1.II.2. Sistemas de ecuaciones lineales: métodos directos.

Manuel Palacios
Departamento de Matemática Aplicada
Centro Politécnico Superior
Universidad de Zaragoza

Primavera 2001

Contents

References

[1] Burden, R. L. and Faires, J. D.: Análisis Numérico. *Grupo Editorial Iberoamerica*, 1985.

[2] Gasca, M.: Cálculo numérico: resolución de ecuaciones y sistemas. *Librería Central*, 1987.

[3] Hairer, E.: Introduction l'Analyse Numérique. *Université de Genéve, Dept. de Mathématiques*, 1993.

[4] Conde, C. y Winter, G.: Métodos y algoritmos del álgebra numérica. *Editorial Reverté*, 1990.

[5] Griffel, D. H.: Linear Algebra and its applications. *Ellis Horwood*, 1989.

[6] Strang, G..: Linear Algebra and its Applications. 3^{th} ed. *Hardcourt Brace Jovanovich, Inc.*, 1988.

Introducción

El problema que se pretende resolver en este capítulo es el de un sistema de m ecuaciones lineales con n incógnitas definido en forma matricial mediante:

$$A\,x = b,$$

en donde $A \in \mathbb{R}^{(m,n)}$ y $b \in \mathbb{R}^m$ son datos y $x \in \mathbb{R}^n$ es el vector incógnita. Explícitamente se puede escribir en la forma:

$$\begin{array}{ccccccccc} a_{11}\,x_1 & + & a_{12}\,x_2 & + & \dots & + & a_{1n}\,x_n & = & b_1 \\ a_{21}\,x_1 & + & a_{22}\,x_2 & + & \dots & + & a_{2n}\,x_n & = & b_2 \\ \vdots & & \vdots & & \dots & & \vdots & & \vdots \\ a_{m1}\,x_1 & + & a_{m2}\,x_2 & + & \dots & + & a_{mn}\,x_n & = & b_m \end{array} \tag{1}$$

Habitualmente supondremos que $m = n$ y que el sistema posee solución única, es decir, $\det A \neq 0$, o bien, $\operatorname{rg} A = n$.

1 Resolución de sistemas triangulares

Un sistema triangular con $m = n$ tiene la forma:

$$\begin{array}{ccccccccc} r_{11}\,x_1 & + & r_{12}\,x_2 & + & \dots & + & r_{1n}\,x_n & = & c_1 \\ & & r_{22}\,x_2 & + & \dots & + & r_{2n}\,x_n & = & c_2 \\ & & & & \ddots & & \vdots & & \vdots \\ & & & & & & r_{nn}\,x_n & = & c_n \end{array} \tag{2}$$

Estos sistemas se resuelven fácilmente mediante el algoritmo siguiente denominado de *sustitución regresiva*:

$$x_n = \frac{c_n}{r_{nn}}, \quad x_i = \frac{1}{r_{ii}}(c_i - \sum_{j=i+1}^{n} r_{ij}\,x_j), \text{ para } i = n-1, \dots, 1 \tag{3}$$

El coste computacional de esta resolución es el siguiente: para cada $i = n, n-1, \dots, 1$, se requiere una división, $n-i$ productos y $n-i$ sumas, es decir, un total de

$$\sum_{i=1}^{n} 1 + 2\,(n-i) = n + n(n-1) = n^2$$

operaciones aritméticas elementales.

2 Matrices elementales

Recordemos que se denomina *matriz elemental*, $P_{ij}(\lambda)$, a una matriz que multiplicada a la izquierda (o a la derecha) de una matriz A realiza sobre ella una operación elemental, es decir,

a) una permutación de filas (columnas), P_{ij},

b) una substitución de la fila i-ésima por ella más la j-ésima fila (la j-esima columna por ella más la i-esima fila) multiplicada por un escalar λ, $P_{ij}(\lambda)$;

c) una multiplicación de la fila (columna) j-ésima por un escalar $\lambda \neq 0$, $P_{jj}(\lambda)$.

Estas operaciones elementales son las operaciones aritméticas más sencillas que transforman un sistema lineal en otro equivalente, es decir, con las mismas soluciones.

Los tres tipos de matrices elementales están definidos en la forma siguiente: a) P_{ij} es la matriz unidad con sus filas i y j cambiadas; b) $P_{ij}(\lambda)$ es la matriz unidad pero su elemento ij es igual a λ; c) $P_{jj}(\lambda)$, $\lambda \neq 0$, es la matriz unidad pero su elemento jj es igual a λ. Por ejemplo, si tomamos matrices de orden 3:

$$P_{23} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 0 & 1 \\ 0 & 1 & 0 \end{bmatrix}, \quad P_{23}(\lambda) = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & \lambda \\ 0 & 0 & 1 \end{bmatrix}, \quad P_{22}(\lambda) = \begin{bmatrix} 1 & 0 & 0 \\ 0 & \lambda & 0 \\ 0 & 0 & 1 \end{bmatrix}.$$

Propiedad 2.1 *Todas las matrices elementales son regulares y sus inversas son las siguientes:*

$$P_{ij}^{-1} = P_{ij}, \quad ,(P_{ij}(\lambda))^{-1} = P_{ij}(-\lambda), \quad ,(P_{22}(\lambda))^{-1} = P_{22}(1/\lambda).$$

3 Eliminación gaussiana y factorización LU.

Supongamos dado el sistema (1), con $m = n$ y $\det A \neq 0$. Si $a_{11} \neq 0$, se puede eliminar la variable x_1 en las ecuaciones de la 2 a la n calculando

$$l_{i1} = \frac{a_{i1}}{a_{11}} \tag{4}$$

y realizando las operaciones elementales sobre las filas de la matriz ampliada (A, b) definidas por las matrices elementales $P_{i1}(-l_{i1}), i = 2, 3, \ldots, n$.

De esta manera, el sistema equivalente obtenido puede ser escrito en la forma siguiente:

$$\begin{array}{ccccccccc} a_{11}^{(1)}\, x_1 & + & a_{12}^{(1)}\, x_2 & + & \ldots & + & a_{1n}^{(1)}\, x_n & = & b_1^{(1)} \\ & & a_{22}^{(1)}\, x_2 & + & \ldots & + & a_{2n}^{(1)}\, x_n & = & b_2^{(1)} \\ \vdots & & \vdots & & \ldots & & \vdots & & \vdots \\ & & a_{n2}^{(1)}\, x_2 & + & \ldots & + & a_{nn}^{(1)}\, x_n & = & b_n^{(1)} \end{array} \tag{5}$$

donde

$$\begin{array}{ll} a_{1j}^{(1)} = a_{1j}, & a_{ij}^{(1)} = a_{ij} - l_{i1}\, a_{1j} \\ b_1^{(1)} = b_1, & b_i^{(1)} = b_i - l_{i1}\, b_1 \end{array} \quad \text{para } i = 2, 3, ..., n \tag{6}$$

(evidentemente, si $a_{11} = 0$, hay que cambiar la primera fila de (A, b) por otra para obtener $a_{11} \neq 0$, lo que siempre es posible, ya que $\det A \neq 0$.

El sistema (5) contiene un subsistema de dimensión $n - 1$ sobre el cual puede repetirse el procedimiento para eliminar x_2 en las ecuaciones 3 a n, es decir, hay que realizar las operaciones elementales sobre las filas de la matriz (5) definidas por las matrices elementales $P_{i2}(-l_{i2}), i = 2, 3, \ldots, n$, siendo $l_{i2} = a_{i2}^{(1)}/a_{22}^{(1)}$.

Después de $n - 1$ etapas de este tipo

$$(A, b) \longrightarrow (A^{(1)}, b^{(1)}) \longrightarrow (A^{(2)}, b^{(2)}) \longrightarrow \ldots \longrightarrow (A^{(n-1)}, b^{(n-1)}) =: (R, c)$$

se obtiene el sistema triangular

$$\begin{array}{ccccccccc} r_{11}\, x_1 & + & r_{12}\, x_2 & + & \ldots & + & r_{1n}\, x_n & = & c_1 \\ & & r_{22}\, x_2 & + & \ldots & + & r_{2n}\, x_n & = & c_2 \\ & & & & \ddots & & \vdots & & \vdots \\ & & & & & & r_{nn}\, x_n & = & c_n \end{array} \tag{7}$$

que se resuelve en n^2 operaciones aritméticas en la forma indicada en (3).

Teorema 3.1 *Sea* $\det A \neq 0$. *El método de eliminación gaussiana permite construir matrices* L *y* U *únicas tales que:*

$$P\,A = L\,U, \tag{8}$$

siendo P *una matriz de permutación y*

$$L = \begin{pmatrix} 1 & & & \\ l_{21} & 1 & & \\ \vdots & \ddots & \ddots & \\ l_{n1} & \dots & l_{n,n-1} & 1 \end{pmatrix}, \qquad U = \begin{pmatrix} u_{11} & u_{12} & \dots & u_{1n} \\ & u_{22} & \dots & u_{2n} \\ & & \ddots & \vdots \\ & & & u_{nn} \end{pmatrix} \tag{9}$$

La fórmula (8) se denomina factorización L U *de la matriz* A.

Notemos que la matriz de permutación es el producto de todas las matrices necesarias para conseguir que ningún pivote salga nulo. Esta matriz es una matriz ortogonal, por lo tanto, $\det P = \pm 1$.

Demostr.: Supongamos que todas las permutaciones necesarias han sido realizadas antes de comenzar la eliminación gaussiana; seguiremos llamando A a la matriz $P\,A$.

El primer paso de la eliminación consiste en la multiplicación de la matriz A por las matrices elementales $P_{i1}(-l_{i1}), i = 2, 3, \dots, n$, es decir, realizando dicho producto por la matriz

$$L_1 = P_{n1}(-l_{n1})...P_{21}(-l_{21}), \tag{10}$$

Analogamente, el paso k-ésimo consiste en la multiplicación de la matriz A de este paso por las matrices elementales $P_{ik}(-l_{2k}), i = k+1, , \dots, n$, es decir, realizando dicho producto por la matriz

$$L_k = P_{nk}(-l_{nk})...P_{k+1,k}(-l_{k+1,k}), \tag{11}$$

Por lo tanto, las sucesivas matrices $A^{(k)}$ son:

$$L_1\,A = A^{(1)}, \quad L_2\,A^{(1)} = A^{(2)}, \quad L_{n-1}\,A = A^{(n-1)} = U$$

En consecuencia,

$$U = L_{n-1}\,L_{n-2}\,...L_1\,A \Longrightarrow A = (L_{n-1}\,L_{n-2}\,...L_1)^{-1}\,U$$

Pero

$$(L_{n-1}\,L_{n-2}\,...L_1)^{-1} = (P_{n,n-1}(-l_{n,n-1})...P_{n1}(-l_{n1})...P_{21}(-l_{21}))^{-1} =$$

$$P_{21}(-l_{21})^{-1}...P_{n1}(-l_{n1})^{-1}...P_{n,n-1}(-l_{n,n-1})^{-1} = P_{21}(l_{21})...P_{n1}(l_{n1})...P_{n,n-1}(l_{n,n-1}) = L.$$

La unicidad de la factorización anterior se prueba como sigue. Supongamos que hubiese dos factorizaciones del tipo mencionado:

$$A = L\,U = \tilde{L}\,\tilde{U}$$

Entonces, ya que $\tilde{L}$ es regular, se tendría: $\tilde{L}^{-1}\,L\,U = \tilde{U}$; como U y $\tilde{U}$ son triangulares superiores y $\tilde{L}$ y L son triangulares inferiores, $\tilde{L}^{-1}\,L$ deber ser la unidad, es decir, $\tilde{L} = L$, y, en consecuencia, $\tilde{U} = U$. ■

La matrices $L_1, L_2, ...$ utilizadas en la eliminación gaussiana

$$L_1 = \begin{pmatrix} 1 & & & & \\ -l_{21} & 1 & & & \\ -l_{31} & 0 & 1 & & \\ \vdots & \vdots & \ddots & \ddots & \\ -l_{n1} & 0 & \dots & 0 & 1 \end{pmatrix}, \qquad L_2 = \begin{pmatrix} 1 & & & & \\ 0 & 1 & & & \\ 0 & -l_{32} & 1 & & \\ \vdots & \vdots & \ddots & \ddots & \\ 0 & -l_{n2} & \dots & 0 & 1 \end{pmatrix}, \tag{12}$$

son denominadas **matrices de Fröbenius**.

En el caso general, $m \geq n$ y rang $A \leq n$, el proceso de eliminación gaussiana se realiza de la misma manera, pero teniendo en cuenta que los pivotes a utilizar siempre deben ser no nulos, para lo que habrá que considerar, posiblemente, como antes alguna matriz de permutación.

Cálculo del determinante de una matriz.

La fórmula (8) implica que $\det P \,.\, \det A = \det L \,.\, \det U$. Pero como $\det P = (-1)^\epsilon$, donde ϵ es el número de permutaciones necesarias en la eliminación, se obtiene

$$\det A = (-1)^\epsilon \, u_{11} \, u_{22} \, ... \, u_{nn} \tag{13}$$

Observar que la factorización $L\,U$ permite, teniendo en cuenta la matriz U, determinar el rang A.

Resolución de sistemas lineales.

En la práctica, se encuentra muchas veces que es preciso resolver una serie de sistemas lineales que tienen todos la misma matriz de coeficientes y en los que los términos independientes sólo se conocen después de haber resuelto el sistema anterior. Por eso, para su resolución se suele proceder en la forma siguiente:

a) obtención de la factorización $A = L\,U$ y

b) resolución de los sistemas triangulares

$$L\,c = P\,b, \quad U\,x = c$$

por sustitución progresiva y regresiva (algoritmo (3)), respectivamente.

Coste computacional de la factorización $L\,U$.

Para el paso de A a $A^{(1)}$ se requieren: $n-1$ divisiones (ver (4)), $2(n-1)\,(n-1)$ multiplicaciones y adiciones (ver (6)).

Para el paso de $A^{(1)}$ a $A^{(2)}$ se requieren: $n-2$ divisiones, $2(n-2)\,(n-2)$ multiplicaciones y adiciones. Y así sucesivamente.

El coste total será:

$$\sum_{k=1}^{n-1}(k-1) + 2\sum_{k=1}^{n-1}(k-1)^2 \approx \frac{n(n-1)}{2} + 2\int_0^n x^2\,dx = \frac{n(n-1)}{2} + \frac{2}{3}n^3 \text{ operaciones.}$$

La elección del pivote.

En la eliminación gaussiana es preciso, de entrada, elegir una ecuación con el elemento $a_{i1} \neq 0$ (que se denomina *pivote*), a partir de la cual se elimina x_1 de todas las restantes ecuaciones. La elección de esta ecuación, es decir, del pivote, puede tener una gran influencia en el resultado numérico, si se realizan las operaciones en coma flotante.

Ejemplo 3.2 *(Forsythe). Considérese el sistema lineal*

$$\begin{aligned} 1.00 \cdot 10^{-4}\, x_1 + 1.00\, x_2 &= \quad 1.00 \\ 1.00\, x_1 + 1.00\, x_2 &= \quad 2.00 \end{aligned} \tag{14}$$

que tiene la solución exacta

$$x_1 = \frac{1}{0.9999} = 1.00010001..., \quad x_2 = \frac{0.9998}{0.9999} = 0.99989998...$$

Realizando las operaciones en coma flotante con tres cifras significativas exactas (en base 10), se aplica la eliminación gaussiana en los dos casos siguientes:

a) Tomando como pivote $a_{11} = 1.00 \cdot 10^{-4}$, se obtiene:

$$l_{21} = a_{21}/a_{11} = 1.00 \cdot 10^4, \quad a_{22}^{(1)} = 1.00 - 1.00 \cdot 10^4 = -1.00 \cdot 10^4, \quad b_2^{(1)} = 2.00 - 1.00 \cdot 10^4 = -1.00 \cdot 10^4$$

En consecuencia:

$$x_2 = \frac{b_2^{(1)}}{a_{22}^{(1)}} = 1.00(\text{¡exacto!}), \quad x_1 = \frac{b_1 - a_{12}\, x_2}{a_{11}} = \frac{1.00 - 1.00 \cdot 1.00}{1.00\, 10^{-4}} = 0$$

bien diferente de la solución exacta.

b) Cambiando las filas de la matriz ampliada, ahora el pivote es 1.00 y la eliminación gaussiana permite obtener:

$$l_{21} = 1.00 \cdot 10^{-4}, \quad a_{22} = 1.00 - 1.00 \cdot 10^{-4} = 1.00, \quad b_2^{(1)} = 1.00 - 2.00 \cdot 1.00 \cdot 10^{-4} = 1.00$$

Resultando,

$$x_2 = \frac{b_2^{(1)}}{a_{22}^{(1)}} = 1.00 \text{ (¡ exacto!)}, \quad x_1 = \frac{b_1 - a_{12}\, x_2}{a_{11}} = \frac{2.00 - 1.00 \cdot 1.00}{1.00} = 1.00 \text{ (¡exacto!)}.$$

Para entender mejor donde se ha perdido información esencial, veamos los subproblemas suma, resta, multiplicación y división separadamente y conozcamos su "condición".

Condición de un problema

Considérese una aplicación $\mathcal{P} : \mathbb{R}^n \longrightarrow \mathbb{R}$, es decir, el problema de calcular $\mathcal{P}(x)$ para los datos $x = (x_1, \ldots, x_n)$. ¿Cómo influyen las perturbaciones en los datos x sobre el resultado $\mathcal{P}(x)$?.

Definición 3.3 *Se denomina condición, κ, de un problema $\mathcal{P}$ al número real más pequeño tal que*

$$\frac{|\hat{x}_i - x_i|}{|x_i|} \leq eps \Longrightarrow \frac{|\mathcal{P}(\hat{x}) - \mathcal{P}(x)|}{|\mathcal{P}(x)|} \leq \kappa \cdot eps \tag{15}$$

Se dice que un problema $\mathcal{P}$ está bien condicionado, si κ no es muy grande, en caso contrario, $\mathcal{P}$ está mal condicionado.

En esta definición, *eps* representa un número pequeño; si *eps* fuese la precisión del ordenador (por ejemplo, 10^{-6} en simple precisión), $\hat{x}_i$ puede ser considerado como el número de máquina de x_i. Téngase en cuenta que el número de condición no depende del algoritmo utilizado para resolver el problema $\mathcal{P}$, sino que depende de los datos x_i y del propio problema $\mathcal{P}$.

Ejemplo 3.4 *Multiplicación de dos números reales. Dados $x_i, i = 1, 2$, considérese el problema de calcular $\mathcal{P}(x_1, x_2) = x_1 \cdot x_2$.*

Tomando los valores perturbados:

$$\hat{x}_1 = x_1(1+\epsilon_1), \quad \hat{x}_2 = x_2(1+\epsilon_2), \quad |\epsilon_i| \leq eps$$

se tiene

$$\frac{\hat{x}_1 \cdot \hat{x}_2 - x_1 \cdot x_2}{x_1 \cdot x_2} = (1+\epsilon_1)\cdot(1+\epsilon_2) - 1 = \epsilon_1 + \epsilon_2 + \epsilon_1 \cdot \epsilon_2$$

Como *eps* es un número pequeño, $\epsilon_1 \cdot \epsilon_2$ se puede despreciar frente $|\epsilon_1| + |\epsilon_2|$; por lo tanto,

$$|\frac{\hat{x}_1 \cdot \hat{x}_2 - x_1 \cdot x_2}{x_1 \cdot x_2}| \leq 2 \cdot eps$$

resulta, $\kappa = 2$, es decir, un problema bien condicionado.

Ejemplo 3.5 *Substracción de dos números reales. Dados $x_i, i = 1, 2$, considérese el problema de calcular $\mathcal{P}(x_1, x_2) = x_1 - x_2$.*

En forma análoga al anterior se obtiene:

$$|\frac{(\hat{x}_1 - \hat{x}_2) - (x_1 - x_2)}{x_1 - x_2}| = |\frac{x_1 \cdot \epsilon_1 - x_2 \cdot \epsilon_2}{x_1 - x_2}| \leq \frac{|x_1| + |x_2|}{|x_1 - x_2|} \cdot eps$$

En consecuencia, si sign $x_1 = -$ sign x_2, lo que corresponde a una suma, no a una resta, se tiene $\kappa = 1$, es decir, un problema bien condicionado.

Por el contrario, si $x_1 \approx x_2$, el número de condición resulta muy grande y el problema muy mal condicionado. Por ejemplo,

$$x_1 = \frac{1}{51}, \quad x_2 = \frac{1}{52} \Longrightarrow \kappa \approx \frac{2/50}{(1/50)^2} = 100$$

Si en esta situación, se realizan las operaciones con 3 cifras exactas (en base 10), se obtiene: $\hat{x}_1 = 0.196 \cdot 10^{-1}$, $\hat{x}_2 = 0.192 \cdot 10^{-1}$ y $\hat{x}_1 - \hat{x}_2 = 0.400 \cdot 10^{-3}$. Como las dos primeras cifras de $\hat{x}_1$ y $\hat{x}_2$ son las misma, la substracción las hace desaparecer y solo se conserva una cifra significativa. Se dice que ha habido *pérdida de cifras significativas*. Esto es lo que ha ocurrido en la resolución a) del problema de Forsythe (Ejemplo (3.2))

En consecuencia, se establece la siguiente

Estrategia de elección de pivote:

a) pivote parcial, se escoge como pivote el elemento de mayor valor absoluto de la columna involucrada en cada etapa.

b) pivote total, se escoge como pivote el elemento de mayor valor absoluto de entre todas las filas y columnas todavía no utilizadas en las etapas anteriores.

En ambos casos, el coeficiente l_{i1} resultante es $|l_{i1}| \leq 1$.

Ya que en el caso b) es preciso permutar filas y columnas, con la consiguiente pérdida de significado de las incógnitas, suele preferirse la estrategia de pivote parcial sobre la otra.

Algoritmo para la factorización LU.

Un algoritmo para la determinación de las matrices L y U de esta factorización puede ser conseguido siguiendo el *proceso de Doolitle-Banaciewicz*, que consiste en determinar en sucesivas etapas la primera fila de U, la primera columna de L, la segunda fila de U, la segunda columna de L, y así sucesivamente. Lo que se concreta en el algoritmo siguiente:

Algoritmo de Doolitle-Banaciewicz

$$\begin{aligned}
&\text{Para} \quad k = 1, 2, ..., n, \\
&\qquad \text{Para } j = k, ..., n, \\
&\qquad\qquad u_{kj} = a_{kj} - \sum_{p=1}^{k-1} l_{kp}\, u_{pj}; \\
&\qquad \text{Para } i = k+1, ..., n, \\
&\qquad\qquad l_{ik} = \frac{1}{u_{kk}} \left(a_{ik} - \sum_{i=1}^{k-1} l_{ip}\, u_{pk}\right).
\end{aligned} \tag{16}$$

Si se impone que los elementos $u_{ii} = 1, i = 1, 2, \ldots, n$, también la factorización $L\,U$ es única y el algoritmo análogo (que comenzaría con la primera columna de L, la primera fila de U, etc.) se denomina *algoritmo de Crôut*.

Este algoritmo es conveniente para ordenadores escalares, pero para ordenadores vectoriales sería mejor trabajar en la forma equivalente siguiente.

Algoritmo de Iserles

Denotando las columnas de L por $l_1, l_2, \ldots, l_n$ y las filas de U por $u_1^T, u_2^T, \ldots, u_n^T$, resulta

$$A = LU = \begin{bmatrix} l_1 & l_2 & \cdots & l_n \end{bmatrix} \begin{bmatrix} u_1^T \\ u_2^T \\ \vdots \\ u_n^T \end{bmatrix} = \sum_{k=1}^{n} l_k\, u_k^T. \tag{17}$$

Ya que las primeras $k-1$ componentes de l_k y u_k son todas cero, cada matriz $l_k\, u_k^T$ tiene ceros en sus primeras $k-1$ filas y columnas.

Como $l_k\, u_k^T$ no cambia si se reemplaza $l_k \to \alpha\, l_k$, $u_k \to \alpha^{-1}\, u_k$, donde $\alpha \neq 0$, la k-ésima fila de $l_k\, u_k^T$ es u_k^T y su k-ésima columna es $U_{k,k}$ veces l_k.

Empezando por $k = 1$, los primeros elementos de l_k y u_k son cero para $k \geq 2$, por lo tanto, se sigue que u_1^T debe ser la primera fila de A y l_1 debe ser la primera columna de A, dividida por $A_{1,1}$, para que $L_{1,1} = 1$.

Una vez halladas l_1 y u_1, se construye la matriz

$$A_1 = A - l_1\, u_1^T = \sum_{k=2}^{n} l_k\, u_k^T.$$

Como las primeras fila y columna de A_1 son cero, se deduce que u_2^T debe ser la segunda fila de $A - l_1\, u_1^T$, mientras que l_2 debe ser la segunda columna, escalada para que $L_{2,2} = 1$.

En resumen, se tiene así el siguiente

Algoritmo

$$\begin{aligned}
&\text{Poner } A_0 := A \\
&\text{Para } k = 1, 2, ..., n \\
&\quad u_k^T = \text{fila } k\text{-ésima de } A_{k-1} \\
&\quad l_k = \text{columna } k\text{-ésima de } A_{k-1}, \text{escalada para que } L_{k,k} = 1 \\
&\quad \text{Calcular } A_k := A_{k-1} - l_k\, u_k^T
\end{aligned} \tag{18}$$

Ejemplo 3.6 *(Ejercicio 1.5 de la colección). Hallar la factorización $L\,U$ de la matriz*

$$A = \begin{pmatrix} 1 & 2 & 3 \\ 3 & 2 & 4 \\ 2 & -1 & 1 \end{pmatrix}$$

Solución: De acuerdo con el algoritmo (18), para k = 1,

$$u_1^T = (1\ 2\ 3), \quad l_1^T = (1\ 3\ 2)^T,$$

$$A_1 = A - l_1\, u_1^T = \begin{pmatrix} 0 & 0 & 0 \\ 0 & -4 & -5 \\ 0 & -5 & -5 \end{pmatrix},$$

para k = 2,

$$u_2^T = (0\ -4\ -5), \quad l_2^T = (0\ -4\ -5)^T/(-4) = (0\ 1\ 5/4)^T,$$

$$A_2 = A_1 - l_2\, u_2^T = \begin{pmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 5/4 \end{pmatrix},$$

para k = 3,

$$u_3^T = (0\ 0\ 5/4), \quad l_3^T = (0\ 0\ 1)$$

En consecuencia,

$$L = \begin{pmatrix} 1 & 0 & 0 \\ 3 & 1 & 0 \\ 2 & 5/4 & 1 \end{pmatrix}, \qquad U = \begin{pmatrix} 1 & 2 & 3 \\ 0 & -4 & -5 \\ 0 & 0 & 5/4 \end{pmatrix}$$

Basta realizar el producto $L\,U$ para comprobar el resultado.

4 Factorización de matrices estructuradas.

En el caso particular de que la matriz A tenga algún tipo de estructura, por ejemplo, sea simétrica o definida positiva o hueca, los algoritmos a utilizar deberían aprovechar dicha estructura. Veamos, en primer lugar el siguiente resultado.

Teorema 4.1 *Si la matriz A es simétrica y definida positiva:*
a) La eliminación gaussiana es posible sin estrategia de pivote.
b) La descomposición $A = L\,U$ satisface

$$U = D\,L^T, \quad \textit{siendo } D = diag(u_{11}, \ldots, u_{nn}) \tag{19}$$

Demostr.: a) Ya que la matriz es definida positiva, $a_{11} > 0$, por lo que se puede elegir como pivote en la primera etapa de la eliminación. Con lo cual se obtiene

$$A = \begin{pmatrix} a_{11} & a^T \\ a & C \end{pmatrix} \quad \longrightarrow \quad A^{(1)} = \begin{pmatrix} a_{11} & a^T \\ 0 & C^{(1)} \end{pmatrix}, \tag{20}$$

siendo

$$C^{(1)} = C - \frac{1}{a_{11}} \cdot a \cdot a^T \tag{21}$$

La matriz $C^{(1)}$ es simétrica, evidentemente, y definida positiva. En efecto, tomemos $y \in \mathbb{R}^{n-1}, y \neq 0$, la partición (20) realizada y el que A es definida positiva implican

$$(x_1, y^T) \begin{pmatrix} a_{11} & a^T \\ a & C \end{pmatrix} \begin{pmatrix} x_1 \\ y \end{pmatrix} = a_{11}\, x_1^2 + 2x_1 \cdot y^T a + y^T C y > 0$$

Así, eligiendo $x_1 = -y^T a/a_{11}$, se obtiene de (21) que

$$y^T C^{(1)} y = y^T C y - \frac{1}{a_{11}} (y^T a)^2 > 0$$

Por recurrencia, se ve que todas las etapas de la eliminación son posibles sin elegir el pivote.

b) La expresión (19) es consecuencia de la unicidad de la factorización $L\,U$, pues es suficiente escribir $U = D\,\hat{L}$ para obtener

$$A = LU = L\,D\,\hat{L} = A^T = U^T\,L^T = \hat{L}^T\,(D\,L^T),$$

de donde, $\hat{L} = L^T$. ■

4.1 Factorización racional de Cholesky

Definición 4.2 *La descomposición*

$$A = L\,D\,L^T, \tag{22}$$

obtenida como consecuencia del teorema 4.1, se denomina descomposición o factorización racional de Cholesky.

Un algoritmo "escalar" análogo al de Doolitle-Banaciewicz se puede conseguir inmediatamente, aunque para ordenadores vectoriales sería mejor trabajar en la forma análoga a la (18) descrita más arriba; para ello, se escribe la factorización racional de Cholesky en la forma

$$A = \begin{bmatrix} l_1 & l_2 & \cdots l_n \end{bmatrix} \begin{bmatrix} D_{1,1} & 0 & \cdots & 0 \\ 0 & D_{2,2} & \ddots & \vdots \\ \vdots & \ddots & \ddots & 0 \\ 0 & \cdots & 0 & D_{n,n} \end{bmatrix} \begin{bmatrix} l_1^T \\ l_2^T \\ \vdots \\ l_n^T \end{bmatrix} = \sum_{k=1}^{n} D_{k,k}\, l_k\, l_k^T.$$

(recordar que l_k es la k-ésima columna de L).

La analogía con el algoritmo de la sección (18) es obvia poniendo $U = DL^T$, pero esta forma es mejor para aprovechar la simetría.

Particularizando, pues, se obtiene el algoritmo siguiente:

Algoritmo:

Poner $A_0 = A$
Para $k = 1, 2, ..., n$

$$\begin{aligned} &D_{k,k} = (A_{k-1})_{k,k} \\ &l_k = \text{columna } k\text{-ésima de } A_{k-1}, \text{escalada para que } L_{k,k} = 1 \\ &\text{Calcular } A_k := A_{k-1} - D_{k,k}\, l_k\, l_k^T \end{aligned} \tag{23}$$

Ejemplo 4.3 *Sea* $A = A_0 = \begin{bmatrix} 2 & 4 \\ 4 & 11 \end{bmatrix}$.

Aplicando el algoritmo,

$$l_1 = \begin{bmatrix} 1 \\ 2 \end{bmatrix}, \quad D_{1,1} = 2 \text{ y } A_1 = A_0 - D_{1,1} l_1 l_1^T = \begin{bmatrix} 2 & 4 \\ 4 & 11 \end{bmatrix} - 2 \begin{bmatrix} 1 & 2 \\ 2 & 4 \end{bmatrix} = \begin{bmatrix} 0 & 0 \\ 0 & 3 \end{bmatrix}.$$

Deducimos que $l_2 = \begin{bmatrix} 0 \\ 1 \end{bmatrix}$, $D_{2,2} = 3$ y, finalmente,

$$A = \begin{bmatrix} 1 & 0 \\ 2 & 1 \end{bmatrix} \begin{bmatrix} 2 & 0 \\ 0 & 3 \end{bmatrix} \begin{bmatrix} 1 & 2 \\ 0 & 1 \end{bmatrix}.$$

4.2 Método de Cholesky

Ya que A es definida positiva, $u_{ii} > 0$ y, por lo tanto, se puede considerar la raíz $D^{1/2} = diag(\sqrt{u_{11}}, \ldots, \sqrt{u_{nn}})$, con lo que la descomposición (22) se convierte en

$$A = (L\, D^{1/2})(D^{1/2}\, L^T) = (L\, D^{1/2})(L\, D^{1/2})^T.$$

Finalmente, volviendo a llamar L a $L\, D^{1/2}$, podemos escribir

$$A = L\, L^T, \text{ donde } L = \begin{pmatrix} l_{11} & & & \\ l_{21} & l_{22} & & \\ \vdots & \ddots & \ddots & \\ l_{n1} & \ldots & l_{n,n-1} & l_{nn} \end{pmatrix} \tag{24}$$

que se denomina *factorización de Cholesky.*

Observar que, por ser la matriz A definida positiva, su matriz diagonal reducida es la matriz unidad, en consecuencia, hubiera sido suficiente realizar la diagonalización de la matriz A mediante el método de las "congruencias" o el de Lagrange hasta llegar a una matriz diagonal coincidente con la identidad utilizando una matriz P triangular inferior regular tal que $P\, A\, P^T = I$. La matriz L anterior es, precisamente, P^{-1}.

Evidentemente, se tiene

$$\begin{array}{ll} i = k, & a_{kk} = l_{k1}^2 + \ldots + l_{kk}^2 \\ i > k, & a_{ik} = l_{i1}\, l_{k1} + \ldots + l_{ik}\, l_{kk} \end{array}$$

de donde se deduce el algoritmo siguiente

Algoritmo

$$\begin{array}{l} \text{Para } k = 1, 2, ..., n \\ \quad l_{kk} = (a_{kk} - \sum_{j=1}^{k-1} l_{kj}^2)^{1/2}; \\ \quad \text{para } i = k+1, ..., n, \\ \quad\quad l_{ik} = \frac{1}{l_{kk}} (a_{ik} - \sum_{j=1}^{k-1} l_{ij}\, l_{kj}). \end{array} \tag{25}$$

Evidentemente, con el algoritmo de Iserles (23), la localización de la matriz L se reduce a determinar sus columnas con la condición $l_{ii} = D_{ii}^{1/2}$.

El coste computacional de la factorización de Cholesky por cualquiera de los algoritmos presentados es del orden de $n^3/3$ operaciones aritméticas elementales, es decir, la mitad que la factorización LU.

Nótese que debido a que para matrices simétricas definidas positivas se tiene que

$$\forall i,j, \qquad a_{ij} \leq a_{ii}\, a_{jj}$$

lo que permitye probar que la factorización de Cholesky es un proceso estable.

4.3 Matrices huecas

Frecuentemente hay que resolver sistemas muy grandes ($n = 10^5$ es un ejemplo no grande), donde la mayoría de los elementos de A son ceros. Estas matrices son llamadas *matrices huecas*. La idea es que, al manipular matrices huecas, deberían resultas factores L y U también huecas. La única herramienta que permite reducir el número de ceros es la permutación de filas y columnas, como puede verse en el siguiente

Ejemplo 4.4

$$\begin{bmatrix} -3 & 1 & 1 & 2 & 0 \\ 1 & -3 & 0 & 0 & 1 \\ 1 & 0 & 2 & 0 & 0 \\ 2 & 0 & 0 & 3 & 0 \\ 0 & 1 & 0 & 0 & 3 \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 \\ -1/3 & 1 & 0 & 0 & 0 \\ -1/3 & -1/8 & 1 & 0 & 0 \\ -2/3 & -1/4 & 6/16 & 1 & 0 \\ 0 & -3/8 & 1/19 & 4/81 & 1 \end{bmatrix} \begin{bmatrix} -3 & 1 & 1 & 2 & 0 \\ 0 & -8/3 & 1/3 & 2/3 & 1 \\ 0 & 0 & 19/8 & 3/4 & 1 \\ 0 & 0 & 0 & 81/19 & 4/19 \\ 0 & 0 & 0 & 0 & 272/81 \end{bmatrix}$$

en donde se ve que la mayoría de ceros de A se han transformado en elementos no nulos. Sin embargo, reordenando simétricamente filas y columnas resulta

$$\begin{bmatrix} 2 & 0 & 1 & 0 & 0 \\ 0 & 3 & 2 & 0 & 0 \\ 1 & 2 & -3 & 0 & 1 \\ 0 & 0 & 0 & 3 & 1 \\ 0 & 0 & 1 & 1 & -3 \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 \\ 1/2 & 2/3 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & -6/29 & 1/3 & 1 \end{bmatrix} \begin{bmatrix} 2 & 0 & 1 & 0 & 0 \\ 0 & 3 & 2 & 0 & 0 \\ 0 & 0 & -29/6 & 0 & 1 \\ 0 & 0 & 0 & 3 & 1 \\ 0 & 0 & 0 & 0 & -272/87 \end{bmatrix}$$

Este es un ejemplo de matrices banda, en donde los elementos no nulos están en la diagonal o cerca de ella.

Teorema 4.5 *Sea la factorización $A = LU$ obtenida sin pivotaje de una matriz hueca. Entonces, todos los elementos nulos del comienzo de las filas de A son heredados por L y todos los ceros del comienzo de las columnas de A son heredados por U.*

4.4 Matrices tridiagonales

Para matrices tridiagonales (matrices banda, de ancho de banda = 3) conviene reconstruir el algoritmo de Doolitle (16) para eliminar operaciones innecesarias, por ejemplo, en la forma siguiente

$$\begin{bmatrix} a_1 & c_1 & & & \\ b_2 & a_2 & c_2 & & \\ & \ddots & \ddots & \ddots & \\ & & b_{n-1} & a_{n-1} & c_{n-1} \\ & & & b_n & a_n \end{bmatrix} = \begin{bmatrix} 1 & & & & \\ \beta_2 & 1 & & & \\ & \ddots & \ddots & & \\ & & \beta_n & 1 & \\ & & & \beta_{n-1} & 1 \quad 0 \end{bmatrix} \begin{bmatrix} \alpha_1 & \gamma_1 & & & \\ & \alpha_2 & \gamma_2 & & \\ & & \ddots & \ddots & \\ & & & \alpha_{n-1} & \gamma_{n-1} \\ & & & & \alpha_n \end{bmatrix}$$

lo que proporciona el siguiente

Algoritmo

$$\begin{array}{l}\alpha_1 = a_1, \quad \gamma_1 = c_1, \quad \beta_2 = b_2/\alpha_1 \\ \text{Para: } k = 2, 3, \ldots, n-1, \\ \qquad \alpha_k = a_k - \beta_k\,\gamma_{k-1}, \quad \gamma_k = c_k, \quad \beta_{k+1} = b_{k+1}/\alpha_k \\ \gamma_n = 0, \quad \alpha_n = a_n - \beta_n\,\gamma_{n-1}\end{array} \tag{26}$$

Obsérvese que esta factorización no siempre es posible; una condición suficiente para ello es que

$$\begin{array}{l}|a_1| > |c_1| > 0, \\ |a_k| \geq |b_k| + |c_k| \\ |a_n| > |b_n| > 0\end{array} \tag{27}$$

Coste computacional

El coste computacional de esta factorización es $3n-3$, ya que son precisas $n-1$ divisiones, $n-1$ multiplicaciones y $n-1$ sumas; para la resolución del sistema de ecuaciones habrá que añadir las operaciones requeridas en las dos sustituciones progresiva y regresiva, que suponen $5n-4$ operaciones más.

5 Sistemas sobredeterminados

Sea el sistema de m ecuaciones lineales con n incógnitas definido en forma matricial mediante:

$$A\,x = b,$$

en donde $A \in \mathbb{R}^{(m,n)}$ y $b \in \mathbb{R}^m$ son datos y $x \in \mathbb{R}^n$ es el vector incógnita; explícitamente se puede escribir en la forma:

$$\begin{array}{ccccccccc} a_{11}\,x_1 & + & a_{12}\,x_2 & + & \ldots & + & a_{1n}\,x_n & = & b_1 \\ a_{21}\,x_1 & + & a_{22}\,x_2 & + & \ldots & + & a_{2n}\,x_n & = & b_2 \\ \vdots & & \vdots & & \ldots & & \vdots & & \vdots \\ a_{m1}\,x_1 & + & a_{m2}\,x_2 & + & \ldots & + & a_{mn}\,x_n & = & b_m \end{array} \tag{28}$$

Evidentemente, el sistema (28) no posee, en general, solución. La idea es la de localizar un vector x tal que

$$||A\,x - b||_2 \qquad \text{sea mínima} \tag{29}$$

con la norma euclídea. Este proceso se denomina *método de los mínimos cuadrados*, cuyo nombre indica la elección de la norma en (29) (la suma de los cuadrados de los errores debe ser mínima).

Teorema 5.1 *Siendo A una matriz $m \times n$ (con $m \geq n$) y $b \in \mathbb{R}^m$, el vector x es solución de (29) si y solo si*

$$A^T\,A\,x = A^T\,b \tag{30}$$

Estas ecuaciones se denominan ecuaciones normales o de Gauss.

Demostr.: Los mínimos de la función cuadrática

$$f(x) := ||A\,x - b||_2^2 = (A\,x - b)^T\,(A\,x - b) = x^T\,A^T\,A\,x - 2x^T\,A^T\,b + b^T\,b$$

están dados por: $0 = f'(x) = 2(A^T\,A\,x - A^T\,b)$. ∎

Interpretación geométrica

El conjunto $W = \{A\,x \,|\, x \in \mathbf{R}^n\}$ es un subespacio lineal de $\mathbf{R}^m$. Para un $b \in \mathbf{R}^m$ dado, x es una solución del problema (29) si y solamente si $A\,x$ es la proyección ortogonal de b sobre W, lo que significa que $A\,x - b \perp A\,z$, para todo $z \in \mathbf{R}^n$; en consecuencia, $A^T\,(A\,x - b) = 0$, lo que proporciona una segunda demostración de (30).

Ejemplo 5.2 *La solubilidad el NO_3K respecto de la temperatura medida se representa en la tabla*

T	*40*	*60*	*80*	*100*	*120*
s	*27*	*39*	*50*	*60*	*69*

Encontrar una relación entre s y T.

Solución: Como $\Delta s_i = 12, 11, 10, 9$ y $\Delta^2 s_i = 1, 1, 1$, se sugiere la relación: $s = a + b\,t + c\,t^2$.

En forma matricial, el problema de mínimos cuadrados se puede plantear

$$A\,x = b \Longleftrightarrow \begin{pmatrix} 1 & 40 & 1600 \\ 1 & 60 & 3600 \\ 1 & 80 & 6400 \\ 1 & 100 & 10000 \\ 1 & 120 & 14400 \end{pmatrix} \begin{pmatrix} a \\ b \\ c \end{pmatrix} = \begin{pmatrix} 27 \\ 39 \\ 50 \\ 60 \\ 69 \end{pmatrix}$$

cuyas ecuaciones normales son:

$$A^T\,A\,x = A^T\,b \text{ con } A^T\,A = \begin{pmatrix} 5 & 400 & 36000 \\ 400 & 36000 & 3520000 \\ 36000 & 3520000 & 363840000 \end{pmatrix} \text{ y } \begin{pmatrix} a \\ b \\ c \end{pmatrix} = \begin{pmatrix} 245 \\ 21700 \\ 2097200 \end{pmatrix}$$

cuya solución por el método de Cholesky da

$$a = 0, \quad b = 0.725, \quad c = -0.00125$$

Obsérvese que si $s_5 = 69.1$, la solución resulta:

$$a = 0.14, \quad b = 0.7202, \quad c = -0.00121$$

y es que, debido al mal condicionamiento de la matriz $A^T\,A$, la soluciones son extremadamente sensibles a los errores en los datos.

Nótese que las ecuaciones normales (30) tienen siempre al menos una solución (la proyección de b sobre W siempre existe). La matriz $A^T\,A$ es semidefinida positiva, y resulta definida positiva si rang $A = n$, es decir, si las columnas de A son linealmente independientes. En estas condiciones, se suele aplicar el algoritmo de Cholesky para resolver el sistema (30); sin embargo, suele ser preferible calcular dicha solución directamente a partir de (29).

6 Factorización Q R

Como se ha visto (sección 3), la eliminación gaussiana consiste en multiplicar el sistema $A\,x = b$ por una matriz triangular inferior de forma que el problema se reduce a la resolución del problema equivalente $R\,x = c$, siendo R triangular superior. Desgraciadamente, la multiplicación de $A\,x - b$ por una tal matriz no conserva la norma del vector.

Para resolver (29), se busca una matriz ortogonal Q tal que

$$Q^T\,(A\,x-b)=R\,x-c=\left(\begin{array}{c}R'\\0\end{array}\right)x-\left(\begin{array}{c}c'\\c''\end{array}\right) \tag{31}$$

siendo R' una matriz cuadrada de orden n triangular superior y $(c',c'')^T$ la parte de $c=Q^T\,b$ tal que $c'\in\mathbb{R}^n$ y $c''\in\mathbb{R}^{m-n}$.

Como el producto por una matriz ortogonal no cambia la norma, ahora, se tiene

$$||A\,x-b||_2^2=||Q^T\,(A\,x-b)||_2^2=||R\,x-c||_2^2=||R\,x-c'||_2^2+||c''||_2^2 \tag{32}$$

Entonces, se obtiene la solución de (29) sin más que resolver el sistema

$$R\,x=c' \tag{33}$$

El problema se ha reducido a calcular una matriz ortogonal Q, es decir, $Q^T\,Q=I$, y una matriz triangular superior R tales que

$$A=Q\,R \tag{34}$$

Definición 6.1 *La descomposición (34) anterior se denomina factorización $Q\,R$*

Evidentemente, cualquier sistema compatible puede ser resuelto utilizando esta factorización. Basta calcular, primero, la factorización $Q\,R$ y, después, resolver el sistema triangular $R\,x=Q^T\,b$.

Una interpretación de la factorización $Q\,R$ es la siguiente: Sea $m\geq n$ y sean $a_1,a_2,...,a_n$ y $q_1,q_2,...,q_m$ las columnas de A y de Q, respectivamente; puesto que

$$[a_1\,a_2\,\ldots a_n]=[q_1\,q_2\,\ldots q_m]\left[\begin{array}{cccc}R_{11}&R_{12}&\ldots&R_{1n}\\0&R_{22}&\ldots&\vdots\\\vdots&\ddots&\ddots&\\0&\ldots&0&R_{nn}\\\vdots&&&\vdots\\0&\ldots&\ldots&0\end{array}\right],$$

se obtiene que

$$a_k=\sum_j^k R_{jk}\,q_j,\qquad k=1,2,...,n$$

En otras palabras, cada columna de A puede ser escrita como combinación lineal de las k primeras columnas de Q. Esto es lo que se obtenía al construir una familia ortogonal a partir de una familia libre por medio del método de Gram-Schmidt; sin embargo, este proceso es muy inestable (entraña realizar productos escalares), por lo que para obtener la factorización $Q\,R$ se suelen utilizar las rotaciones de Givens y las reflexiones de Householder.

6.1 Rotaciones de Givens.

Dada una matriz $A_0=A\in\mathbb{R}^{(m,n)}$, se busca una secuencia de matrices ortogonales $\Omega_1,\Omega_2,...,\Omega_k\in\mathbb{R}^{(m,m)}$ tales que la matriz $A_i=\Omega_i\,A_{i-1}$ tenga más elementos nulos debajo de la diagonal que A_{i-1}, para $i=1,2,...,k$, de forma que la matriz $R=A_k$ final es triangular superior; por lo tanto, se tiene:

$$\Omega_k\,\Omega_{k-1}\,...\,\Omega_1\,A=R$$

y, en consecuencia, $A=Q\,R$, siendo $Q=\Omega_1^T,\Omega_2^T,...,\Omega_k^T$ ortogonal y R triangular superior.

Definición 6.2 *Se denomina matriz de Givens, $\Omega^{[p,q]}$, a cualquier matriz ortogonal que coincide con la matriz unidad, salvo en cuatro elementos, que se construyen en la forma siguiente:*

$$\Omega^{[p,q]}_{pp} = \Omega^{[p,q]}_{qq} = \cos\theta, \quad \Omega^{[p,q]}_{pq} = \operatorname{sen}\theta, \quad \Omega^{[p,q]}_{qp} = -\operatorname{sen}\theta,$$

para algún $\theta \in [-\pi, \pi]$

Por ejemplo, si $m = 4$,

$$\Omega^{[2,4]} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & \cos\theta & 0 & \operatorname{sen}\theta \\ 0 & 0 & 1 & 0 \\ 0 & -\operatorname{sen}\theta & 0 & \cos\theta \end{bmatrix}, \quad \Omega^{[4,2]} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & \cos\theta & 0 & -\operatorname{sen}\theta \\ 0 & 0 & 1 & 0 \\ 0 & \operatorname{sen}\theta & 0 & \cos\theta \end{bmatrix}$$

Teorema 6.3 *Dada una matriz $A \in \mathbf{R}^{m\times n}$, para cada $1 \le p < q \le m$, $i \in \{p, q\}$ y $1 \le j \le n$, existe un $\theta \in [-\pi, \pi]$ tal que $(\Omega^{[p,q]}\, A)_{i\,j} = 0$. Además, todas las filas de A permanecen sin cambios, excepto la p-ésima y la q-ésima que resultan combinaciones lineales de las antiguas filas p-ésima y q-ésima*

Demostr.: Basta definir la matriz $\Omega^{[p,q]}$ adecuadamente. Sea $i = q$; si $A_{pj} = A_{qj} = 0$, cualquier valor de θ sirve; caso contrario, basta escoger:

$$\cos\theta = \frac{A_{pj}}{\sqrt{A^2_{pj} + A^2_{qj}}}, \quad \operatorname{sen}\theta = \frac{A_{qj}}{\sqrt{A^2_{pj} + A^2_{qj}}}$$

Sea $i = p$, basta escoger:

$$\cos\theta = \frac{A_{qj}}{\sqrt{A^2_{pj} + A^2_{qj}}}, \quad \operatorname{sen}\theta = -\frac{A_{pj}}{\sqrt{A^2_{pj} + A^2_{qj}}}. \qquad \blacksquare$$

Ejemplo 6.4 *Sea*

$$A = \begin{pmatrix} 1 & 1 & 1 \\ 1 & 1.0001 & -2 \\ 1 & 2 & 2 \end{pmatrix},$$

basta tomar

$$Q_1 = \Omega^{[2,1]}, \quad Q_2 = \Omega^{[3,1]}, \quad Q_3 = \Omega^{[3,2]},$$

para obtener la factorización $A = Q\,R$, siendo

$$Q = Q_1^T\, Q_2^T\, Q_3^T = \begin{pmatrix} 0.577 & -0.408 & 0.707 \\ 0.577 & -0.408 & -0.707 \\ 0.577 & 0.816 & 0.0000707 \end{pmatrix}, \quad R = \begin{pmatrix} 1.732 & 2.309 & 0.577 \\ 0 & 0.816 & 2.041 \\ 0 & 0 & 2.121 \end{pmatrix},$$

Obsérvese que en muchas ocasiones, por ejemplo, al resolver un sistema de ecuaciones, lo que se necesita es el resultado de multiplicar $Q\,b$, no la matriz Q; por ello, se suele obtener el resultado multiplicando el vector b por las matrices Q_j sucesivas.

Coste computacional

Como para la factorización $Q\,R$ se necesitan menos de $m\,n$ rotaciones y cada una de ellas reeemplaza dos filas por sus combinaciones lineales, resulta un coste total del orden $\mathcal{O}(m\,n^2)$.

6.2 Reflexiones de Householder

Una matriz de la forma

$$H = I - 2u\,u^T \qquad \text{donde } u^T\,u = 1 \tag{35}$$

tiene las siguientes propiedades:

- H define una reflexión en el hiperplano $\{x|u^T\,x = 0\}$, ya que $H\,x = x - 2u\cdot(2u^T\,x)$ y $H\,x + x \perp u$
- H es simétrica
- H es ortogonal, ya que $H^T\,H = (I - 2u\,u^T)^T(I - 2u\,u^T) = I - 4u\,u^T + 4u\,u^T\,u\,u^T = I$.

Mediante el siguiente algoritmo se puede conseguir transformar la matriz A en otra triangular superior haciendo ceros en todos los elementos por debajo de la diagonal de cada columna mediante premultiplicación por una matriz de Householder.

Algoritmo

En una primera etapa, se busca una matriz $H_1 = I - 2u_1\,u_1^T \quad (u_1 \in \mathbf{R}^m$, y $u_1^T\,u_1 = 1)$ tal que

$$H_1\,A = \begin{pmatrix} \alpha_1 & * & \dots & * \\ 0 & * & \dots & * \\ \vdots & \vdots & & \vdots \\ 0 & * & \dots & * \end{pmatrix} \tag{36}$$

Si A^1 es la primera columna de A, se debe cumplir que $H_1\,A^1 = \alpha_1\,e_1 = (\alpha, 0, ..., 0)^T$, luego: $|\alpha_1| = ||H_1\,A^1||_2 = ||A^1||_2$. Por definición de H_1, debe ser

$$H_1\,A^1 = A^1 - 2u_1\,u_1^T\,A^1 = \alpha_1\,e_1$$

Como la expresión $u_1^T\,A^1$ es un escalar, resulta que

$$u_1 = C\,v_1, \quad \text{donde } v_1 = A^1 - \alpha_1\,e_1 \tag{37}$$

y la constante C se determina de forma que $||u_1||_2 = 1$. Como todavía queda la libertad de elegir el signo de α_1, este se escoge en la forma

$$\alpha_1 = -(\text{sign } a_{11})\,||A^1||_2 \tag{38}$$

para evitar una sustracción mal condicionada en el cálculo de $v_1 = A^1 - \alpha_1\,e_1$

Para *calcular* la matriz $H_1\,A$ téngase en cuenta que

$$H_1\,A = A - 2u_1\,u_1^T\,A = A - \beta\,v_1\,(v_1^T\,A), \quad \text{donde } \beta = \frac{2}{v_1^T\,v_1} \tag{39}$$

Este factor β puede ser calculado en la forma siguiente

$$\beta^{-1} = \frac{v_1^T\,v_1}{2} = \frac{1}{2}\,(A_1^T\,A_1 - 2\alpha\,a_{11} + \alpha_1^2) = -\alpha_1\,(a_{11} - \alpha_1). \tag{40}$$

En la segunda etapa, se aplica el mismo procedimiento a la submatriz de (36) de tipo $(m-1)\times(n-1)$; lo cual proporciona un vector $\bar{u}_2 \in \mathbf{R}^{m-1}$ y una matriz de Householder $\bar{H}_2 = I - 2\,\bar{u}_2\,\bar{u}_2^T$.

Escribiendo $u_2 = (0, \bar{u}_2)^T$ y multiplicando (36) por $H_2 = I - 2\,u_2\,u_2^T$, se obtiene

$$H_2\,H_1\,A = H_2 \begin{pmatrix} \alpha_1 & * & \dots & * \\ 0 & & & \\ \vdots & & C & \\ 0 & & & \end{pmatrix} = \begin{pmatrix} \alpha_1 & * & \dots & * \\ 0 & & & \\ \vdots & & \bar{H}_2\,C & \\ 0 & & & \end{pmatrix} = \begin{pmatrix} \alpha_1 & * & * & \dots & * \\ 0 & \alpha_2 & * & \dots & * \\ 0 & 0 & * & \vdots & * \\ \vdots & \vdots & \vdots & & \vdots \\ 0 & 0 & * & \dots & * \end{pmatrix}$$

Este proceso se debe continuar n etapas ($n-1$, si $m = n$) para obtener la matriz triangular

$$H_n \dots H_2\,H_1\,A = R = \begin{pmatrix} R' \\ 0 \end{pmatrix}$$

y la matriz ortogonal $Q^T = H_n \dots H_2\,H_1$.

Nótese que a la hora de construir un programa para este algoritmo, no es necesario calcular explicitamente las matrices H_i, ni la matriz Q, es suficiente retener, además de R, los valores de β_i (o α_i) y los vectores v_i, los cuales contienen toda la información necesaria.

Coste computacional

La primera etapa exige el cálculo de α_1 mediante la fórmula (38), lo que significan $\approx m$ operaciones, el cálculo de $2/v_1^t\,v_1$ mediante (40) requiere 3 operaciones y el cálculo de $H_1\,A$ por la fórmula (39) necesita $\approx (n-1)\,2\,m$ operaciones. Esta etapa requiere, pues, unas $2\,m\,n$ operaciones. Para la factorización $Q\,R$ completase necesitarán:

si $m = n$, $2\,(n^2 + (n-1)^2 + \dots + 1) \approx 2\,n^3/3$ operaciones

si $m >> n$, $2\,m\,(n + (n-1) + \dots + 1) \approx m\,n^2$ operaciones

Puede verse que el coste computacional de la factorización $Q\,R$ es aproximadamente el doble que la factorización $L\,U$.

Si las columnas de A son casi linealmente dependientes, la resolución del sistema (29) mediante factorización $Q\,R$ es preferible a la utilización de las ecuaciones normales.

Ejemplo 6.5 *Sea el sistema definido por*

$$A = \begin{pmatrix} 1 & 1 \\ \epsilon & 0 \\ 0 & \epsilon \end{pmatrix}, \qquad b = \begin{pmatrix} 1 \\ 0 \\ 0 \end{pmatrix}$$

donde ϵ es una pequeña constante (por ejemplo, $\epsilon^2 < eps$).

Calculando de forma exacta

$$A^T\,A = \begin{pmatrix} 1+\epsilon^2 & 1 \\ 1 & 1+\epsilon^2 \end{pmatrix}, \quad A^T\,b = \begin{pmatrix} 1 \\ 1 \end{pmatrix}$$

y la solución resulta

$$x_1 = x_2 = \frac{1}{2+\epsilon^2} = \frac{1}{2} + \mathcal{O}(\epsilon^2).$$

Trabajando en coma flotante, el ϵ^2 desaparece, con lo que el problema se convierte en uno singular que no tiene solución.

Aplicando el algoritmo de Householder (despreciando ϵ^2), se obtiene: $\alpha_1 = -1$, $v_1 = (2, \epsilon, 0)^T$, ... y, finalmente,

$$R = \begin{pmatrix} -1 & -1 \\ 0 & \sqrt{2}\,\epsilon \\ 0 & 0 \end{pmatrix}, \qquad Q^T\, b = \begin{pmatrix} -1 \\ \epsilon/\sqrt{2} \\ -\epsilon/\sqrt{2} \end{pmatrix};$$

la resolución del sistema (33) da una buena aproximación de la solución.

Veámoslo resuelto con Matlab en doble precisión, para $\epsilon = 0.001$

```
alf=-sqrt(1+0.000001)
  -1.00000049999988

v=a(1:3,1)-alf*[1;0;0]
   2.00000049999988
   0.00100000000000
                  0

bet=1/alf/(alf-a(1,1))
   0.49999962500031

hab= ab - bet*v*(v'*ab)
 -1.00000049999988  -0.99999950000038  -0.99999950000038
                 0  -0.00099999950000  -0.00099999950000
                 0   0.00100000000000                   0

alf=+sqrt(0.0009999995*0.0009999995+0.001*0.001)
   0.00141421320882

v=[0;hab(2:3,2)]-alf*[0;1;0]
                 0
  -0.00241421270882
   0.00100000000000

bet=1/alf/(alf-hab(2,2))
     2.928933955901685e+05

hab= hab - bet*v*(v'*hab)
  -1.00000049999988  -0.99999950000038  -0.99999950000038
                 0   0.00141421320882   0.00070710625086
                 0  -0.00000000000000  -0.00070710660441

x2=hab(2,3)/hab(2,2)
   0.49999975000013

x1= (hab(1,3)-hab(1,2)*x2)/hab(1,1)
   0.49999975000012
```

7 Condicionamiento y refinamiento iterativo

Sabemos que el condicionamiento influye en la calidad de la solución de un problema cualquiera. En particular, en el problema de hallar la solución de un sistema lineal nos encontramos con que al comparar el valor exacto del término independiente de un sistema con el calculado puede haber discrepancias. En concreto, definiendo el vector residual $\boldsymbol{r}$ en la forma

$$\boldsymbol{r} = \tilde{b} - b,$$

en donde $\tilde{b}$ es el valor calculado, resulta:

Teorema 7.1 *Si A es una matriz regular, se verifica:*
1) $\|\tilde{x} - x\| \leq \|\boldsymbol{r}\| \, \|A^{-1}\|$

2) $\dfrac{\|\tilde{x} - x\|}{\|x\|} \leq \ A\| \, \|A^{-1}\| \, \dfrac{\|\boldsymbol{r}\|}{\|b\|}$

Demostr.: Como $\boldsymbol{r} = \tilde{b} - b = A\,\tilde{x} - A\,x = A\,(\tilde{x} - x)$, resulta

$$\|\tilde{x} - x\| = \|A^{-1}\,\boldsymbol{r}\| \leq \|\boldsymbol{r}\| \, \|A^{-1}\|$$

Por otro lado, como $A\,x = b$, $\|x\| \geq \dfrac{\|b\|}{\|A\|}$, luego de la desigualdad anterior resulta

$$\frac{\|\tilde{x} - x\|}{\|x\|} \leq \frac{\|\boldsymbol{r}\| \, \|A^{-1}\|}{\|x\|} \leq \frac{\|A^{-1}\,\|A\|\,\|\boldsymbol{r}\|}{\|b\|}$$

De acuerdo con la definición 3.3, podemos dar la siguiente

Definición 7.2 *Se denomina número de condición de una matriz al número*

$$\kappa(A) = \|A\| \, \|A^{-1}\|$$

Propiedad 7.3 *1)* $\kappa(A) \geq 1$
2) $\kappa(\lambda\,A) = \kappa(A)$
3) $\kappa(A^{-1}) = \kappa(A)$
4) Si Q es ortogonal, $\kappa(Q) = 1$

Si $\kappa(A)$ es pequeño, se dice que la matriz A está bien condicionada, si es grande que A está mal condiconada.

Como para encontrar el número de condición de una matriz hace falta conocer la matriz inversa, tarea bastante costosa, se puede buscar una **acotación** o una estimación del mismo. Lo primero se consigue, por ejemplo, tomando varios vectores y, calculando el valor $d = A\,y$ y acotando inferiormente la norma $\|A^{-1}\|$ mediante

$$\frac{\|y\|}{\|d\|} \leq \|A^{-1}\|.$$

Las cotas obtenidas de esta forma suelen ser muy bajas, por ello conviene encontrar una estimación mejor.

Para encontrar una **estimación** del número de condición, se resuelve el sistema $A\,x = b$ por eliminación gaussiana utilizando aritmética de p dígitos, obteniendo la solución $\tilde{x}$; a continuación, se resuelve también con p dígitos el sistema $A\,y = \boldsymbol{r}$ (que es inmediato, ya que se tienen en memoria

las operaciones de la eliminación anterior) obteniendo el vector $\tilde{y}$; finalmente, se usa la desigualdad de Forsythe y Moler siguiente

$$\|\boldsymbol{r}\| \approx 10^{-p}\,\|A\|\,\|\tilde{x}\|$$

para realizar la siguiente estimación:

$$\|\tilde{y}\| \approx \|A^{-1}\,\boldsymbol{r}\| \approx \|A^{-1}\|\,10^{-p}\,\|A\|\,\|\tilde{x}\|,$$

lo que permite estimar el número de condición por la siguiente

$$\kappa(A) \approx \frac{\|\tilde{y}\|}{\|\tilde{x}\|}\,10^{p}$$

Este procedimiento proporciona buenas estimaciones del número de condición (cf. Burden y Faires).

En el siguiente teorema veremos la relación entre las perturbaciones de los elementos del sistema y el número de condición de la matriz de coeficientes.

Suponiendo que la matriz A del sistema

$$A\,x = b$$

es regular, el sistema perturbado se puede escribir en la forma

$$(A + \Delta A)\,(x + \Delta x) = b + \Delta b \tag{41}$$

Propiedad 7.4 *Si*

$$\|\Delta A\|\,\|A^{-1}\| < 1,$$

la matriz de coeficientes $A + \Delta A$ del sistema perturbado es regular

Demostr.:

$$A + \Delta A = A\,(I + A^{-1}\,\Delta A)$$

Como A es regular, existe A^{-1}. La matriz $I + A^{-1}\,\Delta A$, como sabemos, será regular si $\|A^{-1}\,\Delta A\| < 1$, para lo cual es suficiente que $\|\Delta A\|\,\|A^{-1}\| < 1$. ■

Teorema 7.5 *En la situación mencionada, se verifica la siguiente acotación:*

$$\frac{\|\Delta x\|}{\|x\|} \leq \frac{K(A)}{1 - K(A)\dfrac{\|\Delta A\|}{\|A\|}}\left(\frac{\|\Delta A\|}{\|A\|} + \frac{\|\Delta b\|}{\|b\|}\right)$$

Demostr.: De la ecuación (41) se obtiene:

$$x + \Delta x = (A + \Delta A)^{-1}\,(b + \Delta b) = (A + \Delta A)^{-1}\,A\,x + (A + \Delta A)^{-1}\,\Delta b$$

Luego:

$$\Delta x = -(A + \Delta A)^{-1}\,(A + \Delta A)\,x + (A + \Delta A)^{-1}\,A\,x + (A + \Delta A)^{-1}\,\Delta b = (A + \Delta A)^{-1}\,(\Delta b - \Delta A\,x)$$

Tomando normas y aplicando un resultado conocido:

$$\|\Delta x\| \leq \|A^{-1}\|\,\|(I + A^{-1}\,\Delta A)^{-1}\|\,\|\Delta b - \Delta A\,x\| \leq \|A^{-1}\|\,\frac{1}{1 - \|A^{-1}\|\,\|\Delta A\|}\,(\|\Delta b\| + \ |\Delta A\|\,\|x\|)$$

Y, finalmente, como x es solución del problema no perturbado, es decir, $\|A\|\,\|x\| \geq \|b\|$, resulta el enunciado . ■

Corolario 7.6 *Si* $\|\Delta A\| = 0$*, entonces*

$$\frac{\|\Delta x\|}{\|x\|} \leq K(A)\,\frac{\|\Delta b\|}{\|b\|}$$

Corolario 7.7 *Si* $\dfrac{\|\Delta A\|}{\|A\|} \leq \delta$ *y* $\dfrac{\|\Delta b\|}{\|b\|} \leq \delta$*, entonces*

$$\frac{\|\Delta x\|}{\|x\|} \leq 2\,\delta\,\frac{K(A)}{1-\delta\,K(A)} = 2\,\delta\,K(A)\,\frac{1}{1-\lambda}, \quad \textit{si } \lambda = \delta\,\|\Delta A\|\,\|A^{-1}\|$$

En consecuencia, errores pequeños en b pueden producir resultados nefastos en la solución x.

7.0.1 Refinamiento iterativo o corrección residual

En el caso de que la matriz de coeficientes esté mal condicionada se puede utilizar el siguiente algoritmo de refinamiento iterativo o corrección residual para mejorar el resultado. Téngase cuidado porque si el problema no está mal condiconado este procedimiento encarece el coste computacional y no mejora sensiblemente la solución.

Algoritmo.

- Encontrar la solución $\tilde{x}$ del sistema $A\,x = b$
- Encontrar el vector residual $\boldsymbol{r} = A\,\tilde{x} - b$ con doble precisón
- Encontrar la solución $\tilde{e}$ de la ecuación $A\,e = \boldsymbol{r}$
- Construir la nueva aproximación de la solución $x = \tilde{x} + \tilde{e}$