



FACULTADE DE MATEMÁTICAS

Traballo Fin de Grao

# Estrategias de pivoteo para la resolución de sistemas de ecuaciones lineales

Jorge Riesco Dávila

2018/2019

UNIVERSIDADE DE SANTIAGO DE COMPOSTELA



GRAO DE MATEMÁTICAS

Traballo Fin de Grao

# Estrategias de pivoteo para la resolución de sistemas de ecuaciones lineales

Jorge Riesco Dávila

Julio de 2019

UNIVERSIDADE DE SANTIAGO DE COMPOSTELA



# Trabajo propuesto

<b>Área de Coñecemento: Matemática Aplicada</b>
<b>Título: Estrategias de pivoteo para la resolución de sistemas de ecuaciones lineales</b>
<b>Breve descripción do contido</b>
Se trata de elaborar una biblioteca documentada de programas en Fortran 90 para la resolución con métodos de factorización de sistemas de ecuaciones lineales y sus aplicaciones. La memoria incluirá la descripción de los métodos de factorización LU con estrategias de pivote parcial y total, factorizaciones de Crout y Cholesky con elección de pivote diagonal y factorización $AP=QR$ , y, entre las aplicaciones, la determinación del rango de una matriz, el cálculo de determinantes y de inversas de matrices cuadradas y la resolución de sistemas sobre y subdeterminados.
<b>Recomendacións</b>
Buen nivel de programación en Fortran 90. Haber superado las asignatura de Análisis Numérico Matricial y Métodos Numéricos en Optimización y Ecuaciones Diferenciales.
<b>Outras observacións</b>



# Índice general

<b>Resumen</b>	<b>VII</b>
<b>Introducción</b>	<b>IX</b>
<b>1. Método de Gauss y factorizaciones <math>LU</math></b>	<b>1</b>
1.1. Interpretación matricial el método de Gauss . . . . .	1
1.2. Estrategia de pivote parcial . . . . .	5
1.3. Estrategia de pivote total . . . . .	10
1.3.1. Código Fortran 90 . . . . .	14
<b>2. Factorizaciones para matrices hermitianas</b>	<b>17</b>
2.1. Factorización de Crout . . . . .	17
2.2. Factorización de Cholesky . . . . .	19
2.3. Factorización de Cholesky con estrategia de pivote . . . . .	21
2.3.1. Código Fortran 90 . . . . .	25
<b>3. Factorización QR</b>	<b>27</b>
3.1. Factorización QR . . . . .	27
3.2. Factorización QR con pivote . . . . .	31
3.2.1. Código Fortran 90 . . . . .	34
<b>4. Aplicaciones</b>	<b>37</b>
4.1. Cálculo de determinantes . . . . .	37
4.2. Cálculo del rango . . . . .	37
4.3. Cálculo de inversas . . . . .	38
<b>Anexo: Código para la resolución de los sistemas factorizados</b>	<b>39</b>
<b>Bibliografía</b>	<b>45</b>



## Resumen

En esta memoria se describen los métodos directos de factorización de matrices basados en estrategias de pivoteo que se aplican a la resolución de sistemas de ecuaciones lineales. En concreto se verá la factorización  $LU$ , derivada de la eliminación de Gauss, con pivote parcial y total, la factorización de Cholesky con pivote y la factorización  $QR$  con matrices de Householder y pivoteo de columnas. Cuando los métodos numéricos se implementan en el ordenador pueden aparecer problemas de estabilidad numérica en las versiones básicas, que los hacen inadecuados para resolver sistemas mal condicionados o con perturbaciones en los datos. Por ello es necesario introducir estrategias de pivoteo que eliminan o reducen, en la mayoría de los casos, esos comportamientos indeseables. Una vez calculada la factorización se puede resolver con mucho menor coste un gran número de problemas del álgebra lineal numérica. Se incluyen las aplicaciones al cálculo de inversas, determinantes y rango de una matriz.

## Abstract

This work describes the direct methods of pivoting based matrix decompositions in order to solve linear systems. Specifically, the  $LU$  decomposition will be derived from the Gaussian elimination, and partial and total pivoting will be introduced. For the Cholesky decomposition, the symmetric pivoting will be implemented and for Householder  $QR$  decomposition, the column pivoting will be carried out. The use of these strategies is necessary in computer calculations because the basic direct methods exhibit numerical instabilities in the ill-conditioned case. These problems are avoided in most cases by pivoting. Knowing a matrix decomposition, it will be very easy to obtain the rank, the determinant and the inverse.



# Introducción

El objetivo de esta memoria es describir los principales métodos directos de resolución de sistemas de ecuaciones lineales, tanto para matrices hermitianas como para el caso general, mediante factorizaciones en matrices triangulares y/o unitarias. Algunos de estos métodos ya fueron introducidos en la asignatura de segundo curso de Análisis Numérico Matricial ( $LU$  sin pivote y con pivote parcial, factorizaciones de Cholesky y  $QR$ ). Aquí veremos algunas modificaciones de dichos métodos extendiendo las estrategias de pivote a Cholesky y  $QR$  y pivote total para  $LU$ . Este tipo de técnicas son cruciales a la hora de resolver sistemas computacionalmente, pues reducen los errores de redondeo debidos a la precisión finita en la que trabaja el ordenador. Para sistemas de gran tamaño y con una distribución especial de ceros las estrategias de pivote deben ser restringidas con técnicas que no son el objeto de esta memoria.

Como señalan todas las referencias de análisis numérico que tratan el tema, hay métodos que presentan problemas de inestabilidad numérica, que pueden provocar grandes errores incluso en sistemas bien condicionados. El resultado puede ser catastrófico debido a que las operaciones realizadas en aritmética en punto flotante dan lugar a importantes pérdidas de dígitos significativos. Una descripción y análisis muy claro y detallado de los problemas de estabilidad numérica, redondeo y condicionamiento se realiza en [6] .

Aunque los métodos descritos en esta memoria están implementados en la mayoría de librerías y programas destinados al cálculo numérico tales como LAPACK y MATLAB, el desconocimiento de dichos métodos dejaría al usuario limitado a los algoritmos ya implementados. Por otra parte es importante para el usuario poder tomar una decisión informada sobre que algoritmo usar para resolver un determinado problema, o decidir bajo que condiciones el algoritmo se bloquea (cuando el pivote es nulo por debajo de un umbral), ya que de lo contrario podría llevar a una mala interpretación de los resultados. También es importante tener la capacidad de introducir modificaciones para que el usuario pueda adaptar el código a sus necesidades.

Además del trabajo de programación, se ha tratado de dar una formalización rigurosa de las técnicas matriciales interpretando cada una de las etapas.

En el primer capítulo comenzaremos recordando la factorización  $LU$ , su interpretación matricial y el teorema de existencia y unicidad. La siguiente sección se inicia con un ejemplo en el que la factorización  $LU$  no funciona adecuadamente, lo cual nos llevará a introducir la estrategia de pivote parcial, de la que se dará una interpretación matricial así como un teorema de existencia. Finalmente veremos que la estrategia de pivote parcial se puede mejorar para obtener un método numéricamente más estable, la factorización con pivote total, que es el método recomendado para matrices que con pivote parcial presentan un crecimiento exponencial de los coeficientes (vease [2]). De esta última también daremos una interpretación matricial y el teorema de existencia. Este método es el recomendado para calcular el rango de una matriz general.

En el segundo capítulo veremos que hay métodos más eficientes desde el punto de vista del almacenamiento y el tiempo de cálculo si la matriz posee unas determinadas características, más concretamente si es hermitiana y definida positiva. Comenzaremos introduciendo la factorización de Crout y su interpretación matricial, para derivar a continuación la factorización de Cholesky, de la que recordaremos las condiciones necesarias para su existencia y unicidad, y su interpretación matricial. Finalmente se verá que para una matriz hermitiana y definida positiva se puede aplicar una estrategia de búsqueda de pivote en la diagonal, y realizar una permutación simétrica de filas y columnas de la matriz para obtener la factorización de Cholesky con pivote. Terminaremos dando su interpretación matricial, y una manera de implementarla.

En el tercer capítulo se recuerda la factorización  $QR$  para matrices rectangulares usando el método de Householder, por lo que primero se introducen las matrices de Householder y veremos algunas de sus propiedades. A continuación se da la interpretación matricial de la factorización y un teorema que nos garantizará su existencia. Debido a la naturaleza de las transformaciones que reducen la matriz a forma triangular, se realizará una estrategia de pivoteo por columnas. Se enunciará el resultado teórico correspondiente y se dará su interpretación matricial, y una posible implementación.

Para concluir, en el último capítulo veremos algunas de las aplicaciones de los métodos descritos, como son el cálculo del determinante, el rango o la inversa de una matriz. Además se incluye en un anexo las rutinas de resolución de los sistemas resultantes de las factorizaciones.

Finalmente daremos algunas conclusiones y posibles continuaciones de este trabajo.

# Capítulo 1

## Método de Gauss y factorizaciones

### *LU*

En este capítulo se describen los métodos derivados de la eliminación de Gauss y las correspondientes estrategias de pivoteo. La interpretación matricial del método de eliminación conduce, bajo ciertas hipótesis sobre la matriz de coeficientes, a la factorización  $LU$  que permite resolver sistemas lineales fácilmente resolviendo dos sistemas triangulares. En la sección 1.1 se recuerda la eliminación de Gauss, se dará la interpretación matricial y se enunciará el teorema de existencia y unicidad de la factorización  $LU$ . En la sección 1.2 se introducirá la estrategia de pivote parcial y la correspondiente factorización  $PA = LU$  y en la sección 1.3 la estrategia de pivote total y la factorización  $P_\sigma AP_\tau^T = LU$ . Las estrategias descritas en este capítulo se pueden aplicar sin modificación tanto a matrices con coeficientes reales como a matrices con coeficientes complejos.

### 1.1. Interpretación matricial el método de Gauss

La eliminación de Gauss es un método de reducción sistemática que permite transformar un sistema de ecuaciones lineales en otro equivalente con matriz de coeficientes triangular. En la etapa  $k$ -ésima de la eliminación se utiliza la fila  $k$ -ésima de la matriz, si el coeficiente  $a_{kk}$  es no nulo, para reducir a cero los coeficientes de la columna  $k$ -ésima de las filas  $k + 1, k + 2, \dots, n$ . Procediendo en el orden natural, primero la columna uno, luego la dos y así sucesivamente hasta la  $(n - 1)$ -ésima, se consigue, al final del cálculo, un sistema con matriz de coeficientes triangular superior equivalente al original y más sencillo de resolver.

En esta sección recordaremos la interpretación matricial del método de Gauss, lo que nos llevará a la factorización  $LU$  de una matriz, donde la matriz  $U$  es triangular superior

resultante de la eliminación de Gauss, y la  $L$  es triangular inferior con unos en la diagonal. La reducción de Gauss se realiza mediante  $n - 1$  etapas de eliminación siendo  $n$  el orden de la matriz del sistema. En la etapa  $k$ -ésima de la eliminación se pasa de una matriz  $A^{(k-1)}$  con coeficientes nulos por debajo de la diagonal en las  $k - 1$  primeras columnas a  $A^{(k)}$  con coeficientes nulos por debajo de la diagonal también en la columna  $k$ -ésima. Esto se consigue realizando las siguientes operaciones de fila: Si  $a_{kk}^{(k-1)} \neq 0$ ,

$$\mathbf{F}_i \rightarrow \mathbf{F}_i - \frac{a_{ik}^{(k-1)}}{a_{kk}^{(k-1)}} \mathbf{F}_k \quad i = k + 1, k + 2, \dots, n.$$

Denotando por

$$l_{ik} = \frac{a_{ik}^{(k-1)}}{a_{kk}^{(k-1)}}, \quad i = k + 1, \dots, n, \quad (1.1)$$

se tiene  $\hat{L}_k A^{(k-1)} = A^{(k)}$  siendo

$$\hat{L}_k = \left( \begin{array}{cccc|ccc} 1 & 0 & & & 0 & 0 & \dots & 0 \\ 0 & 1 & 0 & & 0 & 0 & \dots & 0 \\ & & \ddots & \ddots & \vdots & \vdots & & \vdots \\ & & & 0 & 1 & 0 & \dots & 0 \\ \hline 0 & \dots & 0 & 0 & 1 & 0 & \dots & 0 \\ \hline 0 & \dots & 0 & 0 & -l_{k+1k} & 1 & \dots & 0 \\ \vdots & & \vdots & \vdots & \vdots & \vdots & \ddots & \vdots \\ 0 & \dots & 0 & 0 & -l_{nk} & 0 & \dots & 1 \end{array} \right) \leftarrow k \quad (1.2)$$

↑  
 $k$

Lo que muestra que la  $k$ -ésima etapa de eliminación es equivalente a multiplicar la matriz resultante de la etapa  $(k - 1)$ -ésima por  $\hat{L}_k$ . En la siguiente proposición se verán algunas propiedades de las matrices  $\hat{L}_k$  que se utilizarán más adelante.

**Proposición 1.1.** *Sea  $L_k$  una matriz triangular inferior con diagonal de unos y todos los coeficientes extradiagonales nulos salvo, eventualmente, en la columna  $k$ , filas  $k + 1, k + 2, \dots, n$ . Se verifica:*

1.  $\det(L_k) = 1$ .

2.  $(L_k^{-1})_{ij} = \begin{cases} 1 & \text{si } i = j \\ -l_{ij} & \text{si } i \neq j \end{cases}$ , donde  $l_{ij}$  es el coeficiente  $(i, j)$ -ésimo de la matriz  $L_k$ .

3. Si  $k < j$ , entonces

$$(L_k L_j)_{rs} = \begin{cases} 1 & \text{si } r = s \\ \alpha_{rk} & \text{si } s = k \\ \beta_{rj} & \text{si } s = j \\ 0 & \text{en otro caso,} \end{cases}$$

denotando con  $\alpha$  los coeficientes extradiagonales de  $L_k$  y con  $\beta$  los de  $L_j$ .

*Demostración.* 1. Basta con desarrollar el determinante por filas.

2. Multiplicando por bloques

$$\left( \begin{array}{c|c|c} I_{k-1} & 0 & 0 \\ \hline 0 & 1 & 0 \\ \hline 0 & -l_{k+1k} & \\ \vdots & \vdots & I_{n-k} \\ \hline 0 & -l_{nk} & \end{array} \right) \left( \begin{array}{c|c|c} I_{k-1} & 0 & 0 \\ \hline 0 & 1 & 0 \\ \hline 0 & l_{k+1k} & \\ \vdots & \vdots & I_{n-k} \\ \hline 0 & l_{nk} & \end{array} \right) = \left( \begin{array}{c|c|c} I_{k-1} & 0 & 0 \\ \hline 0 & 1 & 0 \\ \hline 0 & l_{k+1k} - l_{k+1k} & \\ \vdots & \vdots & I_{n-k} \\ \hline 0 & l_{nk} - l_{nk} & \end{array} \right).$$

3. Multiplicando de nuevo por bloques

$$\left( \begin{array}{c|c|c|c|c} I_{k-1} & 0 & 0 & 0 & 0 \\ \hline 0 & 1 & 0 & 0 & 0 \\ \hline 0 & \alpha_{k+1k} & & 0 & 0 \\ \vdots & \vdots & I_{j-k-1} & \vdots & \vdots \\ \hline 0 & \alpha_{j-1k} & & 0 & 0 \\ \hline 0 & \alpha_{jk} & 0 & 1 & 0 \\ \hline 0 & \alpha_{j+1k} & 0 & 0 & \\ \vdots & \vdots & \vdots & \vdots & I_{n-j} \\ \hline 0 & \alpha_{nk} & 0 & 0 & \end{array} \right) \left( \begin{array}{c|c|c|c|c} I_{k-1} & 0 & 0 & 0 & 0 \\ \hline 0 & 1 & 0 & 0 & 0 \\ \hline 0 & & & 0 & 0 \\ \vdots & 0 & I_{j-k-1} & \vdots & \vdots \\ \hline 0 & 0 & 0 & 1 & 0 \\ \hline 0 & 0 & 0 & \beta_{j+1j} & \\ \vdots & \vdots & \vdots & \vdots & I_{n-j} \\ \hline 0 & 0 & 0 & \beta_{nj} & \end{array} \right) =$$

$$\left( \begin{array}{c|c|c|c|c} I_{k-1} & 0 & 0 & 0 & 0 \\ \hline 0 & 1 & 0 & 0 & 0 \\ \hline 0 & \alpha_{k+1k} & & 0 & 0 \\ \vdots & \vdots & I_{j-k-1} & \vdots & \vdots \\ \hline 0 & \alpha_{j-1k} & & 0 & 0 \\ \hline 0 & \alpha_{jk} & 0 & 1 & 0 \\ \hline 0 & \alpha_{j+1k} & 0 & \beta_{j+1j} & \\ \vdots & \vdots & \vdots & \vdots & I_{n-j} \\ \hline 0 & \alpha_{nk} & 0 & \beta_{nj} & \end{array} \right).$$

Los casos para  $j = k + 1$  y  $k = 1$  serían análogos.  $\square$

*Observación 1.2.* La tercera propiedad se generaliza a un número finito, menor o igual que  $n - 1$ , de matrices siempre que la ordenación de sus índices sea creciente.

Considerando en cada etapa las matrices  $\hat{L}_k$  definidas en (1.2) correspondientes, la eliminación de Gauss completa sobre el sistema  $Ax = A^{(0)}x = b = b^{(0)}$  se representaría por

$$\hat{L}_{n-1}\hat{L}_{n-2}\cdots\hat{L}_2\hat{L}_1A^{(0)}x = A^{(n-1)}x = \hat{L}_{n-1}\hat{L}_{n-2}\cdots\hat{L}_2\hat{L}_1b^{(0)} = b^{(n-1)}.$$

Definiendo  $L_k$  la matriz de los coeficientes extradiagonales dados en (1.1) se tiene aplicando el apartado 2 de la proposición anterior  $\hat{L}_k^{-1} = L_k$ . Entonces

$$A^{(0)}x = L_1L_2\cdots L_{n-2}L_{n-1}A^{(n-1)}x = b^{(0)}.$$

Finalmente, denotando por  $L = L_1L_2\cdots L_{n-2}L_{n-1}$  y  $U = A^{(n-1)}$  se tiene:

$$Ax = LUx = b. \tag{1.3}$$

donde  $L$  es una matriz triangular inferior con diagonal de unos, como consecuencia de la proposición 1.1, que coincide con la matriz  $L_k$  en la columna  $k$ .

El cálculo de una factorización  $LU$  es computacionalmente equivalente a una eliminación de Gauss sobre la matriz  $A$  por lo que su coste es  $\mathcal{O}\left(\frac{2}{3}n^3\right)$  operaciones aritméticas siendo  $n$  el orden de la matriz (véase [4]). Además la factorización se puede almacenar en la propia matriz  $A$ , prescindiendo de guardar los unos de la diagonal de  $L$ .

En la expresión (1.3) vemos que resolver el sistema original es equivalente a resolver dos sistemas triangulares:

$$Ax = b \iff LUx = b \iff \begin{cases} Lv = b, \\ Ux = v. \end{cases}$$

Además el procedimiento de resolución es independiente del segundo miembro del sistema por lo que, una vez factorizada la matriz, podremos resolver con un coste  $\mathcal{O}(2n^2)$  operaciones cualquier sistema de matriz de coeficientes  $A$ .

El siguiente resultado, que se puede consultar en [1], establece condiciones suficientes para la existencia y unicidad de factorización  $LU$ .

**Teorema 1.3.** *Sea  $A \in \mathcal{M}_{n \times n}$  tal que los menores principales de orden  $k$ ,  $\Delta_k$ , son invertibles para  $k = 1, 2, \dots, n-1$ . Entonces existen matrices  $L \in \mathcal{M}_{n \times n}$  triangular inferior,  $l_{ii} = 1$  para  $i = 1, 2, \dots, n$  y  $U \in \mathcal{M}_{n \times n}$  triangular superior tales que  $A = LU$ . Además, si  $\Delta_n = \det(A) \neq 0$ , entonces la factorización es única.*

*Demostración.* Etapa 1. Si  $\Delta_1 = a_{11} \neq 0$  entonces se puede realizar la primera etapa de eliminación de Gauss,  $\hat{L}_1 A^{(0)} = A^{(1)}$ . Además por hipótesis  $\Delta_2 = a_{11}^{(0)} a_{22}^{(1)} \neq 0$  implica que  $a_{22}^{(1)} \neq 0$  y se puede realizar la segunda etapa. Supongamos ahora por hipótesis de inducción que  $A^{(k-1)}$  es la matriz resultante de las  $k - 1$  primeras etapas. Además  $\Delta_k = a_{11}^{(0)} a_{22}^{(1)} \dots a_{k-1, k-1}^{(k-2)} a_{kk}^{(k-1)} \neq 0$ , lo que implica que el coeficiente diagonal de la fila  $k$  es no nulo y se puede hacer la etapa  $k$ ,  $\hat{L}_k A^{(k-1)} = A^{(k)}$  matriz con ceros en la columna  $k$ -ésima por debajo de la diagonal. La hipótesis garantiza que podemos realizar  $n - 1$  etapas de eliminación de Gauss para llegar a una matriz triangular superior  $A^{(n-1)}$ . Además  $A^{(n-1)} = \hat{L}_{n-1} \hat{L}_{n-2} \dots \hat{L}_1 A^{(0)}$  es triangular superior. Tomando  $A^{(n-1)} = U$  y  $L = \hat{L}_1^{-1} \hat{L}_2^{-1} \dots \hat{L}_{n-1}^{-1}$  es una matriz triangular con unos en la diagonal gracias a la proposición (1.1).

Unicidad.

Si  $A$  es invertible entonces  $U$  lo es también porque  $\det(A) = \Delta_n = a_{11}^{(0)} a_{22}^{(1)} \dots a_{nn}^{(n-1)}$ . Supongamos que existen dos factorizaciones  $L_\alpha U_\alpha = L_\beta U_\beta$ . Entonces ya que son invertibles  $L_\beta^{-1} L_\alpha = U_\beta U_\alpha^{-1}$  donde  $L_\beta^{-1} L_\alpha$  es triangular inferior con unos en la diagonal, y  $U_\beta U_\alpha^{-1}$  es triangular superior y por tanto la única posibilidad es  $L_\beta^{-1} L_\alpha = I = U_\beta U_\alpha^{-1}$  con lo que  $L_\alpha = L_\beta$  y  $U_\alpha = U_\beta$ .  $\square$

## 1.2. Estrategia de pivote parcial

En la etapa  $k$ -ésima de la eliminación de Gauss podemos encontrar un coeficiente nulo en la diagonal, lo que impediría continuar el procedimiento, aunque la matriz sea invertible. Por otra parte, si los coeficientes extradiagonales de la matriz  $L$  son muy grandes podemos tener graves errores de redondeo al trabajar en la aritmética finita del ordenador como puede verse en [2]. Con el fin salvar estos inconvenientes se utiliza la estrategia de pivote parcial que consiste en realizar la eliminación de la etapa  $k$  en la columna  $k$ , pero con una fila elegida de manera que los coeficientes de las operaciones de fila que hay que realizar sean de módulo menor o igual que uno:

$$\left| a_{\sigma(k)k}^{(k-1)} \right| = \max_{i=k, \dots, n} \left| a_{\sigma(i)k}^{(k-1)} \right|, \quad (1.4)$$

lo que equivale a elegir como fila base de la eliminación aquella que tenga en la columna  $k$  el coeficiente de mayor módulo y se utilizará para reducir a cero los coeficientes de esa columna en las filas  $\sigma(k+1) \dots \sigma(n)$ .

*Observación 1.4.* Si la elección del pivote en cada etapa es única se tendrá que todos los coeficientes extradiagonales de la matriz  $\hat{L}_k$  son de módulo menor que 1.

Con el fin de cuantificar de alguna manera la calidad de los sistemas triangulares obtenidos en la factorización  $LU$  recordamos la noción de condicionamiento de una matriz.

Si  $\|\cdot\|$  es una norma matricial subordinada y  $A$  es una matriz invertible, el número

$$\text{cond}(A) = \|A\| \|A^{-1}\|$$

se llama condicionamiento de la matriz  $A$ , con respecto a la norma matricial considerada. Este número mide la sensibilidad de la solución  $u$  del sistema lineal  $Au = b$  con respecto a las variaciones de  $A$  y  $b$ .

En el siguiente ejemplo [2] podemos ver como la factorización  $LU$  empeora el condicionamiento del sistema lineal y como la estrategia de pivote parcial nos permite reducir ese deterioro del condicionamiento de la matriz del sistema una vez hecha la eliminación. Utilizaremos la norma 1

$$\|A\|_1 = \max_{j=1\dots n} \left\{ \sum_{i=1}^n |a_{ij}| \right\}$$

**Ejemplo 1.5.** Consideremos el siguiente sistema,  $Ax = b$ , siendo  $0 < \epsilon \ll 1$  un número muy pequeño

$$\begin{pmatrix} \epsilon & 1 \\ 1 & 1 \end{pmatrix} \begin{pmatrix} x_1 \\ x_2 \end{pmatrix} = \begin{pmatrix} 1 \\ 2 \end{pmatrix} \quad (1.5)$$

Al aplicar el método de eliminación de Gauss al sistema, obtenemos el siguiente sistema equivalente

$$Ux = \begin{pmatrix} \epsilon & 1 \\ 0 & 1 - \frac{1}{\epsilon} \end{pmatrix} \begin{pmatrix} x_1 \\ x_2 \end{pmatrix} = L^{-1} \begin{pmatrix} 1 \\ 2 \end{pmatrix} = \begin{pmatrix} 1 \\ 2 - \frac{1}{\epsilon} \end{pmatrix}. \quad (1.6)$$

Con

$$L^{-1} = \begin{pmatrix} 1 & 0 \\ -\frac{1}{\epsilon} & 1 \end{pmatrix}$$

Podemos ver que aunque el sistema original es bien condicionado

$$\text{cond}_1(A) = \|A\|_1 \|A^{-1}\|_1 = 2 \cdot \frac{2}{1 - \epsilon} \approx 4 \quad (1.7)$$

siendo  $A^{-1} = \frac{1}{\epsilon - 1} \begin{pmatrix} 1 & -1 \\ -1 & \epsilon \end{pmatrix}$

sin embargo el sistema con matriz de coeficientes  $U$  es mal condicionado, siendo su condicionamiento peor cuanto mas cerca esté  $\epsilon$  de cero:

$$\text{cond}_1(U) = \|U\|_1 \|U^{-1}\|_1 = \frac{1}{\epsilon} \cdot \frac{1}{\epsilon} = \frac{1}{\epsilon^2} \quad (1.8)$$

siendo  $U^{-1} = \begin{pmatrix} \frac{1}{\epsilon} & \frac{1}{1-\epsilon} \\ 0 & \frac{-\epsilon}{1-\epsilon} \end{pmatrix}$ .

Este empeoramiento en el condicionamiento se debe a que usamos un pivote muy pequeño

en el proceso de eliminación. Esto puede solucionarse utilizando una estrategia de pivote propuesta en (1.4) que llevaría a intercambiar las ecuaciones, así obtendríamos el sistema  $PAx = \tilde{L}\tilde{U}x = Pb$  siendo

$$P = \begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix} \quad \tilde{L} = \begin{pmatrix} 1 & 0 \\ \epsilon & 1 \end{pmatrix} \quad \tilde{U} = \begin{pmatrix} 1 & 1 \\ 0 & 1 - \epsilon \end{pmatrix}$$

el condicionamiento de  $PA$  será el mismo que el de  $A$ . Tras aplicar la eliminación tenemos un sistema con matriz de coeficientes  $\tilde{U}$

$$\begin{pmatrix} 1 & 1 \\ 0 & 1 - \epsilon \end{pmatrix} \begin{pmatrix} x_1 \\ x_2 \end{pmatrix} = \tilde{L}^{-1}P \begin{pmatrix} 1 \\ 2 \end{pmatrix} = \begin{pmatrix} 2 \\ 1 - 2\epsilon \end{pmatrix}. \quad (1.9)$$

Ahora este nuevo sistema está bien condicionado, siendo su condicionamiento

$$\text{cond}_1(\tilde{U}) = \|\tilde{U}\|_1 \|\tilde{U}^{-1}\|_1 = (2 - \epsilon) \cdot \frac{2}{1 - \epsilon} \approx 4 \quad (1.10)$$

donde  $\tilde{U}^{-1} = \frac{1}{1-\epsilon} \begin{pmatrix} 1 - \epsilon & -1 \\ 0 & 1 \end{pmatrix}$

*Observación 1.6.* Las matrices  $L$  en la estrategia de pivote, ya que los coeficientes extra-diagonales de la matriz  $L$  son menores que uno, cumplen que  $\text{cond}_1(L) \leq n^2$ , siendo  $n$  el orden de  $L$ .

Para la representación matricial de este método introducimos las matrices permutación, asociadas a una permutación de  $n$  elementos  $\sigma \in S_n$ ,  $\sigma = (\sigma(1), \dots, \sigma(n))$ . Dada  $\sigma \in S_n$  definimos la matriz permutación asociada

$$P_\sigma = \begin{pmatrix} \mathbf{e}^{\sigma(1)} \\ \mathbf{e}^{\sigma(2)} \\ \vdots \\ \mathbf{e}^{\sigma(n)} \end{pmatrix},$$

donde  $\mathbf{e}^i$  denota el  $i$ -ésimo vector de la base canónica de  $\mathbb{R}^n$ , o equivalentemente,  $(P_\sigma)_{ij} = \delta_{\sigma(i)j}$ . Dada  $A \in \mathcal{M}_{n \times n}$  denotamos por  $\mathbf{F}^k$  (respectivamente,  $\mathbf{C}^k$ ) la  $k$ -ésima fila (respectivamente, columna) de  $A$ .

**Proposición 1.7.** *Se verifica:*

1.  $|\det(P_\sigma)| = 1$ ,
2.  $P_\sigma^{-1} = P_\sigma^T$ ,
3.  $P_\sigma v = (v_{\sigma(1)} \ v_{\sigma(2)} \ \cdots \ v_{\sigma(n)})^t$ , para todo  $v \in \mathbb{R}^n$ ,

$$4. P_\sigma A = (\mathbf{F}^{\sigma(1)} \mid \mathbf{F}^{\sigma(2)} \mid \dots \mid \mathbf{F}^{\sigma(n)})^t \text{ para toda } A \in \mathcal{M}_{n \times n},$$

$$5. AP_\sigma^T = (\mathbf{C}^{\sigma(1)} \mid \mathbf{C}^{\sigma(2)} \mid \dots \mid \mathbf{C}^{\sigma(n)}) \text{ para toda } A \in \mathcal{M}_{n \times n}.$$

*Demostración.* 1. Basta con desarrollar el determinante por filas.

$$2. \text{ Para } i, j = 1, \dots, n \text{ se tiene } (P_\sigma P_\sigma^t)_{ij} = \sum_{k=1}^n (P_\sigma)_{ik} (P_\sigma^t)_{kj} = \sum_{k=1}^n (P_\sigma)_{ik} (P_\sigma)_{jk} = \sum_{k=1}^n \delta_{\sigma(i)k} \delta_{\sigma(j)k} = \delta_{\sigma(i)\sigma(j)} = \delta_{ij}.$$

$$3. \text{ Para } i = 1, \dots, n \text{ se tiene } (P_\sigma v)_i = \sum_{k=1}^n (P_\sigma)_{ik} v_k = \sum_{k=1}^n \delta_{\sigma(i)k} v_k = v_{\sigma(i)}.$$

4. Análoga a 3 considerando la matriz particionada por filas.

5. Como  $AP_\sigma^T = (P_\sigma A^T)^T$ , se deduce de 4 teniendo en cuenta que las filas de la matriz  $A^T$  son las columnas de la matriz  $A$ .

□

Si realizamos una eliminación de Gauss sobre la matriz  $A$  con estrategia de pivote parcial en la que para la etapa  $k$ -ésima se ha utilizado la fila  $\sigma(k)$ -ésima se puede construir la factorización  $LU$  de la matriz  $P_\sigma A$ , esto siempre es posible para una matriz invertible. La resolución de un sistema lineal se reduce después de haber hecho una permutación en el segundo miembro a la resolución de dos sistemas triangulares con matrices  $L$  y  $U$

$$Ax = b \iff P_\sigma Ax = P_\sigma b \iff LUx = P_\sigma b \iff \begin{cases} Lv = P_\sigma b, \\ Ux = v. \end{cases}$$

**Teorema 1.8.** *Dada  $A \in \mathcal{M}_{n \times n}$  invertible existe una permutación  $\sigma \in S_n$  tal que  $P_\sigma A = LU$  con  $U$  triangular superior  $L$  triangular inferior,  $l_{ii} = 1$  y  $|l_{ik}| \leq 1$  si  $i > k$ .*

*Demostración.* Si  $A$  es invertible entonces en la primera columna hay al menos un coeficiente no nulo. El de módulo máximo estará en la posición  $a_{\sigma(1)1}$  y tras aplicar la permutación será el coeficiente de la posición  $(1, 1)$  de la matriz. Entonces realizando la primera etapa de eliminación de Gauss  $\hat{L}_1 P_1 A^{(0)} = A^{(1)}$  y así  $A^{(1)}$  tiene ceros en la primera columna por debajo de la diagonal. En la etapa  $k$  se elige entre las filas  $\sigma(k), \dots, \sigma(n)$  la del coeficiente de módulo máximo, que es no nulo porque de lo contrario la matriz sería singular, y se realiza la eliminación de Gauss sobre la matriz permutada  $\hat{L}_k P_k A^{(k-1)} = A^{(k)}$  siendo  $P_k$  la matriz permutación que lleva el pivote de la etapa  $k$  a la posición  $(k, k)$  y  $\hat{L}_k$  la matriz que realiza la reducción, entonces la eliminación sobre  $A = A^{(0)}$  se puede escribir después de  $n - 1$  etapas como

$$\hat{L}_{n-1} P_{n-1} \hat{L}_{n-2} P_{n-2} \cdots \hat{L}_2 P_2 \hat{L}_1 P_1 A^{(0)} = A^{(n-1)}.$$

En general, las matrices  $\hat{L}_k^{-1}P_{k+1}^t$  no son triangulares inferiores con la diagonal de unos. Esto se puede resolver introduciendo permutaciones que afectan tanto a filas como a columnas de la siguiente manera:

$$\begin{aligned}
& \overbrace{\hat{L}_{n-1} P_{n-1} \hat{L}_{n-2} P_{n-1}^t}^{\tilde{L}_{n-2}} \underbrace{P_{n-1} P_{n-2}}_I \overbrace{\hat{L}_{n-3} P_{n-2} P_{n-1}^t}^{\tilde{L}_{n-3}} \underbrace{P_{n-1} P_{n-2} P_{n-3}}_I \overbrace{\hat{L}_{n-4} P_{n-3} P_{n-2} P_{n-1}^t}^{\tilde{L}_{n-4}} \\
& P_{n-1} P_{n-2} P_{n-3} P_{n-4} \hat{L}_{n-5} \cdots P_{n-1} P_{n-2} \cdots P_2 \overbrace{\hat{L}_1 P_2^t \cdots P_{n-2}^t P_{n-1}^t}^{\tilde{L}_1} \underbrace{P_{n-1} P_{n-2} \cdots P_2 P_1}_I A^{(0)} = A^{(n-1)}. \tag{1.11}
\end{aligned}$$

Ahora el producto  $P_{n-1} \hat{L}_{n-2} P_{n-1}^t = \tilde{L}_{n-2}$  es triangular inferior, con diagonal de unos y coeficientes extradiagonales no nulos a lo sumo en la columna  $n - 2$ ; análogamente,  $P_{n-1} P_{n-2} \hat{L}_{n-3} P_{n-2}^t P_{n-1}^t = \tilde{L}_{n-3}$  tendrá sus coeficientes extradiagonales no nulos en la columna  $n - 3$ . Procediendo sucesivamente se llegaría a  $P_{n-1} P_{n-2} \cdots P_2 \hat{L}_1 P_2^t \cdots P_{n-2}^t P_{n-1}^t = \tilde{L}_1$ , matriz que tiene sus coeficientes extradiagonales no nulos en la primera columna. Denotando por  $L_{n-1} = \hat{L}_{n-1}^{-1}$  y  $L_k = \tilde{L}_k^{-1}$ ,  $k = 1, \dots, n - 2$ , resulta

$$PA = P_{n-1} P_{n-2} \cdots P_2 P_1 A^{(0)} = L_1 L_2 \cdots L_{n-2} L_{n-1} A^{(n-1)} = LU,$$

$L$  triangular inferior con diagonal de unos,  $U$  triangular superior y  $P$  matriz permutación.  $\square$

*Observación 1.9.* El resultado de una eliminación de Gauss con estrategia de pivote parcial es una matriz triangular superior salvo permutación de filas, siendo  $U$  la matriz de la factorización de  $P_\sigma A^{(0)}$ . Los coeficientes de la matriz  $L$ , triangular inferior con diagonal de unos, son los factores de las operaciones de fila de la eliminación cambiados de signo, de manera que en la columna  $k$ -ésima se sitúan los de la etapa  $k$  siguiendo las filas la permutación  $\sigma$ , esto es:

$$L = \left( \begin{array}{cccc|c|ccc} 1 & 0 & & & 0 & 0 & \dots & 0 \\ l_{21} & 1 & 0 & & 0 & 0 & \dots & 0 \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & & \vdots \\ l_{k-11} & \dots & l_{k-1k-2} & 1 & 0 & 0 & \dots & 0 \\ \hline l_{k1} & \dots & l_{kk-2} & l_{kk-1} & 1 & 0 & \dots & 0 \\ \hline l_{k+11} & \dots & \dots & l_{k+1k-1} & l_{k+1k} & 1 & \dots & 0 \\ l_{k+21} & \dots & \dots & l_{k+2k-1} & l_{k+2k} & 1 & \dots & 0 \\ \vdots & & \vdots & \vdots & \vdots & \vdots & \ddots & \vdots \\ l_{n1} & \dots & \dots & l_{nk-1} & l_{nk} & l_{nk+1} & \dots & 1 \end{array} \right) \begin{array}{l} \leftarrow \sigma(2) \\ \leftarrow \sigma(k-1) \\ \leftarrow \sigma(k) \\ \leftarrow \sigma(k+1) \\ \leftarrow \sigma(k+2) \\ \leftarrow \sigma(n) \end{array}$$

$\uparrow$   $\uparrow$   
 Etapa 1 Etapa k

$$l_{ik} = a_{\sigma(i)k}^{(k-1)} / a_{\sigma(k)k}^{(k-1)}, \quad i = k+1, \dots, n.$$

*Observación 1.10.* El número de operaciones aritméticas es el mismo que para la factorización sin estrategia de pivote, sin embargo nos obliga a hacer  $\mathcal{O}(\frac{n^2}{2})$  comparaciones.

*Observación 1.11.* La factorización puede almacenarse en una matriz donde estarían  $L$  y  $U$  y en un vector para la permutación  $\sigma$ .

### 1.3. Estrategia de pivote total

Aunque ya vimos que la estrategia de pivote parcial soluciona el problema del bloqueo del método en el caso de pivote nulo, así como también disminuye el mal condicionamiento del sistema reducido obtenido de Gauss sin pivote, puede no corregirlo convenientemente en algunos casos. Para esto introduciremos la estrategia de pivote total, que consiste en realizar la eliminación de la etapa  $k$ , eligiendo el pivote de módulo máximo posible en la matriz. Se calcula una permutación de filas  $\sigma$  y una permutación de columnas  $\tau$ , para realizar la etapa  $k$  en la columna  $\tau(k)$  tomando la fila  $\sigma(k)$  de tal manera que

$$\left| a_{\sigma(k)\tau(k)}^{(k-1)} \right| = \max_{i,j=k,\dots,n} \left| a_{\sigma(i)\tau(j)}^{(k-1)} \right|.$$

Lo que equivale a elegir como fila base de la eliminación aquella que tenga en la columna  $\tau(k)$  el coeficiente de mayor módulo y se utilizará para reducir a 0 los coeficientes de esa columna en las filas  $\sigma(k+1), \dots, \sigma(n)$ . El siguiente ejemplo que se puede encontrar en [8] muestra que es necesario mejorar la estrategia de pivote parcial.

**Ejemplo 1.12.** Consideremos la siguiente matriz

$$A = \begin{pmatrix} 1 & 0 & 0 & 0 & 1 \\ -1 & 1 & 0 & 0 & 1 \\ -1 & -1 & 1 & 0 & 1 \\ -1 & -1 & -1 & 1 & 1 \\ -1 & -1 & -1 & -1 & 1 \end{pmatrix} \quad (1.12)$$

Aplicando la estrategia de pivote parcial  $PA = LU$  obtenemos que  $P$  es la matriz identidad y que  $L$  y  $U$  son

$$L = \begin{pmatrix} 1 & 0 & 0 & 0 & 0 \\ -1 & 1 & 0 & 0 & 0 \\ -1 & -1 & 1 & 0 & 0 \\ -1 & -1 & -1 & 1 & 0 \\ -1 & -1 & -1 & -1 & 1 \end{pmatrix} \quad U = \begin{pmatrix} 1 & 0 & 0 & 0 & 1 \\ 0 & 1 & 0 & 0 & 2 \\ 0 & 0 & 1 & 0 & 4 \\ 0 & 0 & 0 & 1 & 8 \\ 0 & 0 & 0 & 0 & 16 \end{pmatrix} \quad (1.13)$$

por lo que  $\text{cond}_1(U) = \|U\|_1 \|U^{-1}\|_1 = 31$ , siendo

$$U^{-1} = \frac{1}{16} \begin{pmatrix} 16 & 0 & 0 & 0 & -1 \\ 0 & 16 & 0 & 0 & -2 \\ 0 & 0 & 16 & 0 & -4 \\ 0 & 0 & 0 & 16 & -8 \\ 0 & 0 & 0 & 0 & 1 \end{pmatrix}.$$

Aplicando la estrategia de pivote total  $P_\sigma A P_\tau^T = \tilde{L} \tilde{U}$  obtenemos que  $P_\sigma = I$ ,  $P_\tau^T$  es la matriz asociada a la permutación  $\tau = (1, 5, 2, 3, 4)$  y que  $\tilde{L}$  y  $\tilde{U}$  son

$$\tilde{L} = \begin{pmatrix} 1 & 0 & 0 & 0 & 0 \\ -1 & 1 & 0 & 0 & 0 \\ -1 & 1 & 1 & 0 & 0 \\ -1 & 1 & 1 & 1 & 0 \\ -1 & 1 & 1 & 1 & 1 \end{pmatrix} \quad \tilde{U} = \begin{pmatrix} 1 & 1 & 0 & 0 & 0 \\ 0 & 2 & 1 & 0 & 0 \\ 0 & 0 & -2 & 1 & 0 \\ 0 & 0 & 0 & -2 & 1 \\ 0 & 0 & 0 & 0 & -2 \end{pmatrix} \quad (1.14)$$

por lo que  $\text{cond}_1(\tilde{U}) = \|\tilde{U}\|_1 \|\tilde{U}^{-1}\|_1 = 3$ , siendo

$$\tilde{U}^{-1} = \frac{-1}{16} \begin{pmatrix} -16 & 8 & 4 & 2 & 1 \\ 0 & -8 & -4 & -2 & -1 \\ 0 & 0 & 8 & 4 & 2 \\ 0 & 0 & 0 & 8 & 4 \\ 0 & 0 & 0 & 0 & 8 \end{pmatrix}.$$

Nótese que tomando matrices de la misma estructura que  $A$  de orden  $n$ , tenemos que  $\text{cond}_1(L) = \text{cond}_1(\tilde{L}) = n^2$ , pero mientras que en la estrategia de pivote parcial  $\text{cond}_1(U) = 2^n - 1$ , en la de pivote total  $\text{cond}_1(\tilde{U}) = 3$  para cualquier  $n$ .

**Teorema 1.13.** *Dada  $A \in \mathcal{M}_{n \times n}$  invertible existen permutaciones  $\sigma, \tau \in S_n$  tales que  $P_\sigma A P_\tau^T = LU$  con  $U$  triangular superior,  $L$  triangular inferior,  $l_{ii} = 1$  y  $|l_{ik}| \leq 1$  si  $i > k$ .*

*Demostración.* Si  $A$  es invertible entonces en  $A$  hay al menos un coeficiente no nulo. El de módulo máximo estará en la posición  $a_{\sigma(1)\tau(1)}$  y tras aplicar las permutaciones  $P_\sigma^1$  y  $P_\tau^{1T}$  estará en la posición  $(1, 1)$  de la matriz permutada. Sea  $\hat{L}_1$  la matriz que realiza la primera etapa de eliminación de Gauss  $\hat{L}_1 P_\sigma^1 A^{(0)} P_\tau^{1T} = A^{(1)}$ .  $A^{(1)}$  tiene ceros en la primera columna por debajo de la diagonal. En la etapa  $k$  se elige en la submatriz  $A^{(k-1)}$  entre las filas  $\sigma(k) \cdots \sigma(n)$  y las columnas  $\tau(k) \cdots \tau(n)$  el coeficiente de módulo máximo, que es no nulo porque de lo contrario la matriz sería singular, y se realiza la eliminación sobre la matriz  $A^{(k-1)}$ ,  $\hat{L}_k P_\sigma^k A^{(k-1)} P_\tau^{kT} = A^{(k)}$ , donde  $P_\sigma^k$  y  $P_\tau^{kT}$  son las matrices permutación que llevan el pivote de la etapa  $k$  a la posición  $(k, k)$  y  $\hat{L}_k$  la matriz que realiza la eliminación. Entonces la eliminación de Gauss con pivote total sobre  $A = A^{(0)}$  se escribe como

$$\hat{L}_{n-1} P_\sigma^{n-1} \hat{L}_{n-2} \cdots \hat{L}_2 P_\sigma^2 \hat{L}_1 P_\sigma^1 A^{(0)} P_\tau^{1T} \cdots P_\tau^{n-1T} = A^{(n-1)}.$$

Como ya vimos en (1.11) introduciendo las matrices  $P_\sigma^{kT}$  esta expresión se puede reescribir como

$$\tilde{L}_{n-1} \tilde{L}_{n-2} \cdots \tilde{L}_2 \tilde{L}_1 P_\sigma^{n-1} \cdots P_\sigma^1 A^{(0)} P_\tau^{1T} \cdots P_\tau^{n-1T} = A^{(n-1)},$$

o equivalentemente

$$P_\sigma^{n-1} \cdots P_\sigma^1 A^{(0)} P_\tau^{1T} \cdots P_\tau^{n-1T} = \tilde{L}_1^{-1} \tilde{L}_2^{-1} \cdots \tilde{L}_{n-2}^{-1} \tilde{L}_{n-1}^{-1} A^{(n-1)}.$$

Definiendo  $L_k = \tilde{L}_k^{-1}$  siendo las matrices  $\tilde{L}_k$  triangulares inferiores con unos en la diagonal y coeficientes extradiagonales nulos salvo eventualmente en la columna  $k$  y aplicando la proposición 1.1 llegamos a

$$P_\sigma A P_\tau^T = LU.$$

Con  $L$  triangular inferior con unos en la diagonal y  $U$  triangular superior. □

Dado que  $P_\sigma A P_\tau^T = LU$  y que  $P_\tau^{-1} = P_\tau^T$  entonces  $P_\sigma A = L U P_\tau$  por lo que la resolución de un sistema lineal  $Ax = b$  sería equivalente a la resolución de dos sistemas triangulares y una permutación en el segundo miembro

$$Ax = b \iff P_\sigma Ax = P_\sigma b \iff L U P_\tau x = P_\sigma b \iff \begin{cases} Lv = P_\sigma b \\ Uw = v \\ P_\tau x = w. \end{cases} \quad (1.15)$$

*Observación 1.14.* La última ecuación no obliga a resolver un sistema, ya que  $x_{\tau(i)} = w_i$ ,  $i = 1, \dots, n$ .

*Observación 1.15.* En esta memoria no se hace un cálculo exhaustivo del número de operaciones aritméticas necesarias para los métodos de factorización estudiados porque son las mismas que las de los métodos sin estrategias de pivote. Solamente se indica una estimación del coste de la búsqueda de pivote. Ya se ha visto que para el método de pivote parcial el coste es de  $\mathcal{O}(\frac{n^2}{2})$  mientras que para pivote total es  $\mathcal{O}(\frac{n^3}{3})$ .

*Observación 1.16.* El coste de almacenamiento es una matriz para la factorización y dos vectores, uno para cada permutación.

*Observación 1.17.* Si la matriz  $A \in \mathcal{M}_{n \times n}$  es de rango  $l \leq n$  este procedimiento se bloquea en la etapa  $l + 1$  por tanto se puede usar este método para determinar el rango de una matriz.

$$A^{(l)} = \left( \begin{array}{cccc|cccc} a_{\sigma(1)\tau(1)} & a_{\sigma(1)\tau(2)} & \cdots & a_{\sigma(1)\tau(l)} & a_{\sigma(1)\tau(l+1)} & a_{\sigma(1)\tau(l+2)} & \cdots & a_{\sigma(1)\tau(n)} \\ 0 & \ddots & & a_{2l} & a_{\sigma(2)\tau(l+1)} & a_{\sigma(2)\tau(l+2)} & \cdots & a_{\sigma(2)\tau(n)} \\ \vdots & & \ddots & \vdots & \vdots & \vdots & & \vdots \\ 0 & \cdots & 0 & a_{\sigma(l)\tau(l)} & a_{\sigma(l)\tau(l+1)} & a_{\sigma(l)\tau(l+2)} & \cdots & a_{\sigma(l)\tau(n)} \\ \hline 0 & \cdots & \cdots & 0 & 0 & 0 & \cdots & 0 \\ \vdots & & & \vdots & 0 & 0 & \cdots & 0 \\ \vdots & & & \vdots & \vdots & \vdots & & \vdots \\ 0 & \cdots & \cdots & 0 & 0 & 0 & \cdots & 0 \end{array} \right) \quad (1.16)$$

*Observación 1.18.* Puede usarse este método para discutir un sistema, ya que como vimos en (1.15) después de resolver el sistema triangular inferior llegamos a

$$UP_{\tau}x = L^{-1}P_{\sigma}b = v$$

por lo que examinando las  $n-l$  últimas componentes de  $v$  obtenemos que si  $\|(v_{l+1}, \dots, v_n)\| \neq 0$  el sistema es incompatible, mientras que si  $\|(v_{l+1}, \dots, v_n)\| = 0$  es compatible indeterminado.

*Observación 1.19.* Los métodos de factorización descritos en este capítulo son los más eficientes para calcular el determinante de una matriz general. Si  $A = LU$  entonces  $\det(A) = \det(U)$  y si  $P_{\sigma}AP_{\tau}^T = LU$  entonces  $\det(A) = \det(U) \det(P_{\sigma}) \det(P_{\tau})$  ya que  $\det(L) = 1$ , el determinante de  $P_{\sigma}$  es -1 elevado al número de cambios de filas.

*Observación 1.20.* Si se dispone de un método de factorización el cálculo de la inversa se realiza resolviendo  $n$  sistemas lineales para la base canónica de  $\mathbb{R}^n$  (respectivamente  $\mathbb{C}^n$  para aritmética compleja).

### 1.3.1. Código Fortran 90

```

1  subroutine lupt(a, ip, iq, deter)
2  !*****
3  !factorizacion LU de la matriz "a" con la estrategia de pivote total
4  !*****
5  !ENTRADA:
6  ! a      matriz
7  !SALIDA:
8  ! a      (sup) matriz U
9  ! a      (inf) matriz L sin diagonal de unos
10 ! ip     permutaciones de filas
11 ! iq     permutaciones de columnas
12 ! deter  determinante
13 !*****
14 use mod_creal
15 implicit none
16 real(kind=creal), intent(out)::deter
17 integer, intent(out), dimension(:)::ip, iq
18 real(kind=creal), intent(inout), dimension(:, ::)::a
19 real(kind=creal)::piv
20 integer::cont, i, j, k, n, ipiv, jpiv, ipk, jpk, ipi, jpi
21
22 n=size(a, 1)
23 !inicializacion del determinante
24 deter=1.
25 !inicializacion de la permutacion de filas y columnas
26 do i=1, n
27   ip(i)=i
28   iq(i)=i
29 end do
30 !inicializacion do contador de cambios de filas y columnas
31 cont=0
32 !etapa k-esima da eliminacion
33 do k=1, n-1
34 !busqueda del pivote
35   piv=a(ip(k), iq(k))
36   ipiv=k
37   jpiv=k
38   do i=k, n

```

```

39   do j=k,n
40     if(abs(piv)<abs(a(ip(i),iq(j)))) then
41       piv=a(ip(i),iq(j))
42       ipiv=i
43       jpiv=j
44     end if
45   end do
46 end do
47 !comprobacion de que el k-esimo pivote no es nulo
48 if(abs(piv)<1.e-12) then
49   print*, 'pivote nulo en la etapa: ',k
50   print*, 'la matriz es de rango ',k-1
51   stop
52 end if
53 !actualizacion de las permutaciones de filas
54 if(ipiv/=k) then
55   ipk=ip(ipiv)
56   ip(ipiv)=ip(k)
57   ip(k)=ipk
58   cont=cont+1
59 else
60   ipk=ip(k)
61 end if
62 !actualizacion de las permutaciones de columnas
63 if(jpiv/=k) then
64   jpk=iq(jpiv)
65   iq(jpiv)=iq(k)
66   iq(k)=jpk
67   cont=cont+1
68 else
69   jpk=iq(k)
70 end if
71 !actualizacion del determinante
72 deter=deter*piv
73 !eliminacion
74 do i=k+1,n
75   ipi=ip(i)
76 !coeficiente de L
77   a(ipi,jpk)=a(ipi,jpk)/piv
78   do j=k+1,n
79     jpi=iq(j)
80     a(ipi,jpi)=a(ipi,jpi)-a(ipi,jpk)*a(ipk,jpi)
81   end do
82 end do
83 end do

```

```
84 !comprobacion de que el ultimo pivote no es nulo
85 piv=a(ip(n),iq(n))
86 if(abs(piv)<1.e-12) then
87   print*, 'pivote nulo en la etapa: ',n
88   print*, 'la matriz es de rango! 'n-1
89   stop
90 end if
91 !actualizacion del determinante
92 deter=deter*piv*(-1)**cont
93 end subroutine
```

## Capítulo 2

# Factorizaciones para matrices hermitianas

En este capítulo veremos los métodos de factorización especialmente adaptados al caso de matrices hermitianas. Se empezará obteniendo la factorización de Crout para el caso de una matriz hermitiana invertible con factorización  $LU$ , para concluir con la factorización de Cholesky de una matriz definida positiva que es el método de preferencia para resolver este tipo de sistemas.

### 2.1. Factorización de Crout

En esta sección empezaremos construyendo la factorización de Crout para una matriz hermitiana invertible con factorización  $LU$ .

**Definición 2.1.** Una matriz  $A$  se dice hermitiana si  $A = A^*$  siendo  $A^*$  la transpuesta conjugada de  $A$ .

**Definición 2.2.** Una matriz  $A$  se dice definida positiva si  $x^T Ax > 0$  para todo  $x \in \mathbb{C}^n - \{\mathcal{O}\}$ .

Para el caso de matrices reales tenemos las siguientes definiciones.

**Definición 2.3.** Una matriz  $A$  se dice simétrica si  $A = A^T$ .

**Definición 2.4.** Una matriz  $A$  se dice definida positiva si  $x^T Ax > 0$  para todo  $x \in \mathbb{R}^n - \{\mathcal{O}\}$ .

Recordamos a continuación un resultado de [2] que nos resultará útil en esta sección.

**Proposición 2.5.** *Sea  $A = A^* \in \mathcal{M}_{n \times n}$  definida positiva. Entonces*

1. Si  $M \in \mathcal{M}_{m \times n}$  con  $m \leq n$  tiene rango  $m$  entonces  $MAM^*$  es definida positiva.
2. Todos los menores principales de  $A$  son definidos positivos.

*Demostración.* 1. Sea  $z$  un vector no nulo de  $\mathbb{C}^m$ . El hecho de que  $M$  tenga rango  $m$  implica que  $y := M^*z \neq \mathcal{O}$ , ya que de lo contrario las columnas de  $M$  serían linealmente dependientes, y de esa manera  $M$  tendría rango menor que  $m$ . Por ser  $A$  definida positiva tenemos

$$y^*Ay = z^*MAM^*z > 0$$

con lo que  $MAM^*$  es definida positiva.

2. Sea  $J = \{j_1, \dots, j_m\}$  un subconjunto arbitrario de  $\{1, \dots, n\}$  y  $B$  la submatriz correspondiente a extraer las filas  $\{j_1 \dots j_m\}$  y las columnas  $\{j_1 \dots j_m\}$  de la matriz  $A$ . Definimos la matriz  $R \in \mathcal{M}_{n \times m}$  como

$$r_{jk} = \begin{cases} 1, & j = j_k \\ 0, & \text{otro caso} \end{cases}$$

Vemos entonces que  $B = R^TAR$  y por lo visto en el apartado anterior tomando  $M = R^T$  vemos que  $B$  es definida positiva. □

Sea  $A$  una matriz hermitiana invertible con factorización  $LU$ . Entonces  $U$  es invertible. Además

$$A = LU = A^* = (LU)^* = U^*L^*$$

lo que implica que

$$UL^{-*} = L^{-1}LUL^{-*} = L^{-1}U^*L^*L^{-*} = L^{-1}U^* = D$$

lo que muestra que  $D$  es triangular superior y triangular inferior por lo que solo puede ser diagonal. Además  $U = DL^*$  y por tanto  $A = LDL^*$ .

A esta factorización se la conoce como factorización de Crout. A continuación enunciamos y probamos una propiedad relativa a matrices definidas positivas que será de interés en la siguiente sección.

**Proposición 2.6.** *Si  $A$  es definida positiva entonces  $D$  es definida positiva.*

*Demostración.* Como  $A = LDL^*$  y  $L$  es invertible podemos escribir esa igualdad como  $D = L^{-1}AL^{-*}$  y por lo visto en la proposición 2.5 tomando  $M = L^{-1}$  resulta que  $D$  es definida positiva. □

## 2.2. Factorización de Cholesky

La factorización de Cholesky es el método más eficiente para resolver sistemas cuya matriz de coeficientes es definida positiva y hermitiana. El siguiente resultado establece condiciones suficientes para la existencia y unicidad de la factorización de Cholesky de una matriz hermitiana. El proceso de demostración es una generalización del seguido por [2] para el caso de matrices reales.

**Teorema 2.7.** 1. Si  $A$  es una matriz hermitiana y definida positiva, existe una matriz triangular inferior  $B$  tal que  $A = BB^*$ . Además si  $b_{ii} > 0$ ,  $i = 1, \dots, n$  la factorización es única.

2. Si  $A$  es una matriz simétrica y definida positiva, existe una matriz real triangular inferior  $B$  tal que  $A = BB^T$ . Además si  $b_{ii} > 0$   $\forall i = 1, \dots, n$  la factorización es única.

*Demostración.* 1. Como  $A$  es hermitiana y  $\Delta_i \neq 0$  para  $i = 1, \dots, n$  por ser definida positiva es invertible y tiene factorización  $LU$ . Entonces tiene factorización de Crout con matriz  $D$  hermitiana y definida positiva por la proposición 2.6. Además reescribiendo  $D = D^{\frac{1}{2}}D^{\frac{1}{2}}$  llegamos a  $A = LDL^* = LD^{\frac{1}{2}}D^{\frac{1}{2}}L^* = (LD^{\frac{1}{2}})(LD^{\frac{1}{2}})^*$  y tomando  $B = LD^{\frac{1}{2}}$  se tiene la existencia de la factorización.

Unicidad.  $L$  es única pues la factorización  $LU$  es única y eligiendo signo positivo para las raíces cuadradas en  $D^{\frac{1}{2}}$  también es única ( $b_{ii} = (D^{\frac{1}{2}})_{ii}$ ) para  $i = 1, \dots, n$ .

2. La factorización  $LU$  de una matriz con coeficientes reales tiene coeficientes reales. □

Por analogía a la interpretación matricial de la eliminación de Gauss se pueden organizar los cálculos de la factorización de Cholesky de la siguiente manera, (véase [5])

**Lema 2.8.** Sea  $A \in \mathcal{M}_{n \times n}$  matriz hermitiana y definida positiva, entonces

$$\begin{aligned} \hat{L}A\hat{L}^* &= \left( \begin{array}{c|ccc} 1 & 0 & \cdots & 0 \\ \hline -\frac{a_{21}}{a_{11}} & & & \\ \vdots & & & \\ -\frac{a_{n1}}{a_{11}} & & & \\ \hline & I_{n-1} & & \end{array} \right) \left( \begin{array}{c|ccc} a_{11} & \bar{a}_{21} & \cdots & \bar{a}_{n1} \\ \hline a_{21} & & & \\ \vdots & & & S \\ a_{n1} & & & \end{array} \right) \left( \begin{array}{c|ccc} 1 & -\frac{\bar{a}_{21}}{a_{11}} & \cdots & -\frac{\bar{a}_{n1}}{a_{11}} \\ \hline 0 & & & \\ \vdots & & & I_{n-1} \\ 0 & & & \end{array} \right) = \\ &= \left( \begin{array}{c|ccc} a_{11} & 0 & \cdots & 0 \\ \hline 0 & & & \\ \vdots & & & S - \frac{w_1 w_1^*}{a_{11}} \\ 0 & & & \end{array} \right) \end{aligned} \quad (2.1)$$

siendo  $w_1 = \left(-\frac{a_{21}}{a_{11}}, \dots, -\frac{a_{n1}}{a_{11}}\right)$ . Además  $S - \frac{w_1 w_1^*}{a_{11}}$  es definida positiva.

*Demostración.* La matriz  $B$  está bien definida en cualquier caso puesto que  $a_{11}$  es positivo. La igualdad (2.1) se comprueba realizando el cálculo. Si  $A$  es definida positiva entonces por el apartado 1 de la proposición 2.5  $\hat{L}A\hat{L}^*$  es definida positiva, y por el apartado 2 se tiene que  $S - \frac{w_1 w_1^*}{a_{11}}$  también es definida positiva.  $\square$

Se enuncia la adaptación normalizada del lema anterior que permitirá construir la factorización de Cholesky.

**Lema 2.9.** *Sea  $A \in \mathcal{M}_{n \times n}$  matriz hermitiana y definida positiva, entonces*

$$\begin{aligned} \hat{B}A\hat{B}^* &= \left( \begin{array}{c|ccc} \frac{1}{\sqrt{a_{11}}} & 0 & \cdots & 0 \\ \hline -\frac{a_{21}}{\sqrt{a_{11}}} & & & \\ \vdots & & & \\ -\frac{a_{n1}}{\sqrt{a_{11}}} & & & \\ \hline & I_{n-1} & & \end{array} \right) \left( \begin{array}{c|ccc} a_{11} & \bar{a}_{21} & \cdots & \bar{a}_{n1} \\ \hline a_{21} & & & \\ \vdots & & & \\ a_{n1} & & & \\ \hline & S & & \end{array} \right) \left( \begin{array}{c|ccc} \frac{1}{\sqrt{a_{11}}} & -\frac{\bar{a}_{21}}{\sqrt{a_{11}}} & \cdots & -\frac{\bar{a}_{n1}}{\sqrt{a_{11}}} \\ \hline 0 & & & \\ \vdots & & & \\ 0 & & & \\ \hline & I_{n-1} & & \end{array} \right) = \\ &= \left( \begin{array}{c|ccc} 1 & 0 & \cdots & 0 \\ \hline 0 & & & \\ \vdots & & & \\ 0 & & & \\ \hline & S - w_1 w_1^* & & \end{array} \right) \end{aligned} \quad (2.2)$$

siendo  $w_1 = \left(-\frac{a_{21}}{\sqrt{a_{11}}}, \dots, -\frac{a_{n1}}{\sqrt{a_{11}}}\right)$ . Además  $S - w_1 w_1^*$  también es definida positiva.

*Demostración.* La igualdad se comprueba realizando el cálculo y el carácter definido positivo se deriva de la proposición 2.5.  $\square$

Con las matrices  $\hat{B}$  del lema anterior el cálculo de la  $k$ -ésima etapa de la factorización de Cholesky se escribirá

$$\left( \begin{array}{c|cc} I_{k-1} & 0 & 0 \\ \hline 0 & \frac{1}{\sqrt{a_{kk}}} & 0 \\ \hline 0 & \frac{a_{k+1k}}{\sqrt{a_{kk}}} & \\ \vdots & \vdots & \\ 0 & \frac{a_{nk}}{\sqrt{a_{kk}}} & \\ \hline & I_{n-k} & \end{array} \right) \left( \begin{array}{c|ccc} I_{k-1} & 0 & 0 & \cdots & 0 \\ \hline 0 & a_{kk}^{(k-1)} & \bar{a}_{kk+1}^{(k-1)} & \cdots & \bar{a}_{kn}^{(k-1)} \\ \hline 0 & a_{k+1k}^{(k-1)} & a_{k+1k+1}^{(k-1)} & \cdots & a_{k+1n}^{(k-1)} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 0 & a_{nk}^{(k-1)} & a_{nk+1}^{(k-1)} & \cdots & a_{nn}^{(k-1)} \\ \hline & & & & \end{array} \right) \left( \begin{array}{c|ccc} I_{k-1} & 0 & 0 & \cdots & 0 \\ \hline 0 & \frac{1}{\sqrt{a_{kk}}} & \frac{\bar{a}_{k+1k}}{\sqrt{a_{kk}}} & \cdots & \frac{\bar{a}_{nk}}{\sqrt{a_{kk}}} \\ \hline 0 & 0 & & & \\ \vdots & \vdots & & & \\ 0 & 0 & & & \\ \hline & & I_{n-k} & & \end{array} \right) =$$

$$= \left( \begin{array}{c|cc} I_{k-1} & 0 & 0 & \cdots & 0 \\ \hline 0 & 1 & 0 & \cdots & 0 \\ \hline 0 & 0 & a_{k+1k+1}^{(k)} & \cdots & a_{k+1n}^{(k)} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & a_{nk+1}^{(k)} & \cdots & a_{nn}^{(k)} \end{array} \right),$$

y la última etapa sería

$$\left( \begin{array}{c|c} I_{n-1} & 0 \\ \hline 0 & \frac{1}{\sqrt{a_{kk}}} \end{array} \right) \left( \begin{array}{c|c} I_{n-1} & 0 \\ \hline 0 & a_{kk} \end{array} \right) \left( \begin{array}{c|c} I_{n-1} & 0 \\ \hline 0 & \frac{1}{\sqrt{a_{kk}}} \end{array} \right) = A^{(n)} = I.$$

El método de Cholesky para resolver un sistema lineal  $Au = b$  que posea una matriz  $A$  hermitiana y definida positiva, consiste en calcular la factorización  $A = BB^*$  y a continuación, resolver los dos sistemas lineales de matrices triangulares

$$Ax = b \iff BB^*x = b \iff \begin{cases} Bv = b, \\ B^*x = v. \end{cases}$$

La factorización de Cholesky es computacionalmente más eficiente que la factorización  $A = LU$ , pues su coste es  $O(\frac{n^3}{3})$  operaciones aritméticas, pero nos obliga a calcular  $n$  raíces cuadradas, siendo  $n$  el orden de la matriz (como puede verse en [1]). La matriz  $B$  llamada factor de Cholesky de  $A$  puede almacenarse en  $\mathcal{O}(\frac{n^2}{2})$  zonas de memoria correspondientes a la parte triangular inferior. Finalmente notemos que el cálculo del determinante de la matriz  $A$  es inmediato a partir de su factorización  $A = BB^*$  ya que

$$\det(A) = \det(BB^*) = b_{11}^2 b_{22}^2 \cdots b_{nn}^2.$$

### 2.3. Factorización de Cholesky con estrategia de pivote

Al igual que vimos en la factorización  $LU$  el uso de estrategias con pivote permite reducir los errores de redondeo debidos a la aritmética finita. Aunque en este caso las consecuencias de no usar pivote no son tan dramáticas ya que

$$\|A\|_2 = \rho(A) = \rho(BB^*) = \|B\|_2^2$$

y por consiguiente;  $cond_2(A) = cond_2(B)^2$ , pues una matriz hermitiana es normal ( $A^*A = AA^*$ ) y su norma dos es su radio espectral. Por ello la factorización de Cholesky no deteriora el condicionamiento del sistema a diferencia de la  $LU$ . Pero aun así en problemas mal condicionados resulta interesante realizar una estrategia de pivote. En [2] se realiza un análisis detallado de la estabilidad del método.

En este caso limitaremos la búsqueda del pivote a los coeficientes diagonales para no alterar el carácter simétrico o hermitiano de la matriz. Se factoriza entonces la matriz  $PAP^T = BB^*$ .

En primer lugar veremos que se puede demostrar un resultado análogo a la proposición 1.1 para matrices  $\hat{B}$  de la ecuación (2.2) lo cual nos permitirá, a partir del proceso de eliminación hecho en la anterior sección, dar una interpretación matricial.

**Proposición 2.10.** *Sea*

$$B_k = \left( \begin{array}{c|cc} I_{k-1} & 0 & 0 \\ \hline 0 & b_{kk} & 0 \\ \hline 0 & b_{k+1k} & \\ & \vdots & \\ & b_{nk} & I_{n-k} \end{array} \right)$$

*Se verifica:*

$$1. (B_k^{-1})_{ij} = \begin{cases} 1 & \text{si } i = j \neq k \\ \frac{1}{b_{kk}} & \text{si } i = j = k \\ \frac{-b_{ij}}{b_{kk}} & \text{si } i \neq j \end{cases},$$

2. Si  $1 \leq k < j \leq n$ , entonces

$$(B_k B_j)_{rs} = \begin{cases} \alpha_{kk} & \text{si } r = s = k \\ \beta_{jj} & \text{si } r = s = j \\ \alpha_{rk} & \text{si } r \neq s = k \\ \beta_{rj} & \text{si } r \neq s = j \\ 0 & \text{en otro caso,} \end{cases}$$

denotando con  $\alpha$  los coeficientes de  $B_k$  y con  $\beta$  los de  $B_j$ .

*Demostración.* a) Multiplicando por bloques

$$B_k B_k^{-1} = \left( \begin{array}{c|cc} I_{k-1} & 0 & 0 \\ \hline 0 & b_{kk} & 0 \\ \hline 0 & b_{k+1k} & \\ & \vdots & \\ & b_{nk} & I_{n-k} \end{array} \right) \left( \begin{array}{c|cc} I_{k-1} & 0 & 0 \\ \hline 0 & \frac{1}{b_{kk}} & 0 \\ \hline 0 & -\frac{b_{k+1k}}{b_{kk}} & \\ & \vdots & \\ & -\frac{b_{nk}}{b_{kk}} & I_{n-k} \end{array} \right) =$$

$$= \left( \begin{array}{c|cc} I_{k-1} & 0 & 0 \\ \hline 0 & \frac{b_{kk}}{b_{kk}} & 0 \\ \hline 0 & \frac{b_{k+1k}}{b_{kk}} - \frac{b_{k+1k}}{b_{kk}} & \\ \vdots & \vdots & \\ \frac{b_{nk}}{b_{kk}} - \frac{b_{nk}}{b_{kk}} & & I_{n-k} \end{array} \right).$$

b) Multiplicando por bloques

$$\left( \begin{array}{c|cc|cc} I_{k-1} & 0 & 0 & 0 & 0 \\ \hline 0 & \alpha_{kk} & 0 & 0 & 0 \\ \hline & \alpha_{k+1k} & & & \\ 0 & \vdots & I_{j-k-1} & 0 & 0 \\ & \alpha_{j-1k} & & & \\ \hline 0 & \alpha_{jk} & 0 & 1 & 0 \\ \hline & \alpha_{j+1k} & & & \\ 0 & \vdots & 0 & 0 & I_{n-j} \\ & \alpha_{nk} & & & \end{array} \right) \left( \begin{array}{c|cc|cc} I_{k-1} & 0 & 0 & 0 & 0 \\ \hline 0 & 1 & 0 & 0 & 0 \\ \hline & & I_{j-k-1} & 0 & 0 \\ \hline 0 & 0 & 0 & \beta_{jj} & 0 \\ \hline & & & \beta_{j+1j} & \\ 0 & 0 & 0 & \vdots & I_{n-j} \\ & & & \beta_{nj} & \end{array} \right) =$$

$$\left( \begin{array}{c|cc|cc} I_{k-1} & 0 & 0 & 0 & 0 \\ \hline 0 & \alpha_{kk} & 0 & 0 & 0 \\ \hline & \alpha_{k+1k} & & & \\ 0 & \vdots & I_{j-k-1} & 0 & 0 \\ & \alpha_{j-1k} & & & \\ \hline 0 & \alpha_{jk} & 0 & \beta_{jj} & 0 \\ \hline & \alpha_{j+1k} & & \beta_{j+1j} & \\ 0 & \vdots & 0 & \vdots & I_{n-j} \\ & \alpha_{nk} & & \beta_{nj} & \end{array} \right).$$

Los casos para  $k = 1$ ,  $j = k + 1$  y  $n = j$  supondrían eliminar la fila y columna correspondientes.  $\square$

Ahora ya estamos en condiciones de dar la interpretación matricial de la factorización de Cholesky con pivote. Partimos de la matriz  $A = A^{(0)}$  y en la etapa  $k$ -ésima buscamos en la diagonal el pivote de módulo máximo y lo llevamos a la posición  $(k, k)$ :

$$\left| a_{\sigma(k)\sigma(k)}^{(k-1)} \right| = \max_{i=k, \dots, n} \left| a_{\sigma(i)\sigma(i)}^{(k-1)} \right|.$$

Para ello multiplicamos  $A^{(k-1)}$  por  $P_k$  y  $P_k^T$  siendo éstas matrices de permutación. A continuación se realiza la eliminación multiplicando por  $\hat{B}_k$  por la izquierda y por  $\hat{B}_k^T$  por la derecha para preservar la simetría. Realizadas  $n$  etapas llegamos a

$$\overbrace{\hat{B}_n \hat{B}_{n-1} P_{n-1} \hat{B}_{n-2} \cdots \hat{B}_2 P_2 \hat{B}_1 P_1}^R A^{(0)} \overbrace{P_1^T \hat{B}_1^* P_2^T \hat{B}_2^* P_3^T \cdots \hat{B}_{n-2}^* P_{n-1}^T \hat{B}_{n-1}^* \hat{B}_n^*}^{R^*} = I \quad (2.3)$$

Aunque las matrices  $\hat{B}_k^{-1}$  son triangulares inferiores, en general las matrices  $\hat{B}_k^{-*} P_{k+1}^T$  no lo son, esto se puede resolver incluyendo permutaciones  $P_j$  y  $P_j^T$  de manera similar a lo visto en (1.11). Los productos anteriores se pueden expresar como

$$R = \hat{B}_n \hat{B}_{n-1} \overbrace{P_{n-1} \hat{B}_{n-2} P_{n-1}^T}^{\tilde{B}_{n-2}} \cdots \overbrace{P_{n-1} \cdots P_3 \hat{B}_2 P_3^T \cdots P_{n-1}^T}^{\tilde{B}_2} \overbrace{P_{n-1} \cdots P_3 P_2 \hat{B}_1 P_2^T \cdots P_{n-1}^T}^{\tilde{B}_1} P_{n-1} \cdots P_2 P_1$$

$$R^* = P_1^T P_2^T \cdots P_{n-1}^T \overbrace{P_{n-1} \cdots P_2 \hat{B}_1^* P_2^T P_3^T \cdots P_{n-1}^T}^{\tilde{B}_1^*} \overbrace{P_{n-1} \cdots P_3 \hat{B}_2^* P_3^T \cdots P_{n-1}^T}^{\tilde{B}_2^*} \cdots \overbrace{P_{n-1} \hat{B}_{n-2}^* P_{n-1}^T}^{\tilde{B}_{n-2}^*} \hat{B}_{n-1}^* \hat{B}_n^*$$

Entonces tomando  $\tilde{B}_n := \hat{B}_n$ ,  $\tilde{B}_{n-1} := \hat{B}_{n-1}$  y  $\tilde{B}_{n-2} := P_{n-1} \hat{B}_{n-2} P_{n-1}^T$  y así sucesivamente hasta  $\tilde{B}_1 := P_{n-1} \cdots P_2 \hat{B}_1 P_2^T \cdots P_{n-1}^T$  podemos expresar el producto (2.3) de la siguiente manera:

$$\tilde{B}_n \tilde{B}_{n-1} \cdots \tilde{B}_1 P_{n-1} \cdots P_1 A^{(0)} P_1^T \cdots P_{n-1}^T \tilde{B}_1^* \cdots \tilde{B}_{n-1}^* \tilde{B}_n^* = I.$$

$$P_{n-1} \cdots P_1 A^{(0)} P_1^T \cdots P_{n-1}^T = \tilde{B}_1^{-1} \tilde{B}_2^{-1} \cdots \tilde{B}_n^{-1} \tilde{B}_n^{-*} \cdots \tilde{B}_2^{-*} \tilde{B}_1^{-*}.$$

Ahora las  $\tilde{B}_k^{-1}$  si son triangulares inferiores con diagonal de unos salvo eventualmente en la columna  $k$  por tanto estamos en la condiciones de la proposición 2.10. Definiendo  $P := P_{n-1} \cdots P_1$  y  $B := \tilde{B}_1^{-1} \cdots \tilde{B}_n^{-1}$  llegamos a

$$PAP^T = BB^*.$$

siendo  $B$  una matriz triangular inferior que tiene en la columna  $k$  los coeficientes correspondientes a la matriz empleada en la etapa  $k$ . Este proceso de cálculo que acabamos de desarrollar demuestra el siguiente teorema:

**Teorema 2.11.** *Sea  $A$  una matriz hermitiana definida positiva. Existen matrices  $P$  permutación y  $B$  triangular inferior,  $b_{ii} > 0$  para  $i = 1, \dots, n$  tales que  $PAP^T = BB^*$ . Además  $b_{ii} \geq b_{i+1i+1}$  para todo  $i = 1, \dots, n-1$ .*

Dado que  $PAP^T = BB^*$  y que  $P^{-1} = P^T$  entonces  $PA = BB^*P$  por lo que la solución del sistema lineal se calcularía resolviendo los siguientes sistemas:

$$Ax = b \iff PAx = Pb \iff BB^*Px = Pb \iff \begin{cases} Bv = Pb \\ B^*w = v \\ Px = w. \end{cases}$$

*Observación 2.12.* Puesto que el pivote sólo se toma en las posiciones diagonales el coste de la búsqueda del pivote es  $\mathcal{O}(\frac{n^2}{2})$  para las  $n$  etapas.

*Observación 2.13.* Para realizar el cálculo del determinante basta considerar que

$$\det(A) = \det(PAP^T) = \det(P)^2 \det(A) = \det(BB^*) = b_{\sigma(1)\sigma(1)}^2 \cdots b_{\sigma(n)\sigma(n)}^2.$$

### 2.3.1. Código Fortran 90

```

1  subroutine choleskypiv(a, ip, det)
2  !*****
3  !factorizacion de cholesky de "a" con estrategia de pivote
4  !*****
5  !ENTRADA:
6  ! a      matriz
7  !SALIDA:
8  ! a      matriz factorizada
9  ! ip     permutacion
10 ! det    determinante
11 !*****
12 use mod_creal
13 implicit none
14 real(kind=creal), intent(inout), dimension(:, :):: a
15 integer, intent(out), dimension(:):: ip
16 real(kind=creal), intent(out):: det
17 real(kind=creal):: piv
18 integer :: i, j, k, m, n, ipiv, jpiv, ipk
19
20 n=size(a, 1)
21 !inicializacion del determinante
22 det=1.
23 !inicializacion de la permutacion
24 do i=1, n
25   ip(i)=i
26 end do
27 !etapa k-esima
28 do k=1, n-1
29 !busqueda de pivote

```

```

30  piv=k
31  piv=a(ip(k),ip(k))
32  do m=k+1,n
33    if(abs(piv)<abs(a(ip(m),ip(m)))) then
34      piv=m
35      piv=a(ip(m),ip(m))
36    end if
37  end do
38  if(ipiv/=k) then
39    ipk=ip(ipiv)
40    ip(ipiv)=ip(k)
41    ip(k)=ipk
42  end if
43  piv=a(ip(k),ip(k))
44  det=det*piv
45  !diagonal i=j
46  if(piv<1.e-16) then
47    write(*,*)"no es definida positiva , piv=",piv,"para etapa=",k
48    print*, 'diagonal =',ip(k)
49    stop
50  end if
51  a(ip(k),ip(k))=sqrt(piv)
52  ipk=ip(k)
53  do i=k+1,n
54    piv=ip(i)
55    a(ipiv,ipk)=a(ipiv,ipk)/a(ipk,ipk)
56    a(ipk,ipiv)=a(ipiv,ipk)
57    do j=k+1,i
58      jpiv=ip(j)
59      a(ipiv,jpiv)=a(ipiv,jpiv)-a(ipiv,ipk)*a(ipk,jpiv)
60      a(jpiv,ipiv)=a(ipiv,jpiv)
61    end do
62  end do
63  end do
64  piv=a(ip(n),ip(n))
65  det=det*piv
66  !diagonal
67  if(piv<1.e-16) then
68    write(*,*)"no es definida positiva , piv=",piv,"para etapa=",n
69    stop
70  end if
71  a(ip(n),ip(n))=sqrt(piv)
72  end subroutine

```

## Capítulo 3

# Factorización QR

En este capítulo veremos como usar el método de Householder para calcular la factorización  $QR$  de una matriz, siendo  $Q$  una matriz unitaria y  $R$  una matriz triangular superior. Este método reduce un sistema a una matriz triangular utilizando matrices unitarias, lo que tiene especial interés en la resolución de problemas de mínimos cuadrados. La factorización  $QR$  es el método que se prefiere para resolver sistema sobredeterminados en el sentido de mínimos cuadrados, es por eso que en esta sección se trabajará con matrices rectangulares. Gracias a las propiedades de las matrices unitarias, el condicionamiento de la matriz  $R$  es igual al de la matriz original, lo cual nos conduce a métodos numéricos más estables. Aun así, para problemas mal condicionados o si se desea calcular el rango hay que utilizar una estrategia de pivoto por columnas.

### 3.1. Factorización QR

Comenzaremos definiendo lo que es una matriz unitaria (u ortogonal en el caso real) y recordamos algunas de sus propiedades.

**Definición 3.1.** Sea  $Q \in \mathcal{M}_{n \times n}$  una matriz, diremos que es ortogonal si  $QQ^T = Q^TQ = I$ .

**Definición 3.2.** Sea  $U \in \mathcal{M}_{n \times n}$  una matriz, diremos que es unitaria si  $UU^* = U^*U = I$ .

**Proposición 3.3.** Sea  $Q$  una matriz unitaria (respectivamente ortogonal).

1.  $\det(Q) = \pm 1$ .
2.  $\|Qv\|_2 = \|v\|_2$ .
3.  $\|Q\|_2 = 1$ .
4.  $\|A\|_2 = \|QA\|_2 = \|AQ\|_2 = \|Q^T A Q\|_2$ .

$$5. \text{cond}_2(Q) = 1.$$

$$6. \text{cond}_2(QA) = \text{cond}_2(A) = \text{cond}_2(AQ).$$

*Demostración.* 1.  $\det Q \det Q = \det Q \det Q^* = \det Q \det Q^{-1} = 1$  por lo que  $\det(Q) = \pm 1$

$$2. \|Qv\|_2^2 = v^* Q^* Q v = v^* v = \|v\|_2^2$$

$$3. \|Q\|_2^2 = \rho(Q^* Q) = \rho(I) = 1.$$

4.  $\|QA\|_2^2 = \rho(A^* Q^* Q A) = \rho(A^* A) = \|A\|_2^2$  de manera similar tenemos  $\|AQ\|_2^2 = \rho(Q^* A^* A Q) = \rho(A^* A) = \|A\|_2^2$ , pues  $Q^* A^* A Q$  y  $A^* A$  son matrices semejantes y por lo tanto tienen los mismos autovalores.

$$5. \text{cond}_2(Q) = \|Q\|_2 \|Q^{-1}\|_2 = 1.$$

$$6. \text{cond}_2(QA) = \|QA\|_2 \|A^{-1} Q^{-1}\|_2 = \|A\|_2 \|A^{-1}\|_2 = \text{cond}_2(A).$$

□

A continuación definiremos lo que es una matriz de Householder que utilizaremos para realizar la factorización QR.

**Definición 3.4.** Sea  $v$  un vector no nulo de  $\mathbb{C}^n$ , llamamos matriz de Householder a una matriz de la forma  $H(v) = I - 2 \frac{vv^*}{v^*v}$

**Proposición 3.5.** Sea  $H$  una matriz de Householder, se verifica que:

1.  $H$  es hermitiana (simétrica).

2.  $H$  es unitaria (ortogonal).

*Demostración.* 1. Se obtiene a partir de la definición pues  $vv^*$  es hermitiana.

$$2. H^* H = \left( I - 2 \frac{vv^*}{v^*v} \right) \left( I - 2 \frac{vv^*}{v^*v} \right) = I - 4 \frac{vv^*}{v^*v} + 4 \frac{vv^* vv^*}{(v^*v)^2} = I.$$

□

*Observación 3.6.* Dada una matriz de Householder  $H(v)$  y un vector  $w$  entonces para calcular su producto no es necesario calcular explícitamente  $H(v)$  ya que

$$H(v)w = Iw - 2 \frac{v(v^*w)}{v^*v}.$$

El coste de esta operación es  $\mathcal{O}(4n)$  si  $v$  es un vector unitario en norma dos.

En el siguiente resultado se ha tomado la construcción de [7] por ser el que utiliza aritmética compleja.

**Proposición 3.7.** *Sea  $x \in \mathbb{C}^n$  tal que  $\sum_{i=2}^n |x_i| > 0$ . Existen dos matrices de Householder tales que las  $n - 1$  últimas componentes del vector  $Hx$  son nulas. Más precisamente, sea  $x_1 = |x_1|e^{i\alpha}$ , con  $\alpha \in (-\pi, \pi]$ . Entonces*

$$1. H(x + \|x\|_2 e^{i\alpha} \mathbf{e}_n^1)x = -\|x\|_2 e^{i\alpha} \mathbf{e}_n^1.$$

$$2. H(x - \|x\|_2 e^{i\alpha} \mathbf{e}_n^1)x = +\|x\|_2 e^{i\alpha} \mathbf{e}_n^1.$$

Se denota por  $\mathbf{e}_n^1$  el primer vector de la base canónica de  $\mathbb{C}^n$ .

*Demostración.* Como  $\sum_{i=2}^n |x_i| > 0$  los vectores  $x \pm \|x\|_2 e^{i\alpha} \mathbf{e}_n^1$  son no nulos, por lo que las correspondientes matrices de Householder están bien definidas. Realizando el producto:

$$\begin{aligned} H(x + \|x\|_2 e^{i\alpha} \mathbf{e}_n^1)x &= x - \frac{2}{\beta} \left( x + \|x\|_2 e^{i\alpha} \mathbf{e}_n^1 \right) \left( x + \|x\|_2 e^{i\alpha} \mathbf{e}_n^1 \right)^* x = \\ &= x - \frac{2}{\beta} \left( \|x\|_2^2 + \|x\|_2 |x_1| \right) \left( x + \|x\|_2 e^{i\alpha} \mathbf{e}_n^1 \right) = x - \left( x + \|x\|_2 e^{i\alpha} \mathbf{e}_n^1 \right) = -\|x\|_2 e^{i\alpha} \mathbf{e}_n^1 \end{aligned}$$

siendo  $\beta = \|x + \|x\|_2 e^{i\alpha} \mathbf{e}_n^1\|_2^2 = 2(\|x\|_2^2 + |x_1| \|x\|_2)$ .

En efecto,

$$\begin{aligned} \beta &= \|x + \|x\|_2 e^{i\alpha} \mathbf{e}_n^1\|_2^2 = \left( x + \|x\|_2 e^{i\alpha} \mathbf{e}_n^1 \right)^* \left( x + \|x\|_2 e^{i\alpha} \mathbf{e}_n^1 \right) = \\ &= \|x\|_2^2 + \bar{x}_1 \|x\|_2 e^{i\alpha} + \|x\|_2^2 + x_1 \|x\|_2 e^{-i\alpha} = 2 \left( \|x\|_2^2 + |x_1| \|x\|_2 \right) \in \mathbb{R} \end{aligned}$$

y de manera análoga se obtendría 2. □

*Observación 3.8.* Para reducir los errores de redondeo en el cálculo se elige la opción 1 de la proposición anterior si  $x_1 > 0$  y la opción 2 si  $x_1 < 0$ .

*Observación 3.9.* Si  $\sum_{i=2}^n |x_i| = 0$  se puede tomar la identidad como matriz de Householder, si  $x_1 > 0$  entonces  $Ix = \|x\|_2 \mathbf{e}_n^1 = x_1 \mathbf{e}_n^1$  o  $H(2x_1 \mathbf{e}_n^1)x = \|x\|_2 \mathbf{e}_n^1 = -x_1 \mathbf{e}_n^1$  si  $x_1$  es negativo.

La proposición anterior va a permitir reducir una matriz a forma triangular superior multiplicando por matrices de Householder. Sea  $A^{(k-1)}$  una matriz que tenga ceros por debajo de la diagonal en las  $k-1$  primeras columnas, se considera  $H(\tilde{v}^{(k)})$  la matriz de Householder que anula las  $m - k$  últimas componentes del vector  $\tilde{x}^{(k)} = (a_{kk}^{(k-1)}, \dots, a_{mk}^{(k-1)})$ :

$$H(\tilde{v}^{(k)}) = \pm \|\tilde{x}^{(k)}\|_2 \mathbf{e}_{m-k+1}^1.$$

$$\begin{aligned}
& \left( \begin{array}{c|c} I_{k-1} & 0 \\ \hline 0 & H(\tilde{v}^{(k)}) \end{array} \right) \left( \begin{array}{cccc|ccc} a_{11}^{(1)} & a_{12}^{(1)} & \cdots & a_{1k-1}^{(1)} & a_{1k}^{(1)} & a_{1k+1}^{(1)} & \cdots & a_{1n}^{(1)} \\ 0 & \ddots & & a_{2k-1}^{(2)} & a_{2k}^{(2)} & a_{2k+1}^{(2)} & \cdots & a_{2n}^{(2)} \\ \vdots & & \ddots & \vdots & \vdots & \vdots & & \vdots \\ 0 & \cdots & 0 & a_{k-1k-1}^{(k-1)} & a_{k-1k}^{(k-1)} & a_{k-1k+1}^{(k-1)} & \cdots & a_{k-1n}^{(k-1)} \\ \hline 0 & \cdots & \cdots & 0 & a_{kk}^{(k-1)} & a_{kk+1}^{(k-1)} & \cdots & a_{kn}^{(k-1)} \\ 0 & \cdots & \cdots & 0 & a_{k+1k}^{(k-1)} & a_{k+1k+1}^{(k-1)} & \cdots & a_{k+1n}^{(k-1)} \\ \vdots & & & \vdots & \vdots & \vdots & & \vdots \\ 0 & \cdots & \cdots & 0 & a_{mk}^{(k-1)} & a_{mk+1}^{(k-1)} & \cdots & a_{mn}^{(k-1)} \end{array} \right) = \\
& = \left( \begin{array}{cccc|ccc} a_{11}^{(1)} & a_{12}^{(1)} & \cdots & a_{1k-1}^{(1)} & a_{1k}^{(1)} & a_{1k+1}^{(1)} & \cdots & a_{1n}^{(1)} \\ 0 & \ddots & & a_{2k-1}^{(2)} & a_{2k}^{(2)} & a_{2k+1}^{(2)} & \cdots & a_{2n}^{(2)} \\ \vdots & & \ddots & \vdots & \vdots & \vdots & & \vdots \\ 0 & \cdots & 0 & a_{k-1k-1}^{(k-1)} & a_{k-1k}^{(k-1)} & a_{k-1k+1}^{(k-1)} & \cdots & a_{k-1n}^{(k-1)} \\ \hline 0 & \cdots & \cdots & 0 & \|v^{(k)}\|_2 & a_{kk+1}^{(k)} & \cdots & a_{kn}^{(k)} \\ 0 & \cdots & \cdots & 0 & 0 & a_{k+1k+1}^{(k)} & \cdots & a_{k+1n}^{(k)} \\ \vdots & & & \vdots & \vdots & \vdots & & \vdots \\ 0 & \cdots & \cdots & 0 & 0 & a_{mk+1}^{(k)} & \cdots & a_{mn}^{(k)} \end{array} \right).
\end{aligned}$$

Se da a continuación un teorema de existencia de la factorización  $QR$  que se puede encontrar en [3].

**Teorema 3.10.** *Sea una matriz  $A \in \mathcal{M}_{m \times n}$ , entonces existe una matriz unitaria  $Q \in \mathcal{M}_{m \times m}$  y una matriz triangular superior  $R \in \mathcal{M}_{m \times n}$  tales que  $A = QR$ .*

*Demostración.* En la etapa 1 si  $\|(a_{21}, \dots, a_{m1})\|_2 \neq 0$  existe una matriz de Householder  $Q_1$  tal que  $Q_1 A^{(0)}$  todos los elementos subdiagonales de la primera columna son nulos. En caso contrario se toma  $Q_1 = I$  y se pasa a la etapa 2. De forma general, en la etapa  $k$  si se tiene que si  $\|(a_{k+1k}, \dots, a_{mk})\|_2 \neq 0$  existe una matriz  $Q_k$  que reduce a cero todos los elementos subdiagonales en la columna  $k$  de  $A^{(k-1)}$ . Por tanto después de  $n$  etapas si  $m > n$ , (o  $n - 1$  etapas si  $m = n$ ) llegamos a:

$$Q_n \cdots Q_1 A^{(0)} = R \quad (\text{respectivamente } Q_{n-1} \cdots Q_1 A^{(0)} = R)$$

siendo  $R$  una matriz triangular superior. Dado que las matrices  $Q_1, \dots, Q_n$  son unitarias tenemos que

$$A^{(0)} = Q_1^* \cdots Q_n^* R \quad (\text{respectivamente } A^{(0)} = Q_1^* \cdots Q_{n-1}^* R).$$

Definiendo  $Q := Q_1^* \cdots Q_n^*$  (respectivamente  $Q := Q_1^* \cdots Q_{n-1}^*$ ) obtenemos la factorización  $A = QR$ , donde  $Q$  es unitaria por ser el producto de matrices ortogonales.  $\square$

*Observación 3.11.* En [1] se demuestra la unicidad de factorización para una matriz  $R$  cuadrada e invertible bajo la condición  $r_{ii} > 0, i = 1, \dots, n$ .

*Observación 3.12.* El coste de la factorización es  $\mathcal{O}(2n^2 \frac{3m-n}{3})$  operaciones aritméticas (véase [3]).

*Observación 3.13.* A diferencia de la factorización  $LU$ , Crout o Cholesky, en la factorización  $QR$  sólo se construyen explícitamente los coeficientes de la matriz  $R$ . La matriz  $Q$  está almacenada a través de los vectores que definen cada una de las matrices de Householder. En la implementación que hemos realizado, la matriz  $R$  ocupa la parte triangular superior, las primeras componentes de los vectores de Householder se almacena en un vector adicional, y las restantes en la matriz de partida por debajo de la diagonal. Es decir, el espacio necesario para almacenarlo es  $nm + n$ .

*Observación 3.14.* El cálculo del determinante a partir de la factorización  $QR$  puede hacerse de la siguiente manera

$$\det(A) = \det(R) \cdot (-1)^{n_f}$$

siendo  $n_f$  el número de transformaciones de Householder no triviales.

Dado que  $A = QR$  la resolución de un sistema lineal  $Ax = b$  se calcula resolviendo los siguientes sistemas

$$Ax = b \iff QRx = b \iff \begin{cases} Qv = b & \iff v = Q^*b \\ Rx = v \end{cases}$$

*Observación 3.15.* Dado que la matriz  $Q$  es unitaria, el primer sistema se resuelve multiplicando directamente por  $Q^{-1} = Q^*$ . El coste es de  $\mathcal{O}(2n^2)$  operaciones aritméticas.

*Observación 3.16.* Puesto que  $m > n$  podría ocurrir que el sistema fuese incompatible, la factorización  $QR$  permite resolver el sistema en el sentido de mínimos cuadrados, esto es calcula el  $\hat{x}$  solución del problema de minimización  $\min_{x \in \mathbb{R}^n} \|Ax - b\|_2$  (veanse [2] y [3]).

## 3.2. Factorización QR con pivote

La factorización  $QR$  a diferencia de la  $LU$  no se bloquea con un coeficiente diagonal nulo. Aún en el caso de que  $(a_{kk} \dots a_{mk}) = \mathcal{O}$ , bastaría con tomar como matriz de Householder la identidad. Sin embargo la matriz  $R$  obtenida sería singular lo que podría

plantear problemas en la resolución del sistema correspondiente. Como se detalla en [3] esto no impide que el problema de resolución de mínimos cuadrados tenga solución. La manera eficiente de calcularla es utilizar una estrategia de pivote por columnas en donde se van seleccionando las columnas de módulo máximo.  $AP^t = QR$  con  $A \in M_{m \times n}$ . Se elige una permutación de columnas tal que en cada etapa la norma del vector reducido sea máxima. En cada etapa  $k$  se elige la columna  $\tau(k)$  tal que

$$\left\| \left( a_{k\tau(k)}^{(k-1)}, \dots, a_{m\tau(k)}^{(k-1)} \right) \right\|_2 = \max_{j=k, \dots, n} \left\| \left( a_{k\tau(j)}^{(k-1)}, \dots, a_{m\tau(j)}^{(k-1)} \right) \right\|_2.$$

El resultado de factorización se puede enunciar entonces en los siguientes términos.

**Teorema 3.17.** *Sea  $A \in M_{m \times n}$  una matriz de rango  $l$  entonces existe una matriz de permutación  $P \in M_{n \times n}$ , una matriz ortogonal  $Q \in M_{m \times m}$  y una matriz triangular superior  $R \in M_{m \times n}$  tales que  $AP^T = QR$ .*

*Demostración.* Etapa 1: Se elige  $\tau(1)$  la columna de normas dos máxima, si esta norma es cero es porque la matriz es cero, y por tanto de rango cero. Sea  $P_1^T$  la matriz permutación que lleva esa columna a la posición 1 y  $Q_1$  la matriz de Householder que reduce a cero todos los coeficiente por debajo de la diagonal:  $Q_1 A^{(0)} P_1^T = A^{(1)}$  es una matriz que tiene ceros en la primera columna por debajo de la diagonal. En la etapa  $k$  se consideran los vectores de las columnas correspondientes desde la fila  $k$ -ésima hasta la fila  $m$ -ésima; se elige como  $\tau(k)$ , la que tiene norma dos máxima; sea  $P_k^T$  la permutación de columnas que la lleva a la posición  $k$  y  $Q_k$  la matriz que anula los coeficientes por debajo de la diagonal. Si no fuese posible hallar una columna de norma distinta de cero es porque el rango de la matriz es  $k - 1$ . Cuando se hayan completado  $l$  etapas, siendo  $l$  el rango de la matriz, se llega a  $Q_l \cdots Q_1 A P_1 \cdots P_l = R$ . Por tanto como las matrices de Householder son unitarias y hermitianas por la proposición 3.5 entonces  $Q_i^{-1} = Q_i^* = Q_i$

$$A \overbrace{P_1^T \cdots P_l^T}^{P^T} = \overbrace{Q_1 \cdots Q_l}^Q R$$

Donde  $Q$  es una matriz unitaria por ser producto de matrices unitarias.

□

*Observación 3.18.* La matriz  $R$  estaría almacenada en

$$\left( \begin{array}{cccc|cccc} a_{1\tau(1)}^{(1)} & a_{1\tau(2)}^{(1)} & \cdots & a_{1\tau(l)}^{(1)} & a_{1\tau(l+1)}^{(1)} & a_{1\tau(l+2)}^{(1)} & \cdots & a_{1\tau(n)}^{(1)} \\ 0 & \ddots & & a_{2\tau(l)}^{(2)} & a_{2\tau(l+1)}^{(2)} & a_{2\tau(l+2)}^{(2)} & \cdots & a_{2\tau(n)}^{(2)} \\ \vdots & & \ddots & \vdots & \vdots & \vdots & & \vdots \\ 0 & \cdots & 0 & a_{l\tau(l)}^{(l)} & a_{l\tau(l+1)}^{(l)} & a_{l\tau(l+2)}^{(l)} & \cdots & a_{l\tau(n)}^{(l)} \\ \hline 0 & \cdots & \cdots & 0 & 0 & 0 & \cdots & 0 \\ \vdots & & & \vdots & 0 & 0 & \cdots & 0 \\ \vdots & & & \vdots & \vdots & \vdots & & \vdots \\ 0 & \cdots & \cdots & 0 & 0 & 0 & \cdots & 0 \end{array} \right) \quad (3.1)$$

los  $l$  vectores de Householder estarían almacenados en las  $l$  primeras columnas por debajo de la diagonal, porque solo se han hecho  $l$  transformaciones y en un vector adicional.

*Observación 3.19.* El cálculo del determinante de una matriz cuadrada a partir de la factorización  $QR$  puede hacerse de la siguiente manera

$$\det(A) = \det(Q) \det(R) \det(P) = \det(R) \cdot (-1)^{n_f + n_c}$$

siendo  $n_f$  el número de transformaciones de Householder no triviales, y  $n_c$  el número de permutaciones de columnas.

*Observación 3.20.* Siguiendo a [3] no se han recalculado las normas dos de las columnas en cada etapa, sino que para la etapa  $k$  se han utilizado los valores de la etapa  $k - 1$  eliminando la contribución correspondiente a la fila  $k - 1$ :

$$\|(a_{k\tau(j)}^{(k-1)}, \dots, a_{m\tau(j)}^{(k-1)})\|_2^2 = \|(a_{k-1\tau(j)}^{(k-1)}, a_{k\tau(j)}^{(k-1)}, \dots, a_{m\tau(j)}^{(k-1)})\|_2^2 - |a_{k-1\tau(j)}^{(k-1)}|^2$$

para  $j = k, \dots, n$ . Esto permite reducir el coste de la búsqueda de pivote de  $\mathcal{O}(2mn^2)$  a  $\mathcal{O}(2mn)$  operaciones aritméticas. Además en ambos casos habría que hacer  $\mathcal{O}(\frac{n^2}{2})$  comparaciones.

Dado que  $A = QRP$  hallar la solución del sistema original es equivalente a resolver los siguientes sistemas

$$Ax = b \iff QR Px = b \iff \begin{cases} Qv = b & \iff v = Q^*b \\ Rw = v \\ Px = w, \end{cases}$$

donde el primer sistema se resuelve multiplicando por la inversa, y el tercero es una permutación,  $x_{\tau(i)} = w_i$ .

## 3.2.1. Código Fortran 90

```

1  subroutine qr_ls_piv(a,d,iq)
2  !*****
3  !factorizacion qr de A (metodo de Householder)
4  !ENTRADA:
5  ! a   matriz
6  !SALIDA:
7  ! a   (sup) matriz r
8  ! a   (inf) matriz de vectores de householder
9  ! d   vector de elementos diagonales de w
10 ! iq  permutacion de columnas
11 !*****
12 !*
13 use mod_creal
14 implicit none
15 real(kind=creal),dimension(:, :),intent(inout)::a
16 real(kind=creal),dimension(:),intent(out)::d
17 real(kind=creal),dimension(:),allocatable::sn
18 integer,intent(out)::iq(:)
19 real(kind=creal)::s,coef,beta
20 integer::i,j,k,n,m,jpiv,ind
21
22 n=size(a,2)
23 m=size(a,1)
24 allocate(sn(n))
25 do i=1,n
26   iq(i)=i
27   sn(i)=dot_product(a(1:m,i),a(1:m,i))
28 end do
29 print*, 'normas=',sn(1:n)
30 do k=1,min(m,n+1)-1
31 !busqueda de pivote (columna reducida de mayor modulo)
32   jpiv=k
33   s=sn(iq(k))
34   do j=k+1,n
35     if(sn(iq(j)).gt.s) then
36       jpiv=j
37       s=sn(iq(j))
38     end if
39   end do
40 !permutacion
41   if(jpiv.ne.k) then
42     ind=iq(jpiv)
43     iq(jpiv)=iq(k)

```

```

44     iq(k)=ind
45     end if
46     s=sn(iq(k))
47     !construccion de la matriz de Householder
48     if(s.lt.1.d-40) then
49         rite(6,*) 'rango de la matriz ',k-1
50         stop
51     else if(a(k,iq(k)).gt.0.d0) then
52         d(k)=-sqrt(s)
53     else
54         d(k)=sqrt(s)
55     end if
56     beta = 1.d0/(s-d(k)*a(k,iq(k)))
57     a(k,iq(k))=a(k,iq(k))-d(k)
58     !multiplicacion de columnas
59     do j=k+1,n
60         coef=0.d0
61         do i=k,m
62             coef=coef + a(i,iq(k))*a(i,iq(j))
63         end do
64         coef=coef*beta
65         a(k:m,iq(j))=a(k:m,iq(j))-coef*a(k:m,iq(k))
66     end do
67     print*, 'norma=',d(k), ' primera componente ',a(k,iq(k))
68     !trasvase de d(k) a a(k,k)
69     s=d(k)
70     d(k)=a(k,iq(k))
71     a(k,iq(k))=s
72     !actualizacion de sn
73     do j=k+1,m
74         sn(iq(j))=sn(iq(j))-a(k,iq(j))*a(k,iq(j))
75     end do
76     print*, 'normas=',sn(1:n)
77 end do
78 print*, 'iq=',iq(1:n)
79 return
80 end

```



## Capítulo 4

# Aplicaciones

Además de la resolución de sistemas de ecuaciones lineales que ya vimos en los capítulos anteriores, recordamos a continuación otras aplicaciones de los métodos descritos en esta memoria.

### 4.1. Cálculo de determinantes

Los métodos descritos en este trabajo pueden usarse para calcular el determinante de una matriz  $A \in M_{n \times n}$  de manera eficiente. El cálculo de un determinante por la regla de Cramer es del orden de aproximadamente  $\mathcal{O}((n+1)!)$  operaciones aritméticas (vease [1]), mientras que el cálculo del determinante de una matriz triangular se reduce a tan solo  $\mathcal{O}(n-1)$  operaciones. Por tanto si se hace mediante factorización el coste del cálculo del determinante se reduce al coste de la factorización misma. Las fórmulas ya se han dado en cada una de las secciones.

### 4.2. Cálculo del rango

La factorizaciones  $LU$  con estrategia de pivote total o  $QR$  con estrategia de pivote por columnas pueden usarse para calcular el rango de una matriz, ya que si es  $l \leq n$  ambos procedimientos terminan al completar esa etapa, porque el resto de coeficientes son cero. Esto nos puede servir para detectar en particular matrices singulares.

### 4.3. Cálculo de inversas

Dada una matriz  $A \in M_{m \times n}$  el cálculo de su inversa consiste en encontrar la matriz  $A^{-1}$  tal que  $AA^{-1} = I$ . Esta ecuación podemos expresarla de la siguiente manera

$$A \left( \begin{array}{c|c|c|c} \mathbf{C}^1 & \mathbf{C}^2 & \dots & \mathbf{C}^n \end{array} \right) = \left( \begin{array}{c|c|c|c} \mathbf{e}^1 & \mathbf{e}^2 & \dots & \mathbf{e}^n \end{array} \right) \quad (4.1)$$

donde  $\mathbf{C}^i$  denota la columna  $i$ -ésima de la matriz  $A^{-1}$  y  $\mathbf{e}^i$  el  $i$ -ésimo vector de la base canónica de  $\mathbb{R}^n$ . Por tanto el cálculo de la inversa es equivalente a la resolución de los sistemas

$$A\mathbf{C}^i = \mathbf{e}^i \quad i = 1, \dots, n$$

Por tanto el cálculo de la inversa de  $A$  tendrá un coste dado por el coste de la factorización de  $A$  más el coste de la resolución de  $2n$  sistemas triangulares (en la factorización  $LU$  o Cholesky) y  $n$  sistemas triangulares más  $n$  multiplicaciones de vector por una matriz ortogonal producto de matrices de Householder (en el caso de  $QR$ ) cada uno con un coste de  $\mathcal{O}(n^2)$  operaciones. La principal ventaja de los métodos estudiados en esta memoria es que en el proceso de factorización se puede detectar si una matriz es singular. Esto ocurre con las factorizaciones  $LU$  con pivote, y con las factorizaciones  $QR$  de matrices cuadradas. Sin embargo la factorización de Cholesky puede bloquearse con una matriz invertible cuando no es definida positiva, esto puede ocurrir tanto con como sin estrategia de pivote.

# Conclusiones y futuro trabajo

Se ha elaborado una biblioteca de programas en Fortran 90 que implementa las estrategias de pivote asociadas a las factorizaciones  $LU$ , Cholesky y  $QR$ . Estos métodos permiten resolver de manera eficiente sistemas mal condicionados (tanto en el sentido clásico como en el de mínimos cuadrados) y minimizar los efectos de acumulación del error de redondeo. Los contenidos de esta memoria extienden los métodos vistos en la asignatura de Análisis Numérico Matricial del Grado de Matemáticas.

Este trabajo podría completarse con los métodos para matrices hermitianas no definidas derivados de la factorización de Crout y con estrategias especiales de búsqueda de pivote para matrices dispersas.



# Anexo: Código para la resolución de los sistemas factorizados

## Factorización $LU$ con pivote total

```
1 subroutine sistlupfc(a,v,ip,iq)
2 !resuelve los sistemas triangulares superior e inferior resultantes
3 !de la factorizacion LU con pivote total
4 !entrada
5 !ip  permutacion por filas
6 !iq  permutacion por columnas
7 !a   matriz
8 !v   termino independiente
9 !salida
10 !v   solucion
11
12 use mod_creal
13 implicit none
14 integer, intent(in), dimension(:):: ip, iq
15 real(kind=creal), intent(inout), dimension(:):: v
16 real(kind=creal), intent(in), dimension(:,):: a
17 real(kind=creal), allocatable, dimension(:):: w
18 real(kind=creal):: s
19 integer:: n, i, j
20
21 n=size(a,1)
22 allocate(w(n))
23 !superior
24 w(1) = v(ip(1))
25 do i=2,n
26   s=0.
27   do j=1,i-1
28     s=s + a(ip(i),iq(j))*w(j)
29   end do
```

```

30  w(i)= v(ip(i))-s
31  end do
32  !inferior
33  v(iq(n)) = w(n)/a(ip(n), iq(n))
34  do i=n-1,1,-1
35    s=0.
36    do j=i+1,n
37      s=s + a(ip(i), iq(j))*v(iq(j))
38    end do
39    v(iq(i))= (w(i)-s)/a(ip(i), iq(i))
40  end do
41  !libera memoria
42  deallocate(w)
43  end subroutine sistlupfc

```

## Factorización de Cholesky con pivote

```

1  subroutine sistlltpfc(a,v,iq)
2  !resuelve los sistemas triangulares superior e inferior resultantes de
3  !la factorizacion de Cholesky con pivote
4  !entrada
5  !iq  permutacion por filas
6  !a   matriz
7  !v   termino independiente
8  !salida
9  !v   solucion
10
11  use mod_creal
12  implicit none
13  integer , intent(in) , dimension(:):: iq
14  real(kind=creal) , intent(inout) , dimension(:):: v
15  real(kind=creal) , intent(in) , dimension(:,):: a
16  real(kind=creal) , allocatable , dimension(:):: w
17  real(kind=creal):: s
18  integer :: n, i, j
19
20  n=size(a,1)
21  allocate(w(n))
22  !superior
23  w(1) = v(iq(1))/a(iq(1), iq(1))
24  do i=2,n
25    s=0.
26    do j=1,i-1

```

```

27     s=s + a(iq(i),iq(j))*w(j)
28   end do
29   w(i)= (v(iq(i))-s)/a(iq(i),iq(i))
30 end do
31 !inferior
32 v(iq(n)) = w(n)/a(iq(n),iq(n))
33 do i=n-1,1,-1
34   s=0.
35   do j=i+1,n
36     s=s + a(iq(i),iq(j))*v(iq(j))
37   end do
38   v(iq(i))= (w(i)-s)/a(iq(i),iq(i))
39 end do
40 !libera memoria
41 deallocate(w)
42 end subroutine

```

## Factorización $QR$ con pivote

```

1  subroutine sistqr_piv(a,d,b,iq)
2  !resolucion de los sistemas  $QRx=b$  resultantes de la
3  !factorizacion QR
4  !entrada
5  !a  matriz
6  !d  prinera componente de los vectores de Householder
7  !iq permutacion de columnas
8  !b  termino independiente
9  !salida
10 !b  solucion
11
12 use mod_sistu_q
13 use mod_creal
14 implicit none
15 real(kind=creal)::a(:, :), d(:), b(:), coef, rnor
16 integer::nmin,m,n,k,i,iq(:)
17
18 m=size(a,1)
19 n=size(a,2)
20 nmin=min(m,n+1)-1
21 !multiplicacion por  $Q^{-1}$ 
22 do k=1,nmin
23   coef=d(k)*b(k)
24   rnor=d(k)*d(k)

```

```

25  do i=k+1,m
26    coef=coef + a(i,iq(k))*b(i)
27    rnor=rnor + a(i,iq(k))*a(i,iq(k))
28  end do
29  coef=coef*2/rnor
30  b(k)=b(k)-coef*d(k)
31  b(k+1:m)=b(k+1:m) - coef*a(k+1:m,iq(k))
32 end do
33 !sistema R
34 call sistu_q(a,b,iq,d)
35 b=d
36 return
37 end
38
39 subroutine sistu_q(a,b,iq,z)
40 !resolucion de un sistema triangular superior
41 !entrada:
42 !a  matriz de coeficientes almacenada parte superior (i<=j)
43 !b  termino independiente
44 !iq permutacion de columnas
45 !salida:
46 !z  solucion
47
48 use mod_creal
49 implicit none
50 real(kind=creal),intent(in)::a(:,:),b(:)
51 integer,intent(in)::iq(:)
52 real(kind=creal),intent(out)::z(:)
53 integer::n,i
54
55 n=size(a,2)
56 z(iq(1:n))=b(1:n)
57 do i=n,2,-1
58   z(iq(i))=z(iq(i))/a(i,iq(i))
59   z(iq(1:i-1))=z(iq(1:i-1))-a(1:i-1,iq(i))*z(iq(i))
60 end do
61 z(iq(1))=z(iq(1))/a(1,iq(1))
62 end subroutine sistu_q

```

# Bibliografía

- [1] Ciarlet, P. G., *Introducción á análise numérica matricial e á optimización*, Servicio de publicacións da Universidade de Santiago de Compostela, 1999.
- [2] Gander, W. Gander, M. J. and Kwok, F., *Scientific Computing*, An Introduction using Maple and MATLAB, Springer, 2014.
- [3] Golub G. H. and Van Loan C. H., *Matrix Computations*, The Jhon Hopkins University Press, 2013.
- [4] Isaacson, E. and Keller, H. B., *Analysis of Numerical Methods*, 1966.
- [5] Kielbasinski A., *Linear Algebra and its applications* 88/89 487-494, 1987.
- [6] Kincaid D. R., Cheney E.W., *Numerical Analysis*, 1994.
- [7] Stoer, J. and Bulirsch, R., *Introduction to Numerical Analysis*, Springer-Verlag, 1993.
- [8] Wilkinson J. H., *The Algebraic Eigenvalue Problem*, Oxford University Press, 1965.