de representación de punto flotante *normalizada*). El número 0 es tratado como un caso especial.

Para todo número representado vale que $\beta^{l-1} < m \leq |x^*| \leq M = (1 - \beta^{-t}) \times \beta^u$. De encontrarse fuera del rango $[m, M]$ entonces se dice que ocurrió underflow o overflow. Cabe destacar que, como podemos ver en la **Figura 1**, los números de máquina no están uniformemente distribuidos, se encuentran más concentrados para valores pequeños.

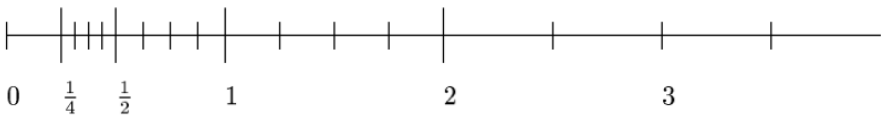

Figura 1: Números normalizados cuando $\beta = 2, l = -1, u = 2$.

Definición. El error de redondeo unitario, o ε de la máquina, es aquel valor tal que $|\delta| \leq \varepsilon = \frac{1}{2}\beta^{-t+1}$, siendo δ aquel valor que verifica $x^* = x(1 + \delta)$. Asimismo, ε determina el menor número tal que $1 + \varepsilon \neq 1$.

Definición. Error relativo y absoluto de una aproximación p^* de p se definen como

$$\begin{aligned} \varepsilon_r(p) &= \frac{|p - p^*|}{|p|} \\ \varepsilon_{abs}(p) &= |p - p^*| \end{aligned}$$

Definición. Se dice que el número p^* aproxima p con t dígitos significativos si t es el entero no negativo más grande para el cual $\varepsilon_r(p) < \frac{1}{2}\beta^{-t+1}$.

Teorema 1.1. Todos los números reales pueden ser representados con t dígitos significativos con un *error relativo* que no supera el error de redondeo unitario siendo

$$\begin{aligned} &\beta^{-t+1}, && \text{si se usa truncamiento} \\ &\tfrac{1}{2}\beta^{-t+1}, && \text{si se usa redondeo} \end{aligned}$$

† Ultima revisión: 9 de septiembre de 2008.

Dentro de los cálculos que más problemas traen cuando trabajamos con aritmética finita tenemos a la sustracción de números casi iguales en (valor absoluto), conocido como **cancelación catastrófica**, la cual genera la supresión de dígitos significativos en el resultado.

Otro cálculo que intenta evitarse fuertemente es la división por números pequeños, puesto que un error mínimo en el dividendo se traduce en uno mucho mayor en el resultado y que la falta de precisión podría ocasionar un overflow o pérdida de dígitos significativos.

Ejemplo. La estrategia de pivoteo (parcial o total) del algoritmo de eliminación de Gauss trata de evitar este último problema buscando siempre el número más grande por el que se pueda dividir. Dado que los números de punto flotante están más concentrados cerca del cero entonces al dividir por un número más grande es más probable conseguir una mejor aproximación.

Otro problema común es que sumar un número grande a uno pequeño puede hacer que el pequeño desaparezca. En ciertos casos esto no ocasiona un problema ya que, si tenemos un número de gran magnitud probablemente podamos considerar al más pequeño despreciable. Sin embargo debe tenerse mucho cuidado con el orden de las operaciones ya que si, por ejemplo, sumamos una gran cantidad de numeros pequeños entre ellos (que juntos tienen un peso considerable) y luego se lo sumamos a un número grande todo funcionará correctamente pero si vamos sumando uno por uno los números pequeños al grande entonces en cada paso el número pequeño será considerado despreciable y llegaremos a un resultado erroneo.

Definición. Se dice que un algoritmo, función o procedimiento es **inestable** si pequeños errores en alguna etapa del algoritmo (por ejemplo al principio) son ampliados en las etapas subsequentes degradando seriamente el cálculo final.

Definición. Se dice que un algoritmo, función o procedimiento está **mal condicionado** cuando pequeños cambios en los datos de entrada producen grandes cambios en la salida.

Definición. Dado un algoritmo, función u operación, se puede analizar qué coeficientes de condición y estabilidad tiene. Esto es interesante ya que permite saber si los errores con los que ya ingresan las variables se amplifican, mantienen o reducen al aplicar las operaciones. La forma de calcularlos para funciones de una variable está dada por

$$\varepsilon(f(x)) = \frac{f'(x)}{f(x)} * x * \varepsilon(x) + \varepsilon_{opf} \tag{1}$$

dónde ε_{opf} es el error intrínsico de la operación que calcula la función f, los coeficientes acompañando este término son los llamados **coeficientes de estabilidad** y los que acompañan al término $\varepsilon(x)$ son los llamados **coeficientes de condición**.

Observación. Los coeficientes de *condición* no dependen del algoritmo si no de los errores de las variables y los coeficientes de *estabilidad* si dependen del algoritmo, procedimiento u orden en que se realizen las operaciones.

2. Factorización LU

Sea $A \in \mathbb{R}^n$ queremos hallar, si existen, tres matrices $L, U, P \in \mathbb{R}^n$ tal que L sea una matriz triangular inferior, U sea una matriz triangular superior, P sea una matriz de permutacion, y que $A = PLU$.

La complejidad de resolver el sistema $Ax = b$ es de $O(n^3)$. Si logramos obtener la factorizacion LU de A podemos resolver el sistema equivalente en $O(n^2)$ tomando $y = Ux$ y resolviendo el sistema $Ly = Pb$ y luego resolviendo el sistema $Ux = y$ ambos en $O(n^2)$ por ser U y L matrices triangulares (se resuelve el sistema con "forward substitution" primero y el segundo con "back substitution"). Si debemos resolver múltiples instancias de la forma $Ax = b_i$, el costo será de $O(n^3)$ para la primera (para averiguar L y U), pero $O(n^2)$ para las siguientes instancias.

Si se puede efectuar la eliminación gaussiana en el sistema $Ax = b$ sin intercambio de filas, entonces es posible factorizar A como el producto de L y U. Si se requiere intercambio de fila, se puede encontrar P (matriz de permutación) tal que $PA = LU$. Para garantizar que esta factorización sea única por lo general se pide que haya unos en la diagonal de L (Doolitle) o en la de U (Crout).

Otra cosa buena que tiene esta factorización, es que si se la calcula, por ejemplo, durante el algoritmo de eliminacion gaussiana, se puede ir almacenando los coeficientes usados para poner ceros en una fila en el mismo lugar donde debería haber un nuevo cero, requiriendo así el algoritmo que calcula LU no más memoria que para almacenar la matriz original.

Teorema 2.1. Una matriz A no singular tiene factorización LU si cumple alguna de las siguientes equivalencias:

- No emerge un pivote igual a cero durante la eliminación gaussiana.
- Todas las submatrices principales de A son no singulares.

Observación. Además, para toda matriz A no singular existe una P matriz de permutación tal que PA tiene factorización LU.

La factorización se realiza con $U = M_{(n-1)}M_{(n-2)} \dots M_{(3)}M_{(2)}M_{(1)}A_{(1)}$, donde M_k es la k-ésima matriz de la transformación gaussiana, formada por 1s en su diagonal, todos ceros excepto la columna k que a partir de la fila $k+1$ contiene los multiplicadores (negados) usados para colocar ceros en cada fila de $A_{(k)}$. L es la inversa del producto de las matrices $M_{(n-1)}M_{(n-2)} \dots M_{(3)}M_{(2)}M_{(1)}$, lo que es igual a la matriz triangular inferior con unos en su diagonal cuyos únicos elementos debajo de la diagonal son los multiplicadores usados para triangular A.

Este método sufre de inestabilidad numérica cuando la matriz A tiene un numero de condición grande. Se puede utilizar pivoteo parcial (intercambiar filas para dividir por el numero de mayor modulo en la primera columna), o pivoteo total (intercambiar filas y columnas para dividir por el numero de mayor modulo en la submatriz que aun no factorizamos), para mejorar la estabilidad numérica de este método (que no es más que la eliminación gaussiana).

3. Factorización QR

Si podemos escribir una matriz $A = QR$ donde Q es ortonormal y R triangular superior, luego podríamos resolver el sistema $Ax = b$ de la siguiente forma:

$$Ax = b \iff QRx = b \iff Q^tQRx = Q^tb \iff Rx = Q^tb$$

y como R es triangular superior, esto se puede resolver en $O(n^2)$. La pregunta es ahora, ¿cómo conseguir esta factorización, cómo encontrar estas Q y R?

Teorema 3.1. Si A es una matriz cuadrada en $\mathbb{R}^{n\times n}$ entonces existen Q ortogonal tal que $A = QR$, con R triangular superior. Si además A es no singular, existen únicas Q y R tales que R tiene diagonal positiva.

Dado que el producto de matrices ortonormales es una matriz ortonormal, una forma razonable de encontrar esta factorización es triangular superiormente a A multiplicándola por matrices ortonormales. Si pensamos en operaciones que conservan la norma (ya que esto caracteriza las matrices ortonormales), las rotaciones y las reflexiones surgen como opciones interesantes. En esta idea se basan los dos métodos siguientes para encontrar la factorización QR de cualquier matriz $A \in \mathbb{R}^{n\times n}$.

Reflexiones (Householder): Una reflexión de householder es una tranformación que toma un vector y lo refleja sobre un plano. Esta tranformación se puede construir de manera que refleje un vector de forma tal de anular todas sus coordenadas menos una.

La idea de su uso para triangular una matriz es obtener un conjunto de matrices reflectoras ortogonales $Q_{(1)}Q_{(2)} \dots, Q_{(i)}$ cada una de las cuales al multiplicar a A reflejan la columna i de manera de generar ceros en $a_{ii}, a_{i(i+1)}, \dots, a_{in}$. En el caso particular de $i = 1$, si llamamos x a la columna en la cual se quieren colocar todos ceros excepto en la primer posición, buscamos Q tal que $Qx = y$, con $y^T = (\pm\|x\|_2, 0, \dots, 0)$. Por haber garantizado que $\|x\|_2 = \|y\|_2$, existe un teorema que nos dice que esa reflexión Q existe, es única y tiene el comportamiento buscado.

De hecho, también se puede probar que $Q = I - 2uu^T$ tomando $u = \frac{x-y}{\|x-y\|_2}$ es una forma de construir la matriz buscada. Multiplicando A por esta matriz $Q_{(1)}$ colocamos ceros en toda la columna 1 debajo de la

primer fila. Para el resto de las columnas se construye Q_i de forma análoga pero analizando la submatriz inferior derecha de la nueva A, y luego completando la matriz obtenida con la identidad para obtener la matriz Q_i que pone nuevos ceros en la columna i y no altera los ceros anteriores.

Si analizamos la forma de definir $Q = I - 2uu^T$, veremos que esta matriz de tranformación lo que hace es reflejar todos los vectores respecto del subespacio ortogonal a u. Al construir u se puede elegir el signo del vector y de manera de evitar una cancelación con x y reducir los errores de cálculos.

En la práctica lo más eficiente es nunca construir directamente la matriz Q, sino operar distribuyendo para no hacer el producto uu^T, con lo cual la factorización tiene complejidad $O(\frac{2}{3}n^3)$.

Rotaciones (Givens): Una matriz de rotación P difiere de la identidad en cuatro elementos como máximo. Estos tienen la forma $p_{ii} = p_{jj} = cos(\theta)$ y $p_{ij} = -p_{ji} = sen(\theta)$ para algún θ e $i \neq j$. Se puede demostrar que con cualquier matriz de rotación P, PA difiere de A sólo en las filas i y j. Además, para cualquier $j \neq i$ se puede elegir un ángulo θ tal que PA tenga un elemento cero para $(PA)_{ij}$. Además, toda matriz de rotación P es ortogonal ya que por definición $PP^T = I$.

Usando esto, para obtener cada cero que necesito debajo la diagonal de A para triangularla, multiplico por una matriz de rotación $Q_{(ik)}$ que rota ese vector (la columna i de A) de manera tal que coloca un cero en la coordenada k; esto lo repito para cada coordenada que quiero anular, y a su vez para cada columna. La matriz Q es el producto de todas las $Q_{(ik)}$ traspuesto, y sigue siendo ortogonal. R es $R = Q^T A$.

Una ventaja sobre HouseHolder es que puede poner ceros en posiciones específicas de una matriz alterando muy poco la estructura original, algo útil si por ejemplo se quiere trabajar con matrices esparsas o con la forma de Hessenberg. La complejidad del método es de $O(\frac{4}{3}n^3)$.

Observación. Una gran particularidad y ventaja de ambos métodos es que son numéricamente muy estables.

Ejemplo. Esta factorización se usa tanto en el algoritmo QR para calcular autovalores como para resolver el problema de cuadrados mínimos trabajando con el sistema equivalente $Rx - Q^T b$.

4. Resolucion de sistemas con matrices especiales

En muchos casos cuando modelamos matemáticamente un problema nos encontramos que por la manera de representar los datos la matriz construida cumple con ciertas características especiales. A continuación nombramos algunas de ellas que, usadas correctamente, pueden ayudarnos a resolver el problema de manera más eficiente.

Definición: Sea $A \in \mathbb{R}^{n \times n}$ se la llama **diagonal dominante** cuando cumple que

$$|a_{ii}| \geq \sum_{j \neq i}^{n} |a_{ij}| \text{ para todo } i \in \{1, \ldots, n\}$$

o sea, en cada fila, el módulo del elemento en la diagonal es mayor o igual a la suma del módulo del resto de los elementos de la fila. Se llama **estrictamente diagonal dominante** cuando la desigualdad es estricta.

Teorema 4.1. Las matrices diagonal dominantes cumplen con las siguientes propiedades

- A es no singular.
- Tiene factorización LU sin pivoteo y los cálculos son estables respecto al crecimiento de los errores de redondeo.
- Jacobi y Gauss-Seidel convergen.
- Ejemplo de uso: En el sistema lineal para determinar el spline cúbico que interpola una serie de puntos.

Teorema 4.2. Si A tiene factorización LU y A es simétrica, se puede probar fácilmente que $A = LDL^t$ donde D es diagonal. Si además la diagonal es de elementos > 0, podemos escribir $A = L\sqrt{D}\sqrt{D}L^t = L\sqrt{D}(L\sqrt{D})^t = KK^t$ (llamada **factorización de Cholesky** donde K es triangular inferior con diagonal estrictamente positiva).

Definición. Se dice que una matriz matriz $A \in \mathbb{R}^{n\times n}$ es **simétrica definida positiva** si cumple $A = A^t$ y $(\forall x \in \mathbb{R}^n, x \neq 0)\ x^tAx > 0$ (o $\geq$ para **semi-definida positiva**).

Teorema 4.3. Las matrices simétricas definidas positivas cumplen con las siguientes propiedades

- Sus menores principales son no singulares, luego también tiene factorización LU.
- A tiene factorización de Cholesky ya que $d_{ii} > 0$ para cada $i \in \{1, \ldots, n\}$.
- $(a_{ij})^2 < a_{ii}a_{jj}$ para cada $i \neq j$.
- A simétrica es d.p. si y sólo si sus primeras submatrices principales tienen determinante positivo.
- Son aptas para el método de direcciones conjugadas.

Definición. Una matriz $A \in \mathbb{R}^{n\times n}$ es **banda-pq** cuando cumple

$$a_{ij} = 0 \ \text{ si } \ j < i - p \ \text{ ó } \ j > i + q$$

para $p, q \geq 0$, o sea: A es una matriz para la que bajo la p-ésima subdiagonal tiene sólo ceros y sobre la q-ésima superdiagonal tiene sólo ceros. Cuando p y q son cero la matriz es diagonal.

Teorema 4.4. Las matrices banda-pq cumplen con las siguientes propiedades

- Se puede ahorrar mucho espacio guardando sólo los elementos de la banda sabiendo que el resto son ceros.
- Si A tiene factorización LU entonces L es banda q y U es banda p.
- Si se aprovecha esta propiedad de A, con por ejemplo A tridiagonal, se puede factorizar A en $A = LU$ (Crout) con sólo $O(5n)$ multiplicaciones y $O(3n)$ sumas.

5. Inestabilidad numérica al resolver sistemas lineales

Dado un sistema lineal de la forma $Ax = b$, por los problemas que uno se encuentra al tratar con aritmética finita puede llegar a una solución $\tilde{x}$ tal que se cumpla $A\tilde{x} = \tilde{b}$ con $|b - \tilde{b}|$ pequeño. ¿Puede uno suponer que la solución es suficientemente buena? La respuesta en general es **no** y la cota de error dependerá del 'condicionamiento' de la matriz.

Una matriz no singular A se dice mal condicionada cuando para el sistema $Ax = b$ un pequeño cambio relativo en b puede causar un cambio relativo grande en la solución x. El grado de mal condicionamiento está indicado por el número de condición de A,

$$\kappa(A) \stackrel{\text{def}}{=} \|A\|\|A^{-1}\|$$

para cualquier norma consistente. Se cumple que siempre $\kappa(A) \geq 1$.

Sea $\tilde{x}$ la solución aproximada hallada del sistema $Ax = b$, siendo A no singular; sea el residuo $r = b - A\tilde{x} = b - \tilde{b}$, entonces valen las siguientes cotas para el error absoluto y relativo

$$\begin{array}{rcccl} \varepsilon_{abs}(x) & \stackrel{\text{def}}{=} & \|x - \tilde{x}\| & \leq & \|r\| * \|A^{-1}\| \\ \varepsilon_r(x) & \stackrel{\text{def}}{=} & \frac{\|x-\tilde{x}\|}{\|x\|} & \leq & \|A\| * \|A^{-1}\| * \frac{\|r\|}{\|b\|} = \kappa(A) * \varepsilon_r(b) \end{array}$$

Un número de condición elevado se debe a que la matriz tiene dos o más columnas 'casi' linealmente dependientes, con lo que la matriz resulta 'casi' singular, y se aproxima a tener infinitas soluciones para el sistema.

Observación. Visto geométricamente en el plano, con $A \in \mathbb{R}^{2\times 2}$, las dos rectas determinadas por las columnas de A son casi paralelas, con lo cual determinar el punto de intersección se presta a mucho error. De hecho, una perturbación mínima en el sistema, producto de los errores, genera que el punto de intersección se mueva drásticamente.

Ejemplo. Sea una matriz A cuyas columnas son los vectores $(1; 1{,}0001)^t$ y $(2; 2)^t$, $b = (3; 3{,}0001)^t$. La solución verdadera es igual a $(1; 1)$, mientras que por errores de redondeo se llega a $(3; 0)$. En este caso, el número de condición de A es ≈ 60000, y el error relativo de x es igual a 2.

6. Métodos iterativos para sistemas lineales

Los métodos iterativos tienen como objetivo aproximar la solución de un sistema de ecuaciones lineales partiendo de un caso base inicial y generando aproximaciones sucesivas que deberían acercarse al la solución exácta del sistema. Difícilmente se usen estos métodos para resolver sistemas lineales chicos, ya que el tiempo necesario para conseguir la exactitud necesaria supera a la de un métodos directo, pero se vuelven eficientes en el caso de sistemas grandes con un alto porcentaje de elementos ceros.

Definición. Dada $A \in \mathbb{R}^{n\times n}$ se define al **radio espectral** de A, $\rho(A)$ como $|\lambda|$ dónde λ es el autovalor de A de mayor módulo.

Teorema 6.1. Algunos resultados de álgebra importantes en métodos iterativos son que dada $A \in \mathbb{R}^{n\times n}$

- $\rho(A) < 1$ sii $\lim_{k\to\infty}(A^k) = 0$
- $\rho(A) < 1$ entonces $(I - A)$ es no singular y $(I - A)^{-1} = \sum_{k=0}^{\infty} A^k$
- $\rho(A) \leq \|A\|$ para toda norma inducida

Teorema 6.2. La sucesión $\{x^k\}_{k\geq 0}$ tal que $x^k = Tx^{k-1} + C$ converge a la solución del sistema $x = Tx + C$ para cualquier x_0 inicial si y solo si $\rho(T) < 1$. Su demostración se basa en las propiedades enunciadas arriba, y lo que nos permite probar es que para ciertos casos particulares de esa sucesión y para algunos tipos de matrices, la sucesión converge para cualquier x_0 (se puede probar viendo únicamente que $\rho(T) < 1$).

Método de Jacobi: En este marco tenemos el método de Jacobi, cuya idea consiste en resolver la i-ésima ecuación de $Ax = b$ para x_i en función de las demás variables, y generar x_i^{k+1} a partir de los componentes de x^k cuando $k \geq 1$. O sea,

$$x_i^{k+1} = \frac{b_i - \sum_{\substack{j=1 \\ j\neq i}}^{n} a_{ij}x_j^k}{a_{ii}} \tag{2}$$

para $i = 1, \ldots, n$ suponiendo $a_{ii} \neq 0$. Es fácil ver la forma matricial de expresar esto. Dado $Ax = b$, descomponemos $A = (D - L - U)$ (donde D es la matriz diagonal cuya diagonal coincide con la de A y $-L$ y $-U$ las partes estrictamente triangular inferior y superior de A. Con esta notación queda la forma matricial de Jacobi

$$x^{k+1} = D^{-1}(L + U)x^k + D^{-1}b \tag{3}$$

Se puede probar que si A es una matriz estrictamente diagonal dominante, tomando $T_j = D^{-1}(L + U)$ y $C_j = D^{-1}b$, el método de Jacobi converge para todo x_0 inicial (probando que $\rho(T_j) < 1$).

Método de Gauss-Seidel: Un método que introduce una idea para mejorar Jacobi es el de Gauss-Seidel. La idea detrás del mismo es, si se quiere calcular x_i^k, como ya fueron calculados $x_1^k, \ldots, x_{i-1}^k$, usar estos nuevos valores como mejores aproximaciones en lugar de $x_1^{k-1}, \ldots, x_{i-1}^{k-1}$. Matricialmente esta idea se traduce en la siguiente iteración

$$x^{k+1} = (D - L)^{-1}Ux^k + (D - L)^{-1}b \tag{4}$$

Nuevamente para que $(D - L)$ sea no singular basta con pedir $a_{ii} \neq 0$. Al igual que Jacobi, si A es una matriz e.d.d. entonces, tomando $T = (D - L)^{-1}U$ y $C = (D - L)^{-1}b$ este método converge. Generalmente el método de Gauss-Seidel tiene una mejor velocidad de convergencia, pero habrá casos donde Jacobi convergerá y este no, y viceversa (ya que recordemos que ambos tienen matrices de iteración distintas, de las cuales una puede cumplir las hipótesis de convergencia y la otra no). Otro hecho interesante es que a menor $\rho(A)$ mayor velocidad de convergencia.

Por último, está el detalle de que el algoritmo de Gauss-Seidel requiere menos memoria, ya que puede pisar las partes de la nueva aproximación ya calculadas. Jacobi en cambio debe guardar y hacer toda una copia.

7. Direcciones conjugadas

El método de direcciones conjugadas se aplica para resolver sistemas $Ax = b$ en los casos en los que la matriz A es **simétrica definida positiva**. Es un método iterativo que se aproxima a la solución x^* del sistema moviéndose sobre direcciones A-conjugadas [2].

Esto se logra convirtiendo el problema anterior en un problema de optimización, que es hallar el mínimo de una función $Q(x)$. Esta función se elige especialmente para que sus mínimos coincidan con las soluciones de $Ax = b$.

$$Q(x) = \frac{1}{2}x^t Ax - b^t x \tag{5}$$

Teorema 7.1. El gradiente de $Q(x)$ es $\nabla Q(x) = Ax - b$ que es igual a cero cuando se satisface $Ax = b$. Es posible afirmar que todos estos puntos críticos son mínimos pues al derivar nuevamente el gradiente se obtiene A y como A es definida positiva todo punto crítico de $Q(x)$ es un mínimo.

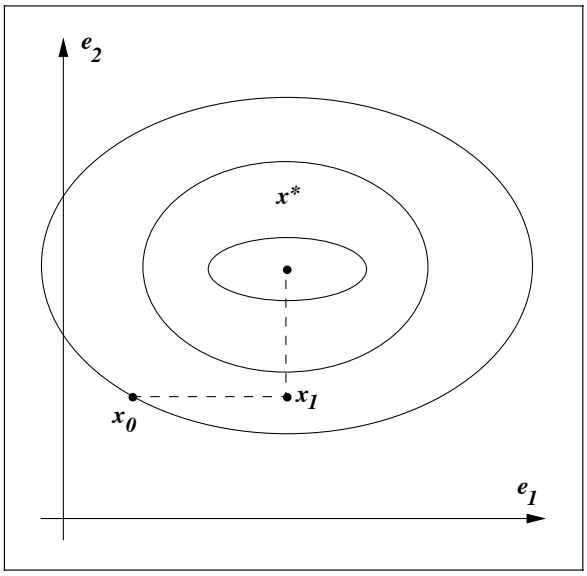

Figura 2: Paraboloide y curvas de nivel en $\mathbb{R}^2$. Minimizaciones siguiendo los vectores canónicos.

Un esquema para los algoritmos de optimización es una sucesión $x_{k+1} = x_k + \alpha_k d_k$ que tiende a la solución, siendo d_k la dirección en la que avanza el método y α_k cuánto avanza. El algoritmo debe determinar cómo elegirlos.

Calculando $Q(x + \alpha d)$ se obtiene una expresión en función de esos tres factores. Siendo que se pretende minimizar $Q(x)$, se puede derivar en función de α y se obtiene la manera de calcularla, y siempre existe un valor d posible para que el movimiento resulte positivo.

$$\alpha = \frac{d^t(b - Ax)}{d^t Ad}$$

Ejemplo. En el caso de $\mathbb{R}^2$, como puede verse en la **Figura 2**, la función obtenida es un paraboloide, las curvas de nivel son elipses con centro en el origen donde se alcanza el mínimo. Si A es diagonal, los ejes de los elipses están alineados con los ejes de coordenadas, con lo que tomando los vectores canónicos como direcciones, en cada iteración obtengo la coordenada en ese eje de la solución final. Es decir, en cada iteración, me acerco al punto en esa dirección. Por lo tanto, en 2 iteraciones el método converge.

Sin embargo, como sucede en la mayoría de los casos, A no es diagonal y los ejes de las elipses no coinciden con los vectores canónicos e_i. En esos casos, utilizarlos como direcciones de movimiento puede llevar a que el método requiera más iteraciones para converger, no necesariamente una cantidad finita. Por lo tanto, se busca moverse en otras direcciones. Aquí es donde entran en juego las direcciones A-conjugadas.

Definición. Sea $A \in \mathbb{R}^{n\times n}$ definida positiva, los vectores $\{d_1, \ldots, d_n\}$ tal que $d_i \neq 0$ son direcciones *A-conjugadas* si y sólo si $d_i^t A d_j = 0$ para todo $i \neq j$. Si además se cumple que $d_i^t A d_i = 1$, son direcciones *A-ortonormales*.

Definición. Llamaremos *residuo* del sistema lineal a

$$r(x) \stackrel{\text{def}}{=} \nabla Q(x) = Ax - b$$

Teorema 7.2. Sea $A \in \mathbb{R}^{n\times n}$ simétrica definida positiva y $\{d_1, \ldots, d_n\}$ direcciones A-conjugadas,

- Las direcciones A-conjugadas resultan linealmente independientes.
- Si en la sucesión anterior se toman las direcciones A-conjugadas de la matriz A, la sucesión converge y en a lo sumo n pasos (en la teórica lo vimos sólo para direcciones A-ortonormales, vale también para A-conjugadas).
- El residuo de una iteración es ortogonal a todas las direcciones anteriores, es decir, si $r_k = b - Ax_k$, entonces $r_k^t d_i = 0$ para todo $i \in \{0, \ldots, k-1\}$.

Teorema 7.3. Sean $\{d_1, \ldots, d_n\}$ direcciones A-conjugadas y $S = [d_1, \ldots, d_n]$ una matriz que tiene como columnas a las direcciones entonces se puede definir un nuevo conjunto de variables

$$\hat{x} = S^{-1}x$$

y se define la nueva $\hat{Q}(\hat{x})$ como

$$\hat{Q}(\hat{x}) \stackrel{\text{def}}{=} Q(S\hat{x}) = \frac{1}{2}\hat{x}^t(S^tAS)\hat{x} - (b^tS)\hat{x}$$

de esta forma, S^tAS es diagonal y se vuelve al caso anterior, puesto que cada dirección e_i en el subespacio generado por S equivale a la dirección d_i en el espacio canónico. Tomando cualquier conjunto de direcciones A-conjugadas se puede aplicar el algoritmo de direcciones conjugadas.

Método del gradiente conjugado: Este método es es una forma particular de elegir las direcciones A-conjugadas para luego aplicar el algoritmo anterior. Definido el residuo en el paso k como

$$r_k \stackrel{\text{def}}{=} r(x_k) = Ax_k - b$$

la dirección en el paso k es elegida como combinación lineal entre la dirección de máximo descenso (el opuesto del gradiente, que equivale al opuesto del residuo) y la dirección anterior d_{k-1}

$$d_k = -r_k + \beta_k d_{k-1}$$

donde β_k se define bajo el requerimiento de que d_{k-1} y d_k sean A-conjugadas

$$\beta_k = \frac{r_k^t A d_{k-1}}{d_{k-1}^t A d_{k-1}}$$

siendo $d_0 = -r_0$.

Lo interesante de este método es que el cálculo de cada dirección se basa solamente en la anterior, por lo que no es necesario mantener en memoria los valores de todas las direcciones ya recorridas.

Teorema 7.4. Las direcciones así generadas verifican que para todo $i \in \{0, \ldots, k-1\}$

$$\begin{array}{ccccc} \langle r_0, \ldots, r_k\rangle & = & \langle d_0, \ldots, d_k\rangle & = & \langle r_0, Ar_0, \ldots, A^k r_0\rangle \\ & & d_k^t A d_i & = & 0 \end{array}$$

Es decir, verifican ser A-conjugadas, puesto que cada dirección generada es A-ortogonal respecto de las anteriores. También se cumple que los residuos son mutuamente ortogonales.

Definición. El subespacio $\langle r_0, Ar_0, \ldots, A^k r_0\rangle$ se denomina *subespacio de Krylov* de grado k de r_0.

Observación. Cabe destacar que el nombre Gradiente Conjugado para el método es una elección poco adecuado, puesto que son las direcciones que se utilizan las que son A-conjugadas, y no los gradientes.

Observación. El método del gradiente conjugado puede verse tanto como un algoritmo exacto para la resolución de sistemas lineales (puesto que termina como mucho en n pasos o incluso menos dependiendo de la distribución de los autovalores de la matriz) o como un método iterativo ya que va generando una sucesión de aproximaciones que convergen a x^* solución del sistema.

8. Interpolación

Muchas veces nos encontramos con un conjunto de puntos $(x_i, f(x_i))$ que provienen de una función desconocida f y nos gustaría poder "estimar" el valor de la función en algún punto $\xi \in [x_0, x_n]$ para el cual no tenemos datos. Otra razón para interpolar puede ser que la función original es demasiado complicada para tratar con ella y queremos simplificarla tomando sólo la información contenida en algunos puntos y "sintetizando" una función más simple. Las funciones interpoladoras hacen justamente lo que estamos buscando.

Es útil poder interpolar con polinomios porque son una clase de funciones muy conocida, que tiene derivadas e integrales fáciles de calcular y que también son polinomios. Los polinomios de Taylor concentran su exactitud alrededor del punto sobre el que están centrados, pero a medida que se aleja del centro deja de ser una buena aproximación, por lo que en general no sirven para intervalos medianamente grandes.

8.1. Polinomio interpolador de Lagrange

A partir de $n+1$ puntos $x_0, x_1, \ldots, x_n$ podemos obtener el polinomio de menor grado que pasa por todos ellos. Se construye un cociente $L_{n,k}(x)$ con la propiedad de que $L_{n,k}(x_i) = 0$ cuando $i \neq k$ y $L_{n,k}(x_k) = 1$. Un polinomio que cumple esto es el siguiente:

$$L_{n,k}(x) = \frac{(x-x_0)(x-x_1)\cdots(x-x_{k-1})(x-x_{k+1})\cdots(x-x_n)}{(x_k-x_0)(x_k-x_1)\cdots(x_k-x_{k-1})(x_k-x_{k+1})\cdots(x_k-x_n)} = \prod_{\substack{i=0 \\ i\neq k}}^{n} \frac{(x-x_i)}{(x_k-x_i)}$$

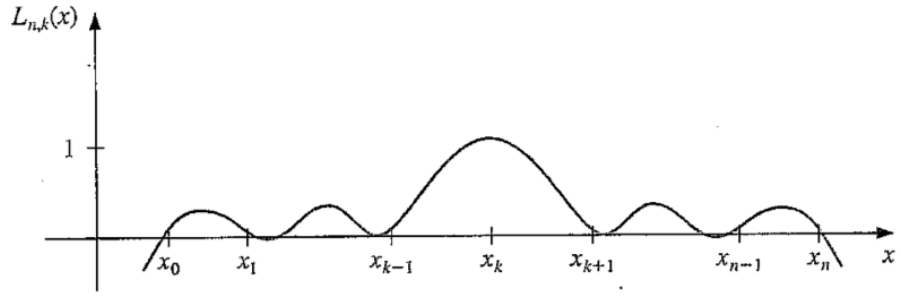

Figura 3: Polinomio $L_{n,k}(x)$.

Teorema 8.1. Si $x_0, x_1, \ldots, x_n$ son $n+1$ números distintos y si f es una función cuyos valores están dados en esos números, entonces **existe** un **único** polinomio P de grado a lo sumo n, con la propiedad de que $f(x_k) = P(x_k)$ para $k = 0, \ldots, n$. Este polinomio está dado por:

$$P(x) = \sum_{k=0}^{n} f(x_i) L_{n,k}(x)$$

Teorema 8.2. Sean $x_0, x_1, \ldots, x_n$ en $[a,b]$, $f \in C^{n+1}[a,b]$ entonces para todo x en $[a,b]$, existe ξ en $[a,b]$, que depende de x, tal que:

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

El uso de los polinomios de Lagrange plantea dos problemas inmediatos: uno es que el término del error es difícil de aplicar. El otro problema es que teniendo una aproximación de grado n, si se quiere obtener ahora la de grado $n+1$, no hay forma de aprovechar los cálculos ya hechos para ahorrar trabajo en el cálculo del nuevo polinomio. Como el polinomio es único, veremos que se puede encontrar otra forma de construirlo que permita agregar más puntos en el futuro sin un costo tan alto.

Definición. Sean k números enteros distintos $m_1, \ldots, m_k$ que cumplen $0 \leq m_i \leq n$ para cada i, se define a $P_{m_1,m_2,\ldots,m_k}(x)$ como el polinomio interpolante en los puntos $x_{m_1}, x_{m_2}, \ldots, x_{m_k}$.

Teorema 8.3. Sea f definida en $n+1$ puntos distintos $x_0, \ldots, x_n$ con x_i y x_j dos puntos del conjunto distintos entre si y $P(x)$ el polinomio de Lagrange de grado a lo sumo n que interpola a f en esos $n+1$ puntos, entonces el polinomio puede expresarse como

$$P(x) = \frac{(x - x_j)P_{0,1,\ldots,j-1,j+1,\ldots,n}(x) - (x - x_i)P_{0,1,\ldots,i-1,i+1,\ldots,n}(x)}{(x_i - x_j)}$$

De acuerdo con el **Teorema 8.3**, los polinomios interpolantes pueden generarse de manera recursiva aprovechando polinomios ya calculados.

8.2. Forma de Newton del polinomio interpolador

Definición. La diferencia dividida cero de f respecto a x_i se define como

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

y la k-ésima diferencia dividida relativa a $x_i, x_{i+1}, \ldots, x_{i+k}$ está dada por

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

Teorema 8.4. Se puede demostrar que el polinomio interpolador $P_n(x)$ se puede expresar como

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

donde $a_k = f[x_0, \ldots, x_k]$.

Usando esta definición se puede ir armando el polinomio interpolador de una serie de puntos de forma incremental, de manera que para agregar un punto más al polinomio se puede aprovechar lo ya calculado.

8.3. Splines

Los polinomios tienen una gran desventaja como interpoladores y es que cuanto mayor es el grado, más oscilan. Un procedimiento alternativo consiste en dividir el intervalo en una serie de subintervalos y en cada subintervalo construir un polinomio distinto de aproximación, basándose en la idea de que si cada intervalo usa un polinomio de un grado pequeño, se obtendrá un resultado mucho mejor que con Lagrange.

La aproximación polinómica fragmentaria más simple consiste en unir una serie de puntos mediante una serie de segmentos de rectas. La aproximación por funciones lineales ofrece una desventaja, que no se tiene la seguridad de que haya diferenciabilidad en los extremos de los subintervalos lo cual geométricamente significa que la función interpolante no es "suave" en esos puntos.

El tipo más simple de función de polinomio fragmentario diferenciable en un intervalo entero $[x_0, x_n]$ es la función obtenida al ajustar un polinomio cuadrático entre cada par consecutivo de nodos. Esto se hace construyendo una cuadrática en $[x_0, x_1]$ que concuerde con la función en x_0 y en x_1, otra cuadrática en $[x_1, x_2]$ que concuerde con la función en x_1 y en x_2 y así sucesivamente. Un polinomio cuadrático general tiene tres constantes arbitrarias, y únicamente se requieren dos condiciones para ajustar los datos en los extremos de cada intervalo, por ello existe una flexibilidad que permite seleccionar la cuadrática de modo que la interpolante tenga una derivada continua en $[x_0, x_n]$. El problema se presenta cuando hay que especificar las condiciones referentes

x	$f(x)$	First divided differences	Second divided differences	Third divided differences
x_0	$f[x_0]$			
		$f[x_0, x_1] = \frac{f[x_1] - f[x_0]}{x_1 - x_0}$		
x_1	$f[x_1]$		$f[x_0, x_1, x_2] = \frac{f[x_1, x_2] - f[x_0, x_1]}{x_2 - x_0}$	
		$f[x_1, x_2] = \frac{f[x_2] - f[x_1]}{x_2 - x_1}$		$f[x_0, x_1, x_2, x_3] = \frac{f[x_1, x_2, x_3] - f[x_0, x_1, x_2]}{x_3 - x_0}$
x_2	$f[x_2]$		$f[x_1, x_2, x_3] = \frac{f[x_2, x_3] - f[x_1, x_2]}{x_3 - x_1}$	
		$f[x_2, x_3] = \frac{f[x_3] - f[x_2]}{x_3 - x_2}$		$f[x_1, x_2, x_3, x_4] = \frac{f[x_2, x_3, x_4] - f[x_1, x_2, x_3]}{x_4 - x_1}$
x_3	$f[x_3]$		$f[x_2, x_3, x_4] = \frac{f[x_3, x_4] - f[x_2, x_3]}{x_4 - x_2}$	
		$f[x_3, x_4] = \frac{f[x_4] - f[x_3]}{x_4 - x_3}$		$f[x_2, x_3, x_4, x_5] = \frac{f[x_3, x_4, x_5] - f[x_2, x_3, x_4]}{x_5 - x_2}$
x_4	$f[x_4]$		$f[x_3, x_4, x_5] = \frac{f[x_4, x_5] - f[x_3, x_4]}{x_5 - x_3}$	
		$f[x_4, x_5] = \frac{f[x_5] - f[x_4]}{x_5 - x_4}$		
x_5	$f[x_5]$			

Figura 4: Diferencias divididas.

a la derivada de la interpolante en los extremos x_0 y x_n: no hay constantes suficientes para cerciorarse de que se satisfagan las condiciones.

La aproximación polinómica fragmentaria más común utiliza polinomios de grado tres entre cada par consecutivo de puntos y recibe el nombre de interpolación por trazadores cúbicos (o spline cúbico). Un polinomio cúbico general contiene cuatro constantes para variar, así ofrece suficiente flexibilidad para garantizar que el interpolante no sólo sea continuamente diferenciable en el intervalo, sino que además tenga una segunda derivada continua en el intervalo, aunque no se espera que las derivadas segundas coincidan con las de la función ni siquiera en los nodos.

Definición: Dada una función f definida en $[a, b]$ y un conjunto de nodos $a = x_0 < x_1 < \ldots < x_n = b$ un spline cúbico S para f es una función que cumple con las siguientes condiciones:

a. $S(x)$ es un polinomio cúbico denotado $S_j(x)$ en el subintervalo $[x_j, x_{j+1}]$ para j de 0 a $n-1$
b. $S(x_j) = f(x_j)$ para j de 0 a n
c. $S_{j+1}(x_{j+1}) = S_j(x_{j+1})$ para j de 0 a $n-2$
d. $S'_{j+1}(x_{j+1}) = S'_j(x_{j+1})$ para j de 0 a $n-2$
e. $S''_{j+1}(x_{j+1}) = S''_j(x_{j+1})$ para j de 0 a $n-2$
f. Se satisface una de las siguientes condiciones de frontera:
   - $S''(x_0) = S''(x_n) = 0$ (spline libre o **natural**)
   - $S'(x_0) = f'(x_0)$ y $S'(x_n) = f'(x_n)$ (spline **sujeto**)

Generalmente en las condiciones de frontera sujeta se logran aproximaciones más exactas, ya que usan más información acerca de la función, pero se requiere tener valores de la derivada en los extremos. Existen también otras condiciones de frontera posibles además de la natural o la sujeta.

Cuando deseo interpolar un conjunto de puntos $x_0, \ldots, x_n$, el planteo de todas las condiciones mencionadas para $S(x)$ se puede llevar a la forma de un sistema de ecuaciones tridiagonal que queda en función de uno de los cuatro coeficientes de cada spline y resulta ser estrictamente diagonal dominante, por lo que tiene solución única, puede almacenarse usando poco espacio y resolverse relativamente rápido.

9. Cuadrados mínimos

Se desea aproximar el valor de una función de la cual se tienen puntos con cierto error pero se sospecha que la fuente corresponde a una función determinada, lineal por poner un ejemplo. En este caso, lo ideal sería hallar los valores de c y d (si se trata de una recta) para los cuales se minimiza el error, es decir, la distancia entre la recta $y = cx + d$ y los puntos.

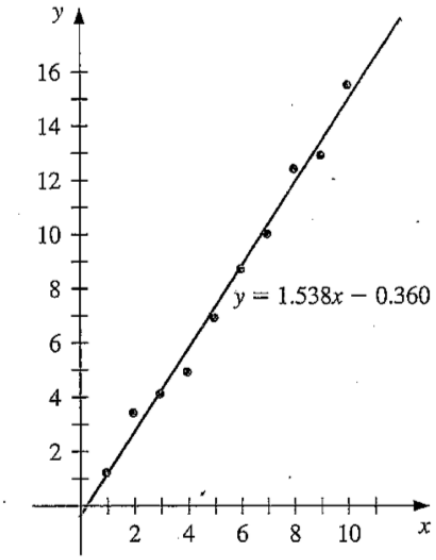

Figura 5: Aproximación de puntos por cuadrados mínimos lineales [1].

Una de las mejores forma de plantear esto es determinar los coeficientes que minimicen el error dado por la suma de los cuadrados de las diferencias entre los valores de la función aproximadora y los puntos dados, o sea que minimicen

$$\sum_{i=1}^{m} [f(x_i) - p(x_i)]^2$$

donde m es la cantidad de puntos y $p(x) = cx + d$ en el caso lineal. Existen otros criterios de minimización de error como, por ejemplo, si minimizamos

$$\sum_{i=1}^{m} |f(x_i) - p(x_i)|$$

este criterio se llama *desviación absoluta*. El problema que tiene es que la función valor absoluto no es derivable en el cero y que no necesariamente se puede obtener la solución. Otro de ellos se llama *minimax* y consta de minimizar

$$\max_{1 \leq i \leq m} |f(x_i) - p(x_i)|$$

su problema es el de no poder ser resuelto mediante métodos elementales y darle demasiada importancia a pocos elementos anómalos (outliers).

Algunas de las ventajas del método de "cuadrados mínimos" es que concede mayor valor relativo al punto que está alejado del resto de los datos, pero a su vez no permitirá que ese punto domine enteramente la aproximación. También al carecer de la función módulo (que entorpece la derivabilidad) es una elección cómoda para trabajar. Por último, existen resultados de probabilidad y estadística que también respaldan la elección de los cuadrados mínimos para el objetivo planteado.

El problema general de minimizar la suma de las diferencias al cuadrado en función de los coeficientes de la función $p(t)$ se puede resolver derivando respecto cada uno de ellos e igualando a cero las derivadas, con lo que se llega a las llamadas ecuaciones normales, sistema de ecuaciones cuya solución son los coeficientes buscados que minimizan el error.

Interpretación Matricial [3, 4]: Si tenemos un conjunto de m mediciones de la forma (x_i, y_i) y queremos aproximarlos por una función modelo $p(t)$ que puede ser expresada como

$$p(t) = \sum_{j=0}^{n} a_j \Phi_j(t)$$

donde $\{\Phi_0, \ldots, \Phi_n\}$ es un conjunto *l.i.* de funciones entonces lo que se quiere es minimizar $f(x) = \|Ax - b\|_2^2$, donde A es la matriz que tiene los números multiplicando los coeficientes a determinar, x contiene a los coeficientes y b a los y_i. El sistema $Ax = b$ queda definido como

$$\begin{bmatrix} \Phi_0(t_1) & \Phi_1(t_1) & \cdots & \Phi_n(t_1) \\ \Phi_0(t_2) & \Phi_1(t_2) & \cdots & \Phi_n(t_2) \\ \vdots & \vdots & & \vdots \\ \Phi_0(t_m) & \Phi_1(t_m) & \cdots & \Phi_n(t_m) \end{bmatrix} \begin{bmatrix} a_0 \\ a_1 \\ \vdots \\ a_n \end{bmatrix} = \begin{bmatrix} y_0 \\ y_1 \\ \vdots \\ y_m \end{bmatrix}$$

En el caso de aproximar con polinomios se puede usar $\Phi_n(t) = t^n$.

En la mayoría de los casos el sistema $Ax = b$ tal como está no tendrá solución por ser *sobredeterminado*. Por lo tanto se intenta encontrar la solución más cercana, minimizando f. Se puede probar que $\nabla f(x) = 0$ si y sólo si $A^t Ax = A^t b$.

Interpretación Geométrica: Siguiendo con la forma matricial del mismo, buscamos

$$\min_x \|Ax - b\|_2^2 = \min_{y \in Im(A)} \|y - b\|_2^2$$

Si $b \in Im(A)$ es claro que encontraremos un x tal que Ax es b y por lo tanto $y - b$ es cero, o sea, la función modelo elegida para explicar los datos coincide con cada uno de ellos. En caso de que $b \notin Im(A)$ lo que nos interesará buscar es el $y \in Im(A)$ más cercano a b, y este es justamente la proyección ortogonal de b sobre la imagen de A como puede verse en la **Figura 6**. Siguiendo este razonamiento formalmente es fácil probar que el x buscado no es otro que el que cumple $Ax = b_1$ donde b_1 es la proyección ortogonal de b en $Im(A)$.

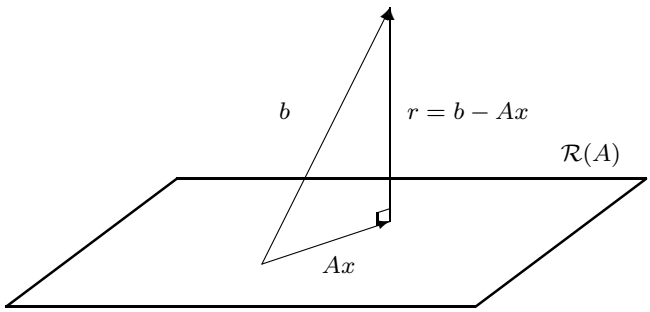

Figura 6: Interpretación geométrica de Cuadrados Mínimos [4].

Para que $\|b - y\|_2$ sea mínima, con y perteneciendo a un subespacio S, entonces es necesario que $b - y$ pertenezca al complemento ortogonal de S. Es decir, que y sea la proyección ortogonal de b sobre S, o, en este caso, la imagen de A. Por lo tanto, $b - Ax$ debe pertenecer a $Im(A)^\perp = Null(A^t)$. Para que eso suceda, debe pasar que $A^t(Ax - b) = 0$, que es la solución al problema de cuadrados mínimos.

Algunas observaciones que se desprenden del enfoque anterior es que el problema de cuadrados mínimos siempre tiene solución, esta es única sii $Null(A) = \{0\}$, y cuando $Ax = b$ tiene una única solución entonces lo mismo vale para $A^t Ax = A^t b$ y ambos sistemas coinciden en ella (permitiéndonos tratar todos los problemas de cuadrados mínimos con este enfoque). También notemos que lo bueno de resolver $A^t Ax = A^t b$ es que la matriz $A^t A$ es cuadrada, simétrica y al menos semi definida positiva (cuando la solucion es única es definida postitiva).

A pesar de las bondades del método anterior, en casos donde la matriz A está mal condicionada esta característica puede empeorar aún más en el sistema con $A^t A$, razón por la cual existen métodos alternativos numéricamente más estables para resolver el problema de cuadrados mínimos.

9.1. Cuadrados mínimos y QR

Si Q es una matriz ortogonal, minimizar la norma de $Ax - b$ o la de $Q^t(Ax - b)$ es lo mismo. Por lo tanto, se busca la descomposicion QR de la matriz A para resolver más fácilmente el sistema. Según el rango de la matriz A se puede dividir en dos casos para caracterizar mejor el conjunto de soluciones.

9.1.1. Rango completo

Se plantea el nuevo sistema $Q^tAx = Q^tb$ que equivale a $Rx = c$, donde $\hat{c}$ son los primeros m elementos de c y d los restantes. El residuo s resulta $s = c - Rx$, donde los primeros m elementos de s son iguales a $\hat{c} - \hat{R}x$ y los restantes a d. De esta forma, el cuadrado del residuo, es decir, lo que se busca minimizar, es igual a

$$\|s\|_2^2 = \|\hat{c} - \hat{R}x\|_2^2 + \|d\|_2^2 \tag{6}$$

Puesto que el segundo término, d, no depende de x, se busca minimizar el primero. Como $\hat{R}$ era no singular, entonces la solución del sistema $\hat{R}x = \hat{c}$ es única y es la solución de cuadrados mínimos. Cabe destacar que el término $\|d\|_2^2$ es la norma del residuo asociado con la solución obtenida.

$$R = \begin{bmatrix} \hat{R} \\ 0 \end{bmatrix} = \begin{bmatrix} \hat{r}_{11} & \hat{r}_{12} & \hat{r}_{13} \\ 0 & \hat{r}_{22} & \hat{r}_{23} \\ 0 & 0 & \hat{r}_{33} \\ 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix} \quad c = \begin{bmatrix} \hat{c} \\ d \end{bmatrix} = \begin{bmatrix} \hat{c}_1 \\ \hat{c}_2 \\ \hat{c}_3 \\ d_1 \\ d_2 \\ d_3 \end{bmatrix}$$

Figura 7: Ejemplo de rango completo.

9.1.2. Rango incompleto

Para rango incompleto, es necesaria otra variación de la descomposición QR: QR con pivoteo de columnas. Esta modificación, en cada iteración del algoritmo, toma la columna de mayor norma, para dejar las columnas iguales a cero para el final. El algoritmo encuentra ceros luego de r iteraciones, siendo r el rango de la matriz. Resulta $AP = QR$, siendo P una matriz de permutación, R una matriz con ceros debajo de la fila r, y cuyo menor principal R_r, o R_{11} es una matriz triangular superior no singular.

$$R = \begin{bmatrix} R_{11} & R_{12} \\ 0 & 0 \end{bmatrix} = \begin{bmatrix} r_{1,1} & \dots & r_{1,r} & 0 & \dots & 0 \\ 0 & \ddots & \vdots & \vdots & & \vdots \\ 0 & 0 & r_{r,r} & 0 & \dots & 0 \\ 0 & \dots & 0 & 0 & \dots & 0 \\ \vdots & & \vdots & \vdots & & \vdots \\ 0 & \dots & 0 & 0 & \dots & 0 \end{bmatrix}$$

Figura 8: Rango incompleto.

En este caso, resulta que los primeros r elementos del residuo s son iguales a $\hat{c} - R_{11}x_1 - R_{12}x_2$, siendo $x = (x_1, x_2)^t$, con $x_1 \in \mathbb{R}^r$; y los restantes iguales a d. Nuevamente se busca minimizar el primer término.

Se busca un x_2 cualquiera, a partir del cual hay un único x_1 tal que $R_{11}x_1 = \hat{c} - R_{12}x_2$ puesto que R_{11} es no singular, y además triangular superior, con lo que el sistema puede resolverse mediante back substitution. Se obtienen así infinitas soluciones.

De todas maneras, este método es poco estable; sumado al hecho de que hallar el rango de A no siempre es sencillo por errores de redondeo. Por eso se busca el método SVD.

9.2. Cuadrados mínimos y SVD

Teorema 9.1. Sea A en $\mathbb{R}^{m\times n}$ con $Rng(A) = r$, entonces existen $\{v_1, \ldots, v_n\}$ base ortonormal de $\mathbb{R}^n$ y $\{u_1, \ldots, u_m\}$ base ortonormal de $\mathbb{R}^m$, tal que $U^tAV = D$ diagonal con

$$D = \begin{pmatrix} D_r & 0 \\ 0 & 0 \end{pmatrix}$$

donde $D_r = \text{diag}(s_1, \ldots, s_r)$ y s_i son los valores singulares de A ordenados de mayor a menor. Los u_i son los autovectores normalizados de AA^t, y los v_i son los autovectores normalizados de A^tA.

Esto implica que $A = UDV^t$, o que $AV = UD$, siendo U y V matrices ortogonales de $m \times m$ y $n \times n$ respectivamente y D una matriz diagonal de $m \times n$, misma dimensión que A. Esta descomposición existe para cualquier matriz A.

Con el mismo criterio que en QR, puesto que U es ortogonal, minimizar la norma de $b - Ax$ es equivalente a minimizar la norma de $U^t(b - Ax) = U^tb - DV^tx$. Tomando $c = U^tb$ e $y = V^tx$, minimizar la norma de $b - Ax$ equivale a minimizar la de $c - Dy$. Calculando,

$$\|b - Ax\|_2^2 = \|c - Dy\|_2^2 = \sum_{i=1}^{r} |c_i - s_iy_i|^2 + \sum_{i=r+1}^{m} |c_i|^2$$

Por lo tanto, la solución del sistema es $y_i = c_i/s_i \ \forall i = 1, \ldots, r$. Sin embargo, si $r < m$, es decir, la matriz original A no tenia rango completo, entonces los valores y_i para $i = r + 1 \ldots m$ no están involucrados en la expresión a minimizar, y se puede tomar cualquier valor para ellos. Si se busca el x de norma mínima resultante de $x = Vy$, entonces se toman los y_i restantes iguales a cero.

El método SVD es más caro que el método QR, independientemente de cómo se organicen los cálculos. Se pueden hacer optimizaciones, como no calcular nunca la matriz U sino aplicar solamente los reflectores sobre b, o calcular solamente las primeras r columnas de V que serán las necesarias para resolver $x = Vy$, con los y_i $i > r$ iguales a cero. De todas formas, la estabilidad de SVD en casos de rango incompleto lo hace preferible por sobre QR.

Teorema 9.2. La estabilidad de la solucion de cuadrados minimos lineales esta dada por la siguiente cota, siendo $\bar{x}$ la solución al problema de cuadrados mínimos y $\bar{b}$ el proyector sobre la imagen de A,

$$\varepsilon_r(\bar{x}) \leq \chi(A)\varepsilon_r(\bar{b})$$

Donde $\chi(A) = \|A\|_2\|(A^tA)^{-1}\|_2$ es una generalización del número de condición, cuando A no tiene necesariamente inversa.

10. Sistemas de inecuaciones lineales

Es posible modificar cualquier sistema de inecuaciones para llevarlo a la forma requerida por Simplex y resolverlo utilizando ese método.

- Todo sistema de inecuaciones puede llevarse a la forma $Ax \leq b$ y viceversa, invirtiendo signos o convirtiendo igualdades en dos desigualdades opuestas.
- Todo $Ax \leq b$ puede llevarse a la forma $Ax \leq b \ \wedge \ x \geq 0$ y viceversa, reescribiendo toda x como $x^+ - x^-$, donde cada una es positiva, y duplicando las columnas de A para llevarla a la forma $(A, -A)$ con $x = (x^+, x^-)^t$.
- Todo sistema de la forma anterior puede llevarse a $Ax = b$ con $x \geq 0$ y viceversa, agregando slack variables pasando al sistema $(A, I) * (x, s)^t = b$.
- Todo sistema anterior puede pedírsele también que los coeficientes b sean positivos, multiplicando por -1 donde sea necesario.
- Puesto que para el sistema $Ax = b$ tenga un resultado equivalente al $Ax + Ia = b$ es necesario que los a valgan cero, se propone minimizar la sumatoria de los a_i; si se llega a cero, entonces se halló una solución del sistema original. Como es un problema de optimización, se puede usar SIMPLEX.

11. Simplex

Simplex es un método que permite optimizar una cierta expresión denominada funcional, respecto de ciertas variables que deben cumplir determinadas desigualdades. O sea, permite resolver un sistema de maximización sujeto a restricciones de la forma $Ax \leq b$, o de minimización sujeto a $Ax \geq b$.

Puesto que el Simplex necesita un sistema de la forma $Ax = b$ con $x \geq 0$, se agrega m slack variables (siendo m la cantidad de restricciones), para convertir el sistema a $Ax + Ia = b$.

11.1. Interpretación geómetrica

Las curvas de nivel del funcional resultan encerradas en una región determinada por las restricciones. Esta región es un polígono, y el método SIMPLEX recorre los vértices de ese polígono, puesto que el máximo se halla en uno de esos puntos.

Puede suceder que la región no sea cerrada. En ese caso, el funcional puede estar o no acotado, esto depende de su dirección de crecimiento: si crece hacia una restricción, estará acotado; si no, podrá crecer tanto como se desee.

11.2. Problemas del Simplex

Los problemas del método SIMPLEX pueden dividirse principalmente en Inicialización, Iteración y Finalización, detallados a continuación.

11.2.1. Inicialización

Siendo que se busca resolver el problema $Ax = b$, se pueden agregar las slack variables suficientes para transformar el problema en $Ax + Ia = b$, con lo cual tomando el punto inicial $x_0 = (0, \dots 0, b_1, b_1, \dots, b_m)$ siempre verifica.

Luego se toma el resultado obtenido y se lo convierte al problema original.

11.2.2. Iteración

Cada iteración se basa en elegir una variable que deja de ser básica y una no básica que ocupa su lugar.

Para elegir la variable no básica, basta con buscar aquella que en la ecuación del funcional tenga su coeficiente negativo. En caso de que no haya ninguna, el algoritmo termina exitosamente, puesto que se llegó al óptimo buscado. Si hay uno o más, se elige alguno arbitrariamente.

Para hallar la variable de salida, se busca aquella variable básica que impone la mayor restricción al valor de la variable de entrada. Es decir, aquella variable básica que se anula cuando se toma el mayor valor posible para la entrada. De haber múltiples, se genera una solución degenerada, pero esto no impide seguir con la iteración, ya que se toma cualquiera arbitrariamente.

Por lo tanto, SIMPLEX no tiene problemas de iteración.

11.2.3. Finalización

El método SIMPLEX tiene el riesgo de no terminar nunca y quedarse en loop infinito entre los mismos valores. Esto se debe a que los puntos que recorre, al ser vértices de un polígono, son finitos, por lo cual o bien arriba al resultado, o bien cicla infinitamente.

En caso de que se pase por una solución degenerada, y esto implique que el funcional no aumente sino que se mantenga constante, esas iteraciones se denominan también degeneradas. Puede suceder que luego de una determinada cantidad de iteraciones degeneradas se vuelva a aumentar el funcional, o puede que se cicle infinitamente entre los mismos puntos.

Esto se debe a que si hay dos diccionarios con las mismas variables básicas, entonces los dos diccionarios son iguales, y si se pasa dos veces por el mismo diccionario, entonces el método cicla.

Para detectar los ciclos, no es viable guardar una historia con los puntos ya recorridos por la gran longitud que puede llegar a tener dicha lista. Lo mas usual es setear un valor máximo (bastante alto) para cantidad de iteraciones degeneradas consecutivas del algoritmo.

Una solución posible a este problema de finalización es la regla del menor índice. En cualquiera de los casos en los que la iteración ofrece distintas elecciones para las variables de entrada o de salida, se toma siempre la de menor índice. En ese caso, es posible asegurar que el algoritmo siempre termina. Cabe destacar que es posible recurrir a esta regla solamente luego de una cierta cantidad de iteraciones degeneradas consecutivas, y una vez rota esa cadena, retomar algún otro método de elección que se crea conveniente; por ejemplo, que provea mayor estabilidad numérica[1].

Otra posibilidad para evitar soluciones degeneradas, y por lo tanto el loop infinito, es la perturbación. Puesto que una solución degenerada sucede cuando varias variables se hacen simultáneamente cero, se introduce una pequeña perturbación que elimina este comportamiento, y que hace que la modificación del sistema sea mínima.

Como el método de perturbación puede fallar, se utiliza el método lexicográfico, que consiste en considerar los valores ε usados como símbolos en lugar de números, y compararlos mediante los coeficientes que los acompañan. Si se utiliza esta regla siempre, es posible demostrar que el algoritmo termina.

11.3. Forma matricial del Simplex

En la interpretación estándar del simplex, mediante diccionarios, las ecuaciones son las expresiones de las variables básicas en función de las no básicas, más la del funcional. Es decir,

$$x_{B_i} = \tilde{b}_i - \sum_{x_j \in x_N} \tilde{a}_{i,j} * x_j \tag{7}$$

$$z = z^* + \sum_{x_j \in x_N} \tilde{c}_j * x_j \tag{8}$$

Las primeras ecuaciones, correspondientes a las variables, se pueden expresar matricialmente como $Ax = b$. Teniendo n variables originales y m slack variables, La matriz A se construye con n columnas a partir de los coeficientes de las variables originales, más m columnas de la identidad. Entonces, se puede reescribir lo anterior como $Ax = A_B x_N + A_N x_N = B x_B + A_N x_N = b$, siendo x_B y x_N los vectores que contienen las variables básicas y no básicas en cada iteración. De la ecuación anterior, se desprende que

$$x_B = B^{-1}b - B^{-1}A_N x_N \tag{9}$$

Es posible demostrar que la matriz B siempre tiene inversa. De esta forma, se reescriben matricialmente las primeras n ecuaciones, es decir, las variables básicas en función de las no básicas.

Para reescribir el funcional, se plantea $z = cx$, donde c tiene n componentes iguales a los resultantes de la función a minimizar, y m componentes nulos, provenientes de las slack variables. Luego c se descompone en c_B y c_N, con los coeficientes correspondientes a las variables básicas y no básicas en cada iteración (las cuales, recordemos, no se corresponden necesariamente con las originales y las slack). Por ende, $z = c_N x_N + c_B x_B$. Reemplazando el valor de x_B obtenido, se llega a

$$z = c_B B^{-1} b + (c_N - c_B B^{-1} A_N) x_N \tag{10}$$

Donde $c_B B^{-1} b$ es z^* y $(c_N - c_B B^{-1} A_N)$ son los coeficientes de las variables no básicas x_N en función de las cuales se expresa el funcional.

11.4. Simplex revisado

Al principio de cada iteración del algoritmo de Simplex revisado, se llega con los valores calculados de $x_B^* = B^{-1}b$ y la matriz B.

[1] Hay un criterio que determina la 'zero tolerance', se eligen valores muy pequeños para distintas operaciones y cualquier valor menor a ese se considera problemático. Dependiendo de la operación, puede considerarse que el valor es cero (por ejemplo, para analizar si un coeficiente es negativo en el funcional) o descartarse la elección de variables realizada y elegir otra si es posible (en el caso de encontrarse con una división por un número cercano a cero).

Lo primero es hallar qué variable tiene un coeficiente negativo entre los de las variables no básicas en la ecuación del funcional. Para ello es necesario hallar dichos coeficientes $c_N - c_B B^{-1} A_N$. Como no es necesario calcularlos todos, se los calcula individualmente hasta hallar alguno que sirva. Esto define los primeros dos pasos del algoritmo.

- Calcular el valor intermedio $y = c_B B^{-1}$ resolviendo el sistema $yB = c_B$
- Hallar la variable no básica tal que $c_j - y * a < 0$, siendo a la columna de A_N correspondiente a c_j

Lo siguiente es encontrar la columna d de $B^{-1}A_N$ que corresponde a la variable que *entra* a la base. Esto surge a partir de que $x_B = x_B^* - B^{-1}A_N x_N$, con lo cual x_B pasa de x_B^* a $x_B^* - td$.

Como d es la columna de $B^{-1}A_N$ que corresponde a la variable que sale, entonces $d = B^{-1}a$, donde a era la columna que se elegía en el segundo paso del algoritmo. Entonces, los siguientes pasos del algoritmo se basan en calcular esa columna d y en el máximo valor t posible, tal que se las restricciones se respeten.

- Hallar d mediante el sistema $Bd = a$
- Hallar el mayor t tal que $x_B^* - td$, recordar que se entra a la iteración con x_B^* calculado

La componente que queda igualada a cero en $x_B^* - td$ es la que sale de la base. Si no es posible hallar t, entonces la solución del problema no está acotada y puede aumentarse tanto como se quiera.

El último paso del algoritmo actualiza la matriz B para la próxima iteración y recalcula los x_B^*.

- Se recalcula x_B^* como $x_B^* - td$
- Se setea el valor de la variable que entra a la base igual a t
- Se reemplaza la columna de B correspondiente a la variable que sale con la columna a usada para hallar d

12. Ceros de funciones

Son métodos iterativos que permiten hallar las raíces de funciones no lineales. En general, para algoritmos iterativos, deben determinarse criterios de parada y calcularse el orden de convergencia.

12.1. Orden de convergencia

Definición. Sea $\{\alpha_n\}_{n\geq 0}$ una sucesión que tiende a α, y sea $\{\beta_n\}_{n\geq 0}$ una sucesión que tiende a cero, entonces la sucesión $\{\alpha_n\}_{n\geq 0}$ tiene orden de convergencia β_n si

$$|\alpha_n - \alpha| \leq k|\beta_n|$$

para algun $k > 0$ y a partir de un n suficientemente grande.

Definición. Sea $\{\alpha_n\}_{n\geq 0}$ una sucesión que tiende a α, si se cumple

$$\lim_{n\to\infty} \frac{|\alpha_{n+1} - \alpha|}{|\alpha_n - \alpha|^p} = k$$

para algún $k > 0$ entonces la sucesión tinde a α con orden de convergencia p.

12.2. Criterios de parada

Hay distintos criterios de parada para un algoritmo iterativo de búsqueda de ceros, ninguno de ellos lo suficientemente seguro. Tienden a ser preferibles los que utilizan el error relativo en lugar del absoluto, y se los combina junto con cantidad de iteraciones.

- $|x_n - x_{n-1}| < \varepsilon$, la diferencia entre dos soluciones es menor a ε.

- $\frac{|x_n - x_{n-1}|}{|x_n|} < \varepsilon$, la diferencia *relativa* entre dos soluciones es menor a ε.
- $|f(x_n)| < \varepsilon$, el valor de la función se acerca a cero "lo suficiente".
- $|f(x_n) - f(x_{n-1})| < \varepsilon$, entre dos iteraciones me acerco al cero menos que ε.
- $\frac{|f(x_n) - f(x_{n-1})|}{|f(x_n)|} < \varepsilon$, entre dos iteraciones me acerco *relativamente* al cero menos que ε.
- $\#\text{iters} > k$, limite en la cantidad de iteraciones.

Un caso problemático típico es la serie geométrica, que tiende a cero, aunque no sólo no tiene raíces sino que incluso diverge.

12.3. Bisección

Es el método de búsqueda binaria, se basa en el Teorema de Bolzano-Weierstrass, requiere solamente continuidad de la función y hallar dos puntos iniciales a y b tal que el signo de la función sea distinto en los dos puntos. Su convergencia, si bien es lineal, está garantizada. Tiende a usarse para aproximarse a un entorno de la solución y luego utilizar métodos más veloces pero que requieren de un intervalo inicial más acotado.

El método consta en, dados dos puntos iniciales que cumplan las propiedades anteriormente mencionadas, partir el intervalo a la mitad y generar el punto $c = \frac{b-a}{2}$. Para la próxima iteración se usarán los puntos $\{a, c\}$ si cumplen $f(a)f(c) < 0$ y $\{c, b\}$ si en cambio se cumple $f(c)f(b) < 0$.

Observación. El error en el paso n es $\varepsilon_n = |p_n - p| \leq \frac{b-a}{2^n}$.

12.4. Punto fijo

Los problemas de búsqueda de raíces y los de punto fijo son equivalentes, ya que dado el problema de encontrar la p tal que $f(p) = 0$, podemos definir una función g con un punto fijo en p, por ej. con $g(x) = x - f(x)$, de manera que cuando p es punto fijo de g, también es raíz de f.

Teorema 12.1. Si g es continua en $[a, b]$ y $g(x)$ pertenece a $[a, b]$ para todo x en $[a, b]$, entonces g tiene un punto fijo en $[a, b]$. Si además $g'(x)$ existe en (a, b) y $|g'(x)| \leq k < 1$ para toda x en (a, b), entonces el punto fijo en $[a, b]$ es único.

Para aproximar el punto fijo de una función defino la sucesión $p_n = g(p_{n-1})$. Si esta sucesión converge, lo hace al punto fijo. En la **Figura 9** se puede ver el comportamiento de la iteración de punto fijo para varias funciones, algunas divergentes y otras convergentes.

Teorema 12.2. Sea g continua en $[a, b]$ tal que $g(x)$ pertenece a $[a, b]$ para todo x en $[a, b]$. Además supongamos que existe g' en (a, b) y una constante $0 < k < 1$ tal que $|g'(x)| \leq k$ para todo x en (a, b), entonces para cualquier número p_0 en $[a, b]$, la sucesión de punto fijo converge al único punto fijo p en $[a, b]$.

Corolario 12.3. El error absoluto del paso n es $|p_n - p| \leq k^n * max(p_0 - a, b - p_0)$. La convergencia de la iteración puede ser monótona o alternante.

Observación. Si $0 < g'(x) < 1$ entonces si p_0 está a la derecha (izquierda) del punto fijo, siempre converge por la derecha (izquierda). Si $g'(x) < 0$, converge alternadamente, y sé que el punto fijo está "dentro".

Teorema 12.4. Si la iteración de punto fijo converge, $g(x) \in C^n$, $g'(p) = g''(p) = \cdots = g^{(n-1)}(p) = 0$ y $g^{(n)}(p) \neq 0$ entonces $p_{n+1} = g(p_n)$ tiene orden de convergencia n.

12.5. Newton-Raphson

El método de Newton-Rhapson es muy usado en la resolución de ecuaciones no lineales. Las hipótesis son mucho más fuertes que para el método de bisección pero ese es el precio que pagamos para tener una velocidad de convergencia más rápida.

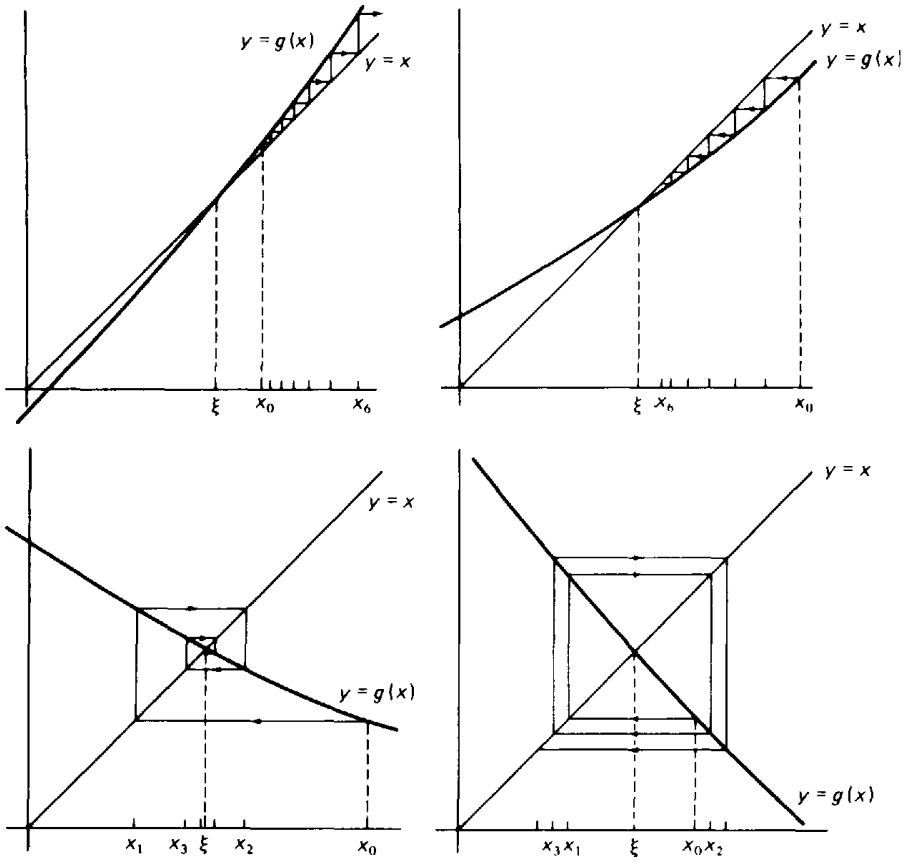

Figura 9: Convergencia de punto fijo.

Derivación por convergencia cuadrática: Supongamos que tengo una función generica de punto fijo

$$g(x) = x - h(x)f(x)$$

y quiero encontrar los ceros de $f(x)$. Si pido que $h(x) \neq 0$ entonces $g(x) = x$ sólo cuando $f(x) = 0$. Además

$$g'(x) = 1 - h'(x)f(x) - f'(x)h(x)$$

pero como quiero que $f(x) = 0$, me queda

$$g'(x) = 1 - f'(x)h(x)$$

Para obtener convergencia al menos cuadrática, pido que la derivada de g sea cero en ese punto, o sea

$$\begin{aligned} g'(p) &= 0 \\ 1 - f'(p)h(p) &= 0 \\ h(p) &= \frac{1}{f'(p)} \end{aligned}$$

entonces con

$$h(x) = \frac{1}{f'(x)}$$

me aseguro que esto se cumpla, y el punto fijo de g es raiz de f, además como la derivada en p es cero, la convergencia es cuadrática.

Teorema 12.5. Sea f en $C^2[a, b]$, $f(p) = 0$, $f'(p) \neq 0$ entonces existe $\delta > 0$ tal que si p_0 está en el intervalo $[p - \delta, p + \delta]$, la sucesión de Newton

$$x_{n+1} = x_n - \frac{f(x_n)}{f'(x_n)} \tag{11}$$

converge a p cuadraticamente.

Gráficamente, como muestra la **Figura 10**, la aproximación se obtiene usando tangentes sucesivas. Comenzando con la aproximación inicial p_0, la siguiente aproximación p_1 es la intersección con el eje x de la línea tangente a la gráfica de f en $(p_0, f(p_0))$, y así sucesivamente.

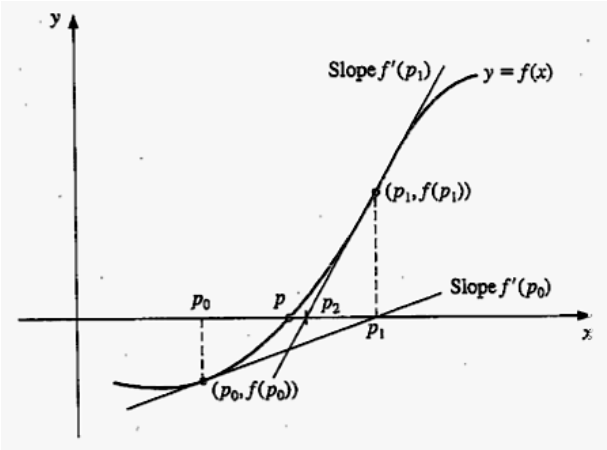

Figura 10: Convergencia del método de Newton-Rhapson.

Derivación por Taylor: Otra forma de llegar a lo mismo es con el polinomio de Taylor de grado 1 de $f(x)$. Supongamos que $f(p) = 0$, con p^* una aproximación de p de manera que $f(p^*) \neq 0$ pero $|p^* - p|$ es pequeño, entonces

$$f(x) = f(p^*) + f'(p^*)(x - p^*) + f''(\xi(x))\frac{(x - p^*)^2}{2!}$$

Suponemos que como $|p^* - p|$ es pequeño, entonces al elevarlo al cuadrado queda más pequeño aún y podemos omitir todo el último término, o sea que cuando $x = p$ tenemos

$$0 = f(p^*) + (p - p^*)f'(p^*)$$

Despejando p de esta ecuación sale la iteración de Newton

$$p = p^* - \frac{f(p^*)}{f'(p^*)}$$

En este caso, la suposición de que $|p^* - p|$ es suficientemente pequeño sería falsa si y no estuviera lo suficientemente cerca de p, causando la divergencia del método. En algunos casos, no todos, esto es así. En la demostración de la convergencia del método, se puede ver que el valor de la constante k que acota a la derivada indica la rapidez de convergencia del método, disminuyendo a cero a medida que el procedimiento avanza.

El método de Newton es muy poderoso, pero presenta un grave problema: la necesidad de conocer el valor de la derivada de f en cada aproximación, lo que con frecuencia puede ser un cálculo complejo con muchas operaciones o ser un dato no disponible, ya que quizás ni siquiera se conoce la forma analítica de la función.

12.6. Método de la secante

Este método surge como una variante de Newton-Raphson, eliminando el cálculo de la derivada de f en cada iteración. La derivada es aproximada por un cociente incremental. Geométricamente, comienza con dos aproximaciones iniciales p_0 y p_1, la aproximación p_2 es la intersección en x de la recta secante que une $(p_0, f(p_0))$ y $(p_1, f(p_1))$. La aproximación p_3 es la intersección de la recta que une $(p_1, f(p_1))$ y $(p_2, f(p_2))$ y así sucesivamente. La fórmula de iteración es:

$$X_n = X_{n-1} - f(X_{n-1}) \underbrace{\frac{X_{n-1} - X_{n-2}}{f(X_{n-1}) - f(X_{n-2})}}_{\text{aproximación de } f'(x)^{-1}} \tag{12}$$

El precio que se paga para prescindir de la derivada es la velocidad de convergencia, que es más lenta que Newton: es superlineal. Una desventaja de este método también es que cuando $f(X_n)$ y $f(X_{n-1})$ se parecen mucho, la resta trae problemas numéricos al trabajar con aritmética finita.

12.7. Regula Falsi

El método de Regula Falsi genera aproximaciones del mismo modo que el de la secante, pero ofrece una prueba para asegurarse de que la raíz quede entre dos iteraciones sucesivas. Primero se eligen las aproximaciones iniciales p_0 y p_1 con $f(p_0)f(p_1) < 0$. La aproximación p_2 se escoge de la misma manera que con el método de la secante: como la intersección en x de la línea que une $(p_0, f(p_0))$ y $(p_1, f(p_1))$. Para decidir con cuál secante calcular p_3 se verifica que $f(p_2)f(p_1) < 0$. Si esto se cumple, p_1 y p_2 encierran un raíz, entonces uso como p_3 la intersección con el eje x de la recta que una a $(p_1, f(p_1))$ y $(p_2, f(p_2))$. Por otro lado, si $f(p_2)f(p_1) > 0$, elegimos p_3 como la intersección del eje x con la recta que pasa por $(p_0, f(p_0))$ y $(p_2, f(p_2))$, intercambiando después los índices de p_0 y p_1.

Como X_n y X_{n-1} tienen distinto signo se evita la resta de dos números muy parecidos obteniendo una mayor estabilidad del algoritmo. De la misma forma la raíz está siempre acotada entre dos valores de distinto signo (aunque es importante notar que el tamaño de este intervalo puede no tender a cero).

Por otro lado, este método suele requerir más cálculos que el método de la secante y no tiene la convergencia supralineal asegurada.

13. Sistemas no lineales

Si tenemos el problema de resolver un sistema de n ecuaciones no lineales con n incógnitas de la forma:

$$\begin{cases} f_1(x_1, \ldots, x_n) &= 0 \\ f_2(x_1, \ldots, x_n) &= 0 \\ \vdots & \vdots \\ f_n(x_1, \ldots, x_n) &= 0 \end{cases} \tag{13}$$

lo podemos reescribir como encontrar el cero de una nueva función $F : \mathbb{R}^n \to \mathbb{R}^n$ definida como $F(x_1, \ldots, x_n) = (f_1, \ldots, f_n)$, o sea, resolver el sistema de ecuaciones no lineales se traduce en hallar un valor X^* que satisfaga $F(X^*) = 0$.

13.1. Punto fijo en varias variables

Los métodos de punto fijo en una variable tienen su versión generalizada en n variables. El sistema (13) puede reescribirse como

$$\begin{cases} x_1 &= g_1(x_1, \ldots, x_n) \\ x_2 &= g_2(x_1, \ldots, x_n) \\ \vdots & \vdots \\ x_n &= g_n(x_1, \ldots, x_n) \end{cases}$$

convirtiendo el problema original en uno de punto fijo. Analogamente al caso de una variable enunciamos las siguientes condiciones de existencia y unicidad de punto fijo.

Teorema 13.1. Sea $G : \mathbb{R}^n \to \mathbb{R}^n$ y $D \subseteq \mathbb{R}^n$. Si G es continua en D tal que $G(D) \mapsto D$. Además supongamos que existen las derivadas parciales de G en D y una constante k positiva tal que

$$\left| \frac{\delta g_i}{\delta x_j} \right| \leq \frac{k}{n} < \frac{1}{n}$$

para todo elemento en D, entonces para cualquier X_0 en D, la sucesión de punto fijo $X_{n+1} = G(X_n)$ converge al único punto fijo X^* en D.

13.2. Método de Newton en varias variables

El método de Newton también tiene su generalización a varias variables. Siendo que el método en una variable estaba determinado por la sucesión

$$x_{k+1} = x_k - \frac{f(x_k)}{f'(x_k)}$$

que se deducía del modelo lineal $f(x_k) + f'(x_k)(x - x_k) = 0$, en varias variables se determina por

$$X_{k+1} = X_k - J^{-1}(X_k)F(X_k)$$

Observación. Sin embargo, en una implementación lo que se efectúa es la resolución del sistema $J(X_k)(X_{k+1} - X_k) = -F(X_k)$, para evitar el costoso (y poco estable) cálculo de la inversa del Jacobiano.

El método de Newton, si el Jacobiano es continuo en una región D alrededor del X^*, es superlineal. Si verifica continuidad de Lipschitz, es decir, existe una $\beta_L > 0$ tal que $\|J(X_0) - J(X_1)\| \leq \beta_L \|X_0 - X_1\|$ para cualquier par $X_0, X_1 \in D$, entonces es cuadrático. Estos órdenes se dan en un cierto entorno alrededor del X^*.

También vale que si $F(p) = 0$, es decir, la función se anula en el punto fijo, y $J(p)$ es inversible, entonces en un entorno la iterada converge cuadráticamente.

Observación. Los problemas que tiene el método de Newton son varios,

- Si no se comienza lo suficientemente cerca del punto, el algoritmo puede no converger.
- Si el Jacobiano es singular, puede haber iteraciones indefinidas.
- Puede ser dificil calcular el Jacobiano.
- El jacobiano en la raiz puede ser singular, con lo que la convergencia de Newton cae a lineal.
- Puede ser muy caro calcular el valor exácto de Newton en una iteración alta.

Por eso se utilizan métodos alternativos.

13.3. Métodos de cuasi-Newton

También conocidos como métodos de la secante. El objetivo es eliminar el principal problema que tenía Newton, el Jacobiano, por alguna otra matriz. Así se pasa a la forma

$$F(X_k) + B_k(X_{k+1} - X_k)$$

Una condición que se le pide a la matriz B es la denominada condición secante:

$$B_k(X_{k-1} - X_k) = F(X_{k-1}) - F(X_k)$$

Es análogo al método de la secante en una variable, con la diferencia de que en este caso sólo determina n de los n^2 elementos de la matriz B. Esto da lugar a distintos métodos.

Una desventaja de estos métodos es que son superlineales en lugar de cuadráticos. Otra desventaja de estos métodos es que a diferencia de Newton, no se corrigen a sí mismos. El método de Newton generalmente corregirá el error de redondeo con iteraciones sucesivas, pero no así el de Broyden.

13.4. Método de Broyden

Broyden busca una matriz B_k lo más parecida posible a la de la iteración anterior, buscando la B que cumpla la condición secante y minimice $\|B - B_{k-1}\|_2$, de esta forma B es única.

$$\begin{aligned} B_k &= B_{k-1} + \frac{(J_{k-1} - B_{k-1}S_{k-1})S_{k-1}^t}{S_{k-1}^t S_{k-1}} \\ S_{k-1} &= X_k - X_{k-1} \\ J_{k-1} &= F(X_k) - F(X_{k-1}) \end{aligned}$$

El X_k puede obtenerse del sistema $B(X_k)(X_{k+1} - X_k) = -F(X_k)$ al igual que en el caso de Newton, o bien apelar a la fórmula de Sherman-Morrison para calcular fácilmente la inversa. Esta fórmula indica que si una matriz A es no singular y $y^t A^{-1} x \neq -1$, entonces $A + xy^t$ es no singular y su inversa es igual a

$$A^{-1} - \frac{A^{-1} x y^t A^{-1}}{1 + y^t A^{-1} x}$$

En la fórmula de Broyden, tomando adecuadamente los coeficientes, es posible hallar fácilmente la inversa de B exclusivamente mediante productos entre matrices, sin necesidad de invertir ninguna matriz.

$$\begin{aligned} A &= B_{k-1} \text{ (cuya inversa está precalculada de la iteración anterior)} \\ x &= J_{k-1} - B_{k-1} S_{k-1} \\ y^t &= \frac{S_{k-1}^t}{S_{k-1}^t S_{k-1}} \end{aligned}$$

De esta forma, se puede calcular $X_{k+1} = X_k - B_k^{-1} F(X_k)$, bajando las cuentas a $O(n^2)$.

14. Cálculo de autovalores

Una forma "directa" de calcular los autovalores de una matriz es obtener las raíces del polinomio característico $P(\lambda) = det(A - \lambda I)$ y luego se pueden obtener los vectores característicos (autovectores) resolviendo el sistema lineal asociado a cada autovalor. Claramente para matrices grandes es difícil obtener $P(\lambda)$, aún si se lo consiguiera no es fácil calcular todas las raíces de cualquier polinomio de n-ésimo grado. Es por eso que los métodos de aproximación se presentan como una buena opción para estas situaciones.

14.1. Método de la potencia

Para aplicar este método pedimos como hipótesis que $A \in \mathbb{R}^{n \times n}$ tenga n autovalores tales que $|\lambda_1| > |\lambda_2| \geq \cdots \geq |\lambda_n|$ con una base de autovectores asociados $\{v_1, v_2, \ldots, v_n\}$. Es muy importante la existencia del autovalor dominante λ_1 y que el valor inicial x_0 elegido no sea ortogonal al autovector asociado a λ_1.

Teorema 14.1. Como los autovectores forman una base entonces existen constantes $\beta_1, \ldots, \beta_n$ tal que para todo x se cumple

$$x = \sum_{i=1}^{n} \beta_i v_i$$

si multiplicamos a izquierda por A^k nos queda

$$A^k x = \sum_{i=1}^{n} \beta_i A^k v_i = \sum_{i=1}^{n} \beta_i \lambda_j^k v_i$$

y sacando factor común λ_1^k nos deja

$$A^k x = \lambda_1^k \sum_{i=1}^{n} \beta_i \left(\frac{\lambda_j}{\lambda_1} \right)^k v_i$$

como $\lambda_1 > \lambda_i$ para $i \neq 1$ entonces se cumple

$$\lim_{k \to \infty} A^k x = \lim_{k \to \infty} \lambda_1^k \sum_{i=1}^{n} \beta_i \left(\frac{\lambda_j}{\lambda_1} \right)^k v_i = \lim_{k \to \infty} \lambda_1^k \beta_1 v_1$$

La idea es usar la sucesión $x_k = A^k x$, sucesión que, por lo anteriormente mencionado, se puede reescribir como

$$x_k = \lambda_1^k (\beta_1 v_1 + \varepsilon_k)$$

donde ε_k tiende a cero para k tendiendo a infinito. Notemos que esta sucesión tiende a cero o diverge dependiendo del valor de λ_1, es por eso que la idea del método de las potencias es normalizar esta sucesión y aplicarle una función de manera que tienda a λ_1.

Entonces, sea $\Phi : \mathbb{R}^n \to \mathbb{R}$ una función continua, tal que $\Phi(\alpha x) = \alpha\Phi(x)$ y que sea distinta de cero siempre que x no se anule. Definimos entonces una nueva sucesión

$$\mu_k = \frac{\Phi(x_{k+1})}{\Phi(x_k)}$$

la cual es equivalente a

$$\lambda_1 \frac{\Phi(\beta_1 v_1 + \varepsilon_{k+1})}{\Phi(\beta_1 v_1 + \varepsilon_k)}$$

quien para k tendiendo a infinito converge a

$$\lambda_1 \frac{\Phi(\beta_1 v_1)}{\Phi(\beta_1 v_1)} = \lambda_1$$

He aquí un método para aproximar el autovalor de mayor módulo, sólo resta definir quién es la función $\Phi(x)$. Una buena opción es definirla como $\Phi(x) = \|x\|_\infty$, el componente de máxima magnitud del vector x (y si hay más de uno, que sea el primero). El método de las potencias para obtener el autovalor λ_1 quedaría entonces:

x_0 `= x inicial`
`Para` $k = 1$ `a` M
`1.` $y = Ax$
`2.` $r = \frac{\Phi(y)}{\Phi(x)}$
`3.` $x = y$

La velocidad de convergencia para este método dependerá de la velocidad con la que $\varepsilon_k$tienda a cero, y esto a su vez dependerá de cuán chico será el cociente $\frac{\lambda_2}{\lambda_1}$, o sea, de cuán lejos está el autovalor de mayor módulo del que le sigue en magnitud y en consecuencia del resto de los autovalores.

14.2. Método de la potencia inversa

Esta variación del método de la potencia sirve para cuando A es no singular para calcular el autovalor de módulo mínimo. La idea es que si se puede aplicar el método de la potencia sobre A^{-1}, el resultado será la la inversa del autovalor de menor magnitud de A. Notar que las hipótesis del método anterior exigen sobre A que $|\lambda_1| \geq |\lambda_2| \geq \ldots > |\lambda_n|$ y nuevamente que $\{v_1, \ldots, v_n\}$ sea una base de autovectores, de esta forma obtendremos como resultado $\frac{1}{\lambda_n}$.

El método de la potencia tiene la desventaja de que al inicio no se sabe si la matriz tiene o no un único autovalor dominante. Tampoco se sabe cómo seleccionar x_0 para asegurar que no sea ortogonal al vector característico asociado al autovalor dominante, en caso de que exista.

Referencias

[1] R. L. Burden, J. D. Faires, *Análisis Numérico.*

[2] J. Nocedal, S. Wright, *Numerical Optimization.*

[3] D. Watkins, *Fundamentals of Matrix Computations.*

[4] D. Dhalquist, *Numerical Methods in Scientific Computing.*