Universidad de La Laguna

SECCIÓN DE MATEMÁTICAS | FACULTAD DE CIENCIAS

Cristina González Marrero

Álgebra Lineal Numérica

Numerical Linear Algebra

Trabajo Fin de Grado
Grado en Matemáticas
La Laguna, Septiembre de 2018

DIRIGIDO POR
Ruymán Cruz Barroso

Ruymán Cruz Barroso
Departamento de Análisis
Matemático
Universidad de La Laguna
38271 La Laguna, Tenerife

Agradecimientos

A Ruymán Cruz Barroso por su dedicación para la elaboración de la presente Memoria. A mis amigos y en especial a mi familia, su ayuda siempre ha sido un soporte fundamental para mi formación.

Resumen · Abstract

Resumen

Este trabajo tiene por objetivo el estudio de manera más profunda de algunos de los contenidos sobre Álgebra Lineal Numérica *estudiados durante el Grado en Matemáticas. Se abordará un análisis sobre matrices ortogonales, ciertas factorizaciones matriciales de suma importancia, así como el estudio de las reflexiones de Householder y las rotaciones de Givens, con el propósito de obtener las factorización QR de una matriz dada. Como aplicación, se abordará la resolución numérica de sistemas lineales de ecuaciones y el cálculo numérico de autovalores y autovectores mediante el algoritmo QR.*

Palabras clave: *Factorizaciones matriciales – reflexiones de Householder – rotaciones de Givens – factorización QR – sistemas lineales de ecuaciones – algoritmo QR.*

Abstract

The aim of this Project is to delve in some of the contents on Numerical Linear Algebra *that were already studied during the Degree of Mathematics. An analysis of orthogonal matrices, certain matrix factorizations of special interest, along with the study of Householder reflections and Givens rotations, will be carried out with the aim of to obtain the QR factorization of a given matrix. As an application, the solution of linear systems of equations and the numerical calculus of eigenvalues and eigenvectors by means of the QR algorithm will be analyzed.*

Keywords: *Matrix factorizations – Householder reflections – Givens rotations – QR factorization – linear systems of equations – QR algorithm.*

Contenido

Introducción

Esta Memoria tiene por objetivo el profundizar en algunos de los contenidos sobre Álgebra Lineal Numérica que fueron estudiados en la asignatura *Métodos Numéricos I* del Grado en Matemáticas. Se ha estructurado en cuatro capítulos. En el primero de ellos se abordan cuestiones elementales sobre vectores y matrices que serán fundamentales para el desarrollo del trabajo. No obstante, este capítulo introductorio contiene algunos resultados básicos sobre Álgebra Lineal que tampoco fueron estudiados durante el Grado, en particular, abordamos el estudio de matrices ortogonales, algunas factorizaciones matriciales de importancia, como es el caso de la factorización de Schur, y la descomposición espectral de una matriz. En el segundo capítulo se introducen algunas nuevas factorizaciones matriciales generales, como son las trasformaciones ortogonales, geométricas, de reflexión (Householder) y de rotación (Givens). Presentamos así a continuación otras dos factorizaciones adicionales de especial relevancia dentro del Álgebra Lineal Numérica: las factorizaciones LU y QR. Todas estas transformaciones tienen numerosas aplicaciones, especialmente por el hecho de que una matriz dada que representa alguna transformación de un vector, es transformada de manera que permita determinar un vector a partir de otro dado. Los dos ejemplos más ilustrativos sobre esto son abordados en los dos capítulos restantes. En el tercer capítulo se estudia el problema de resolver numéricamente un sistema lineal de ecuaciones, poniendo especial énfasis en el método de sobre-relajación sucesiva, al no haber sido estudiado durante el Grado en Matemáticas. Por la misma razón, en el cuarto y último capítulo se introduce brevemente el problema del cálculo numérico de los autovalores y autovectores de una matriz mediante el algoritmo QR. La Memoria contiene un Apéndice donde se incluyen dos códigos implementados en lenguaje Matlab: el primero permite obtener la factorización QR de una matriz dada mientras que el segundo consiste en una codificación del algoritmo QR.

1

Resultados preliminares sobre Álgebra Lineal

Este primer capítulo tiene como objetivo el establecer algunos resultados preliminares sobre vectores y matrices que serán necesarios para el desarrollo de esta Memoria. El capítulo contiene definiciones y resultados que son ampliamente conocidos, pero también contiene otras cuestiones básicas sobre Álgebra Lineal que serán esenciales para abordar este trabajo sobre Álgebra Lineal Numérica, que no han sido estudiadas a lo largo del Grado en Matemáticas. En particular, introduciremos una serie de factorizaciones matriciales útiles muy conocidas en la literatura: la forma canónica equivalente, factorización de Schur, factorización canónica de semejanza y la descomposición espectral.

1.1. Vectores y matrices

Comenzamos esta Memoria introduciendo algo de notación y algunos resultados y conceptos necesarios. En lo que sigue, y salvo que se diga lo contrario, el vector x representará un vector columna de $\mathbb{R}^n$: $x = (x_1, \ldots, x_n)^T$. Las matrices serán denotadas por:

$$A = \begin{pmatrix} a_{11} & \ldots & a_{1m} \\ \vdots & \ddots & \vdots \\ a_{n1} & \ldots & a_{nm} \end{pmatrix} = (a_{ij}) \in \mathcal{M}_{n\times m}.$$

Sea V un espacio vectorial. Si cada vector $v \in V$ puede ser expresado como combinación lineal de vectores de algún conjunto G, entonces G recibe el nombre de *conjunto generador*. Si además, todas las combinaciones lineales de elementos de G están en V, el espacio vectorial V es el *espacio generado por* G. Al espacio generado por m vectores $x_1, \ldots, x_m$ lo denotamos por $\text{span}\{x_1, \ldots, x_m\}$. Recordar que una *base* de V es un conjunto de vectores de V linealmente independientes que forman un conjunto generador. Toda base B de V está formada por el mismo número de elementos, la dimensión de V : $dim(V) = Card(B)$. Tomando $V = \mathbb{R}^n$ definimos

$$\langle x, y\rangle := x^T y = y^T x = \sum_{i=1}^{n} x_i y_i$$

como el *producto interior* o *producto escalar euclídeo*, el cual es una aplicación desde el espacio vectorial $\mathbb{R}^n \times \mathbb{R}^n$ en $\mathbb{R}$ que cumple que $\forall x, y, z \in \mathbb{R}^n$ y $\forall a, b \in \mathbb{R}$:

- Definida positiva: $\langle x, x\rangle \geq 0$, $\langle x, x\rangle = 0 \Leftrightarrow x = 0$.
- Conmutatividad: $\langle x, y\rangle = \langle y, x\rangle$.
- Linealidad: $\langle ax + by, z\rangle = a\langle x, z\rangle + b\langle y, z\rangle$.

Nótese que si $V = \mathbb{C}^n$, entonces debemos cambiar la definición x^T por x^H, traspuesta Hermitiana, que consiste en cambiar filas por columnas y tomar conjugación, de modo que $\langle x, x\rangle = \sum_{i=1}^{n} |x_i|^2 \geq 0$. Por último, cabe mencionar una propiedad útil que verifica todo producto interior:

$$\langle x, y\rangle \leq \langle x, x\rangle^{1/2} \cdot \langle y, y\rangle^{1/2} \qquad \text{(Desigualdad de Cauchy-Schwarz)}.$$

Una *norma* es una función $\|\cdot\|$ definida de un espacio vectorial V en $\mathbb{R}$ que satisface $\forall x, y \in V$, $\forall a \in \mathbb{R}$ ($\mathbb{C}$) las siguientes condiciones:

1. Si $x \neq 0$, entonces $\|x\| > 0$, y $\|0\| = 0$.
2. $\|ax\| = |a| \cdot \|x\|$.
3. $\|x + y\| \leq \|x\| + \|y\|$ (Desigualdad triangular).

Un espacio vectorial en el que se ha definido una norma se denomina *espacio normado*: $(V, \|\cdot\|)$.

Tomando de nuevo $V = \mathbb{R}^n$, una clase muy frecuente de normas es la L_p-*norma (Norma Minkowski o Norma Hölder)*, para $p \geq 1$:

$$\|x\|_p = \Big(\sum_{i=1}^{n} |x_i|^p\Big)^{1/p}.$$

Las normas vectoriales más comunes son:

- $\|x\|_1 = \sum_{i=1}^{n} |x_i|$.
- $\|x\|_2 = \sqrt{\langle x, x\rangle} = \sqrt{\sum_{i=1}^{n} x_i^2}$, llamada *Norma Euclidea* o longitud del vector.
- $\|x\|_\infty = \max_{1\leq i\leq n} |x_i|$, llamada *Norma Máxima o Norma de Chebyshev*.

Es fácil comprobar tanto la relación $\|x\|_\infty = \lim_{p\to\infty} \|x\|_p$, como las siguientes desigualdades de las normas L_p de un vector $x \in \mathbb{R}^n$:

$$\|x\|_\infty \leq \|x\|_2 \leq \|x\|_1, \quad \|x\|_\infty \leq \|x\|_2 \leq \sqrt{n} \cdot \|x\|_\infty, \quad \|x\|_2 \leq \|x\|_1 \leq \sqrt{n} \cdot \|x\|_2.$$

Más generalmente, dado $x \in \mathbb{R}^n$ y $p \geq 1$, entonces $\|x\|_p$ será una función decreciente en p. Al igual que ocurre con el caso $p = 2$, todo producto interior $\langle\cdot,\cdot\rangle$ induce una norma: $\|x\| = \sqrt{\langle x, x\rangle}$. En particular, se cumple la relación $\|x + y\|^2 = \|x\|^2 + \|y\|^2 + 2\langle x, y\rangle$. El vector normalizado será denotado por $\tilde{x} = \frac{1}{\|x\|} x$.

Diremos que dos vectores v_1 y v_2 son *ortogonales* si $\langle v_1, v_2\rangle = 0$. Esta condición se denota por $v_1 \perp v_2$. Si $\langle v_1, v_1\rangle = \langle v_2, v_2\rangle = 1$, diremos que son *vectores ortonormales*. Dos espacios vectoriales V_1 y V_2 son *ortogonales*, escrito $V_1 \perp V_2$, si cada vector de uno de los espacios es ortogonal a cada vector del otro. Si $V_1 \perp V_2$ y $V_1 \oplus V_2 = \mathbb{R}^n$, entonces V_2 es el *complemento ortogonal* de V_1, y se expresa como $V_2 = V_1^{\perp}$. Más generalmente, si $V_1 \perp V_2$ y $V_1 \oplus V_2 = V$, entonces se dice que V_2 es *complemento ortogonal de V_1 con respecto a V*. Esto es, obviamente una relación simétrica: si V_2 es complemento ortogonal de V_1, entonces V_1 es complemento ortogonal de V_2.

La proyección del vector y sobre el vector x viene dada por $\hat{y} = \frac{\langle x,y\rangle}{\|x\|^2}x$. Una propiedad importante de una proyección es que, cuando ésta es sustraída del vector que fue proyectado, el vector resultante, llamado *residual*, es ortogonal a la proyección, esto es, si $r := y - \frac{\langle x,y\rangle}{\|x\|^2}x = y - \hat{y}$, entonces $r \perp \hat{y}$. Nótese que las relaciones pitagóricas se mantienen: $\|y\|^2 = \|\hat{y}\|^2 + \|r\|^2$ (véase la Figura 1.1). Dados m vectores

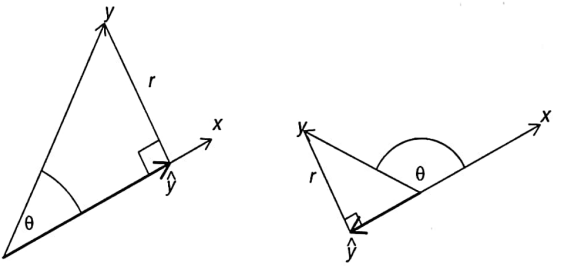

Figura 1.1. Proyección y ángulos.

linealmente independientes $\{x_1, \ldots, x_m\} \in V$, es fácil formar un sistema de m vectores ortonormales, $\{\tilde{x}_1, \ldots, \tilde{x}_m\} \in V$, que generen el mismo subespacio vectorial de V. Una manera simple de hacer esto es secuencialmente. Primero, normalizamos x_1 y lo llamamos $\tilde{x}_1$. Después, proyectamos x_2 sobre $\tilde{x}_1$ y sustraemos esta proyección de x_2. Este resultado es ortogonal a $\tilde{x}_1$ y, por tanto, lo normalizamos y lo llamamos $\tilde{x}_2$ (véase la Figura 1.2):

$$\tilde{x}_1 = \frac{1}{\|x_1\|}x_1, \qquad \tilde{x}_2 = \frac{1}{\|x_2 - \langle \tilde{x}_1, x_2\rangle \tilde{x}_1\|}\left(x_2 - \langle \tilde{x}_1, x_2\rangle \tilde{x}_1\right). \tag{1.1}$$

A estas expresiones se les denomina *transformaciones de Gram-Schmidt*. De manera inductiva, para los m vectores tenemos

$$\tilde{x}_k = \frac{x_k - \sum_{i=1}^{k-1} \langle \tilde{x}_i, x_k\rangle \tilde{x}_i}{\left\| x_k - \sum_{i=1}^{k-1} \langle \tilde{x}_i, x_k\rangle \tilde{x}_i \right\|}, \quad k = 2, 3, \ldots, m. \tag{1.2}$$

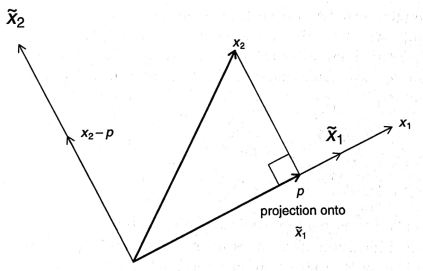

Figura 1.2. Ortogonalización de x_1 y x_2.

Obsérvese que en este método ortogonalizamos, en el paso k-ésimo, el k-ésimo vector calculando su residual con respecto al plano formado por todos los previos $k-1$ vectores ortonormales.

Una base para un espacio vectorial V es a menudo elegida por un conjunto ortonormal porque es fácil trabajar con los vectores de tal conjunto. Si $B = \{u_1, \ldots, u_n\}$ es una base ortonormal de V, entonces

$$x = c_1 u_1 + \cdots + c_n u_n, \quad \text{con} \quad c_i = \langle x, u_i \rangle, \quad \forall x \in V. \tag{1.3}$$

La representación de un vector en una combinación lineal de vectores base ortonormales, como en la ecuación (1.3), se llama *desarollo de Fourier*, y los coeficientes c_i reciben el nombre de *coeficientes de Fourier*. Tomando producto interior a cada lado de la ecuación (1.3) consigo mismo, obtenemos la conocida *Identidad de Parseval*: $\|x\|^2 = \sum_{i=1}^{n} c_i^2$. Otra expresión útil de la identidad de Parseval en la expansión de Fourier es $\|x - \sum_{i=1}^{k} c_i u_i\|^2 = \langle x, x \rangle - \sum_{i=1}^{k} c_i^2$. Consideremos a continuación el espacio vectorial de matrices $V = \mathcal{M}_{n\times m}$, en general con coeficientes reales, salvo que se especifique lo contrario. Definimos la *traza* de una matriz cuadrada $A \in \mathcal{M}_n = \mathcal{M}_{n\times n}$ como la suma de los elementos de su diagonal principal: $tr(A) = \sum_{i=1}^{n} a_{ii}$. Se verifican las siguientes propiedades:

$$tr(A) = tr(A^T), \;\; tr(cA) = c \cdot tr(A), \;\; tr(AB) = tr(BA), \;\; tr(ABC) = tr(BCA) = tr(CAB).$$

Anteriormente hemos visto dos propiedades de vectores que dependen del producto interior: la ortogonalidad entre dos vectores y la norma de un vector, que son invariantes con la orientación de vectores. Las propiedades derivadas del producto interior pueden corresponder mejor con la aplicación si usamos una forma bilinear en la cual la matriz refleja las diferentes distancias a lo largo de los ejes de coordenadas. Una matriz diagonal cuyas entradas tengan valores relativos a los inversos de las escalas relativas de los ejes puede ser más útil, es decir, en lugar de considerar $x^T y$, podemos usar $x^T D y$, donde D es esta matriz diagonal. Del mismo modo, una forma bilinear $x^T A y$ puede corresponder mucho mejor a las propiedades de la aplicación que el producto interno estándar. Así, definimos la ortogonalidad de dos vectores x e y con respecto de una matriz A como $\langle x, y \rangle_A := x^T A y = 0$,

$x, y \in \mathbb{R}^n$. En este caso, decimos que x e y son *A-ortogonales*. En general, si A es una matriz real simétrica definida positiva entonces $\langle x, y\rangle_A$ define verdaderamente un producto interior y la correspondiente norma inducida vendrá dada entonces por $\|x\|_A = \sqrt{\langle x, x\rangle_A} = \sqrt{x^T Ax}$. A esta norma se le conoce en ciertos contextos relacionados con la resolución numérica de Ecuaciones en Derivadas Parciales (métodos de los Elementos Finitos) como *norma energía*.

Resulta un ejercicio elemental de operatoria de matrices el probar que una transformación elemental realizada a una matriz puede obtenerse como el resultado de multiplicar a ésta, bien por la derecha o bien por la izquierda, por una matriz particular que recibe el nombre de *matriz elemental*. En efecto, sean dos vectores columna $u, v \in \mathbb{R}^n$ tales que $v^T u \neq 1$. Entonces, las matrices de la forma $I - uv^T$ reciben el nombre de matrices elementales. Si consideramos el vector e_i como el i-ésimo de la base canónica de $\mathbb{R}^n$, entonces es fácil comprobar las siguientes propiedades para ciertas matrices elementales particulares:

1. Definiendo la matriz $E_{i,j} = I - uu^T$, con $u = e_i - e_j$, entonces $\tilde{A} = E_{i,j} \cdot A$ da como resultado una matriz que consiste en permutar las filas i y j-ésimas de A, mientras que $\tilde{A} = A \cdot E_{i,j}$ produce el mismo efecto por columnas.
2. Definiendo $E_i(\alpha) = I - (1-\alpha)e_i e_i^T$ con $\alpha \in \mathbb{C}$, entonces $\tilde{A} = E_i(\alpha) \cdot A$ da como resultado una matriz que consiste en multiplicar por α la fila i-ésima de A, mientras que $\tilde{A} = A \cdot E_i(\alpha)$ realiza la misma operación cambiando filas por columnas.
3. Definiendo $E_{i,j}(\alpha) = I + \alpha e_j e_i^T$ con $\alpha \in \mathbb{C}$, entonces $\tilde{A} = E_{i,j}(\alpha) \cdot A$ da como resultado una matriz que consiste en sumarle a la fila j-ésima el resultado de multiplicar la fila i-ésima por α, mientras que $\tilde{A} = A \cdot E_{i,j}(\alpha)$ produce el mismo efecto por columnas.

Sea una matriz $A \in \mathcal{M}_{n\times m}$ con $rang(A) = r > 0$. Sean P y Q matrices formadas como productos de matrices elementales tales que

$$PAQ = \begin{pmatrix} I_r & 0 \\ 0 & 0 \end{pmatrix}. \tag{1.4}$$

La expresión PAQ, resultado del proceso de la *Eliminación Gaussiana,* recibe el nombre de *forma canónica equivalente* de A, y existe para cualquier matriz A no nula. Las matrices P y Q no son únicas. El orden en el que son construidas a partir de matrices de operaciones elementales es muy importante, garantizando así la conservación de la precisión en los cálculos (pivotación y escalado, véase el Capítulo 3). A pesar de que las matrices P y Q no son únicas, la forma canónica equivalente por sí misma si que obviamente lo es (lado derecho de (1.4)). Como las matrices elementales son invertibles, las matrices P y Q también lo serán. De una forma canónica equivalente de una matriz A con rango r, tenemos por tanto, la *factorización canónica equivalente de A:*

$$A = P^{-1} \begin{pmatrix} I_r & 0 \\ 0 & 0 \end{pmatrix} Q^{-1}.$$

Finalizamos esta sección introduciendo un concepto que será clave en esta Memoria. Una matriz cuyas filas o columnas constituyen un conjunto de vectores ortonormales se llama *matriz ortogonal*. Si Q es una matriz ortogonal $n \times m$, entonces $QQ^T = I_n$ si $n \leq m$, y $Q^TQ = I_m$ si $n \geq m$. Si $Q \in \mathscr{M}_n$ es una matriz ortogonal cuadrada, entonces $QQ^T = Q^TQ = I_n$. En el caso de tratar con matrices con entradas complejas, la operación Q^T debe reemplazarse por la traspuesta hermitiana Q^H (cambiando filas por columnas y tomando conjugación), y en ese caso a las matrices ortogonales se les denomina *unitarias*. El determinante de una matriz ortogonal cuadrada vale ± 1:

$$1 = |I_n| = |Q \cdot Q^T| = |Q| \cdot |Q^T| = |Q|^2 \Rightarrow |Q| = \pm 1.$$

Cabe destacar que las matrices elementales $E_{i,j} = I - uu^T$, con $u = e_i - e_j$, son simétricas y ortogonales, mientras que las matrices elementales $E_i(\alpha) = I - (1-\alpha)e_i e_i^T$ y $E_{i,j}(\alpha) = I + \alpha e_j e_i^T$, con $\alpha \in \mathbb{C}$, son simétricas.

1.2. Autovalores y autovectores. Factorizaciones canónicas

Sea $A \in \mathscr{M}_n$. Si $v \in \mathbb{C}$, $v \neq 0$, es un vector tal que

$$Av = cv, \tag{1.5}$$

para algún valor $c \in \mathbb{C}$, entonces a v se le denomina *autovector* de la matriz A, y a c un *autovalor, valor característico ó valor singular*. Nos referimos al par (c, v) como un *eigen-par*. Mientras restringimos un autovector a ser distinto de 0, un autovalor puede ser nulo y en este caso, la matriz será singular. Aunque A sea real, tanto un autovalor como un autovector de A pueden ser expresiones complejas.

Debemos remarcar la relación $Av = cv$. El efecto de multiplicar una matriz por un autovector es el mismo que el de multiplicar ese autovector por un escalar. Así, el autovector es un *invariante* de la transformación A en el sentido de que su dirección no cambia.

Enunciamos algunas de las propiedades elementales de autovalores y autovectores son las siguientes.

Lema 1.1. *Si $Av = cv$, con A matriz real, entonces:*

1. *bv es un autovector de A, donde b es cualquier escalar distinto de cero.*
2. *bc es un autovalor de bA, donde b es cualquier escalar distinto de cero.*
3. *(c^{-1}, v) es un eigen-par de A^{-1}, si A es no singular.*
4. *Si A es diagonal o triangular, entonces los autovalores de A son las entradas de la diagonal principal de A, a_{ii}, con autovectores correspondientes e_i (base canónica de $\mathbb{R}^n$).*
5. *(c^k, v) es un eigen-par de A^k para $k = 1, 2, \ldots$.*

6. *Si A y B son matrices cuadradas del mismo orden y si B^{-1} existe, entonces los autovalores de BAB^{-1} son los mismos autovalores que los de A.*

Para la ecuación $(A-cI)v=0$ que definen los autovalores y autovectores de A, vemos que como $v\neq 0$, $(A-cI)$ debe ser singular, y por tanto, $|A-cI|=0$. Esta ecuación permite el cálculo de los autovalores de A. El determinante es un polinomio de grado n en c, $p_A(c)$, llamado el *polinomio característico*, y cuando es igualado a 0 se llama la *ecuación característica*. Un autovalor de A será por tanto una raíz del polinomio característico. Del Teorema Fundamental del Álgebra se sigue que una matriz A tiene exactamente n autovalores complejos (y a lo sumo n reales). Si A es real, los autovalores complejos aparecerán en pares conjugados. Podemos escribir el polinomio característico de forma factorizada como $p_A(c)=(-1)^n(c-c_1)\cdots(c-c_n)$. Definamos el espectro de una matriz A como el conjunto de sus autovalores: $\sigma(A):=\{\lambda\in\mathbb{C}/\lambda$ es autovalor de $A\}$. Sabemos ya que el espacio de autovectores asociado a un autovalor dado tiene dimensión al menos 1. Resulta importante definir el número de veces que un autovalor λ se repite como raíz del polinomio característico, lo que se conoce como *multiplicidad algebraica*, $m_a(\lambda)$. Además, a la dimensión del autoespacio asociado se le llama *multiplicidad geométrica* de λ, $m_g(\lambda)$. Estos dos números están relacionados por la desigualdad $m_g(\lambda)\leq m_a(\lambda)$ (véase la demostración, por ejemplo, en [2, págs 112-114]). A la cantidad $r_\sigma(A)=\max_{i=1,\ldots,n}|c_i|$ se le denomina *radio espectral* de A. Un autovalor c_i que cumpla $|c_i|=r_\sigma(A)$ es llamado *autovalor dominante*, lo denotaremos por c_1.

Dos matrices $A,B\in\mathscr{M}_n$, se dice que son *semejantes* ($A\sim B$) si existe una matriz no singular $P\in\mathscr{M}_n$ tal que

$$B=P^{-1}AP. \tag{1.6}$$

La transformación en dicha ecuación se llama *transformación de semejanza*, que es una relación de semejanza conmutativa y transitiva. Además, si $A\sim B$, entonces $\sigma(A)=\sigma(B)$:

$$|A-cI|=|P^{-1}|\cdot|A-cI|\cdot|P|=|P^{-1}AP-P^{-1}cIP|=|B-cI|.$$

Un tipo importante de transformaciones de semejanza está basado en que la matriz P en (1.6) sea ortogonal. Si Q es ortogonal y $B=Q^TAQ$, se dice que A y B son *ortogonalmente semejantes*. Si B en dicha ecuación es una matriz diagonal, se dice que A es *ortogonalmente diagonalizable*, y a QBQ^T se le llama la *factorización diagonal ortogonal* o *factorización ortogonalmente semejante de A*.

El siguiente resultado muestra la importancia de este tipo de factorizaciones.

Teorema 1.2 (Factorización de Schur). *Toda matriz cuadrada A (compleja en general) puede descomponerse en la forma $A=QUQ^H$, donde Q es una matriz unitaria y U es una matriz triangular superior cuyas entradas en la diagonal principal son los autovalores de A.*

Demostración. Sea $A \in \mathscr{M}_n$, (c, v) un eigen-par arbitrario de A con v normalizado, y formemos una matriz U ortogonal con v en su primera columna: $U = [v|U_2]$. Entonces,

$$U^H AU = \begin{pmatrix} v^H Av & v^H AU_2 \\ U_2^H Av & U_2^H AU_2 \end{pmatrix} = \begin{pmatrix} c & v^H AU_2 \\ 0 & U_2^H AU_2 \end{pmatrix},$$

donde $U_2^H AU_2 \in \mathscr{M}_{n-1}$. Además, sabemos que los autovalores de $U^H AU$ son los mismos que los de A.

Para probar el resultado procedemos por inducción sobre n. Si $n = 2$, entonces $U_2^H AU_2$ es un escalar y el resultado es trivial. Asumamos que la factorización existe para cualquier matriz de $\mathscr{M}_{n-1}$. Sea $A \in \mathscr{M}_n$ y (c, v) un eigen-par arbitrario de A (con v normalizado). Siguiendo el mismo proceso que el párrafo anterior podemos aplicar la hipótesis de inducción a $U_2^H AU_2$, por lo que existe una matriz $V \in \mathscr{M}_{n-1}$ ortogonal tal que $V^H(U_2^H AU_2)V = T$, donde T es triangular superior. Sea $Q = U_2 \begin{pmatrix} 1 & 0 \\ 0 & V \end{pmatrix}$. Multiplicando vemos que $Q^H Q = I$, es decir, Q es unitaria, y

$$Q^H AQ = \begin{pmatrix} c & v^H AU_2 V \\ 0 & V^H U_2^H AU_2 V \end{pmatrix} = \begin{pmatrix} c & v^H AU_2 V \\ 0 & T \end{pmatrix} = U,$$

lo cual demuestra la existencia de la factorización de Schur para cualquier matriz cuadrada A. □

Estudiemos a continuación el concepto de diagonalización. Si V es una matriz cuyas columnas corresponden a los autovectores de A, y C es una matriz diagonal cuyas entradas son los autovalores correspondientes a las columnas de V, usando la definición (1.5), podemos escribir $AV = VC$. Ahora, si V es no singular se tiene que

$$A = VCV^{-1}, \tag{1.7}$$

que representa la *factorización diagonal* de la matriz A. Vemos pues que una matriz A con autovalores $c_1, \ldots, c_n$ que se pueda factorizar de esta manera es similar a la matriz $\text{diag}(c_1, \ldots, c_n)$, y esta representación recibe el nombre de *forma canónica de semejanza de A* o *factorización canónica de semejanza de A*. No todas las matrices pueden ser factorizadas como en la ecuación (1.7). Obviamente, depende de que V sea o no singular, esto es, que el conjunto de autovectores que forma la matriz V sea o no linealmente independiente. Si una matriz puede ser factorizada como en (1.7), se llama *matriz diagonalizable*.

Una condición suficiente y necesaria para que una matriz sea diagonalizable se puede establecer en términos de los autovalores únicos y sus multiplicidades.

Teorema 1.3 (Diagonalizabilidad). *Supongamos que $A \in \mathscr{M}_n$ con autovalores distintos $\lambda_1, \ldots, \lambda_k$ y multiplicidades algebraicas $m_1, \ldots, m_k$, respectivamente $(m_1 + \cdots + m_k = n)$. Entonces, $rang(A - \lambda_l I) = n - m_l$, para todo $l = 1, \ldots, k$, si y sólo si, A es diagonalizable.*

Demostración. Recordemos que A siendo diagonalizable es equivalente a la existencia de una matriz V no singular tal que $AV = VC$ donde C es una matriz diagonal cuyas entradas son los autovalores correspondientes a las columnas de V.

Para ver que la condición es suficiente, asumamos que para cada $l \in \{1,\dots,k\}$, $rang(A-\lambda_l I) = n-m_l$, por lo que $(A-\lambda_l I)x = 0$ tiene exactamente $n-(n-m_l) = m_l$ soluciones linealmente independientes, que son por definición, autovectores de A asociados a λ_l. Sea $\{w_1,\dots,w_{m_l}\}$ un conjunto de autovectores linealmente independientes asociados a λ_l, y u un autovector asociado a λ_j con $\lambda_j \neq \lambda_l$. Los vectores $w_1,\dots,w_{m_l}$ y u son columnas de V. Ahora si u es linealmente dependiente de $w_1,\dots,w_{m_l}$, podemos escribir $u = \sum_{k=1}^{m_l} b_k w_k$, y por tanto $Au = \sum_{k=1}^{m_l} Ab_k w_k = \lambda_l \sum_{k=1}^{m_l} b_k w_k = \lambda_l u$, contradiciendo que u no es un autovector asociado a λ_l. Así, los autovectores asociados con autovalores diferentes son linealmente independientes, y por tanto V es no singular.

Para ver que la condición es necesaria, asumamos que V es no singular; esto es, V^{-1} existe. La matriz $(C-\lambda_l I)$ tiene exactamente m_l ceros en la diagonal principal, y por tanto, $rang(C-\lambda_l I) = n-m_l$. Ya que $V(C-\lambda_l I)V^{-1} = (A-\lambda_l I)$, y dado que al premultiplicar y postmultiplicar una matriz por otra de rango completo no cambia el rango de ésta, tenemos así $rang(A-\lambda_l I) = n-m_l$. □

Algunas propiedades de matrices diagonalizables son:

- Los autovectores de una matriz diagonalizable son linealmente independientes.
- Se cumplen las relaciones

$$|A| = |VCV^{-1}| = |V||C||V^{-1}| = |C|, \qquad tr(A) = tr(VCV^{-1}) = tr(V^{-1}VC) = tr(C).$$

- El número de autovalores distintos de cero de una matriz A diagonalizable es igual al rango de A. Este debe ser el caso porque el rango de la matriz C diagonal es su número de elementos distintos de cero y el rango de A debe ser el mismo que el rango de C.
- La suma de las multiplicidades de los autovalores únicos distintos de cero es igual al rango de la matriz; esto es, $\sum_{i=1}^{k} m_i = rang(A)$, para la matriz A con k autovalores distintos con multiplicidades m_i.

Para finalizar esta sección, nos centraremos en un caso particular de matrices que aparecen con suma frecuencia en numerosos procesos aplicados: las matrices reales y simétricas. Como primer resultado tenemos el siguiente

Lema 1.4. *Si A es una matriz real y simétrica, entonces $\sigma(A) \subset \mathbb{R}$.*

Demostración. Probaremos que si $\lambda \in \sigma(A)$, entonces $\lambda^2 \geq 0$. De la relación $A\cdot x = \lambda \cdot x$ se sigue $x^T \cdot A^T = \lambda \cdot x^T$, y como $A^T = A$, tenemos que $x^T \cdot A \cdot x = \lambda \cdot x^T \cdot x$, lo que implica $\lambda \cdot x^T \cdot A \cdot x = \lambda^2 \cdot x^T \cdot x$, o equivalentemente, $x^T \cdot A^2 \cdot x = \lambda^2 \cdot x^T \cdot x$. Por otro lado, tenemos que $< A\cdot x, A\cdot x >= (A\cdot x)^T \cdot A \cdot x = x^T \cdot A^T \cdot A \cdot x = x^T \cdot A^2 \cdot x$, por lo que concluimos $\|A\cdot x\|_2^2 = \lambda^2 \cdot \|x\|_2^2$, implicando $\lambda^2 \geq 0$. □

En el caso de una matriz real y simétrica A, cualquier par de autovectores correspondientes a dos autovalores diferentes son ortogonales. En efecto, asumamos que c_1 y c_2 son dos autovalores distintos con correspondientes autovectores v_1 y v_2. Si consideramos $v_1^T v_2$, entonces multiplicándolo por c_2 obtenemos

$$c_2 v_1^T v_2 = v_1^T A v_2 = v_2^T A v_1 = c_1 v_2^T v_1 = c_1 v_1^T v_2 \quad \text{donde hemos usado} \quad A = A^T.$$

Como $c_1 \neq c_2$, solo puede darse $v_1^T v_2 = 0$. Por otro lado, consideremos ahora dos autovectores distintos v_i y v_j asociados a un autovalor c (con multiplicidad geométrica mayor que 1) tales que $v_i \neq \lambda \cdot v_j, \forall\ \lambda \in \mathbb{C}$. Por lo que hemos visto, todo autovector asociado a un autovalor c_k, $c_k \neq c$, es ortogonal al espacio generado por los autovectores de c. Tomemos v_i normalizado y apliquemos la transformación de Gram-Schmidt a v_j, produciendo un vector $\tilde{v}_j = \frac{1}{\|v_j - \langle v_i, v_j\rangle v_i\|}(v_j - \langle v_i, v_j\rangle v_i)$ que es ortogonal a v_i tal que $\text{span}\{v_i, v_j\} = \text{span}\{v_i, \tilde{v}_j\}$. Ahora, tenemos que

$$\begin{aligned} A\tilde{v}_j &= \frac{1}{\|v_j - \langle v_i, v_j\rangle v_i\|}(Av_j - \langle v_i, v_j\rangle Av_i) = \frac{1}{\|v_j - \langle v_i, v_j\rangle v_i\|}(cv_j - \langle v_i, v_j\rangle cv_i) \\ &= c\frac{1}{\|v_j - \langle v_i, v_j\rangle v_i\|}(v_j - \langle v_i, v_j\rangle v_i) = c\tilde{v}_j. \end{aligned}$$

Concluimos así el siguiente resultado:

Teorema 1.5. *Si $A \in \mathcal{M}_n$ es real y simétrica, entonces existe una base ortogonal de $\mathbb{R}^n$ formada por vectores propios.*

Una matriz real y simétrica es ortogonalmente diagonalizable, porque V en la ecuación (1.7) puede ser elegida que sea ortogonal, y por tanto escrita como

$$A = VCV^T, \tag{1.8}$$

donde $VV^T = V^T V = I$, y así, tenemos que $V^T AV = C$. Tal matriz es ortogonalmente semejante a una matriz diagonal formada por sus autovalores. Por otro lado, cuando A es real y simétrica y los autovectores v_i son elegidos ortonormales, podemos escribir

$$I = \sum_{i=1}^{n} v_i v_i^T \Rightarrow A = A\sum_{i=1}^{n} v_i v_i^T = \sum_{i=1}^{n} Av_i v_i^T = \sum_{i=1}^{n} c_i v_i v_i^T.$$

Esta representación se llama *descomposición espectral* de A. Es esencialmente la misma ecuación que (1.8), por tanto, a la expresión $A = VCV^T$ también se le denomina descomposición espectral de A. Esta representación es única excepto por el orden en la elección de los autovalores y autovectores. Si el rango de la matriz es r, suelen ser denotados por $|c_1| \geq \ldots \geq |c_r| > 0$, y si $r < n$, entonces $c_{r+1} = \cdots = c_n = 0$. Nótese que las matrices en la descomposición espectral son matrices de proyección, que son ortogonales entre sí, pero que no son matrices ortogonales, y suman la identidad. Definiendo $P_i = v_i v_i^T$ tenemos que

$$P_i^2 = P_i, \quad P_i P_j = 0 \ (i \neq j), \quad \sum_{i=1}^{n} P_i = I, \quad A = \sum_{i=1}^{n} c_i P_i.$$

A las matrices P_i se les llama *proyectores espectrales*. Obsérvese que la descomposición espectral también se aplica a las potencias de A: $A^k = \sum_{i=1}^{n} c_i^k v_i v_i^T$, $k \in \mathbb{Z}$.

Finalmente, supongamos que la matriz real y simétrica A, es además definida positiva, es decir que $x^T Ax > 0$, $\forall x \in \mathbb{R}^n$ (o lo que es lo mismo, $\sigma(A) \subset \mathbb{R}^+$). Entonces, $A = VCV^T$ con C una matriz diagonal con elementos en la diagonal principal positivos. De este modo, $A^{-1} = VC^{-1}V^T$ y como C^{-1} es la matriz diagonal cuyas entras son los inversos de C, tenemos que A^{-1} es también definida positiva. Vemos por tanto que A es definida positiva, si y solo si, A^{-1} es definida positiva.

1.3. Normas matriciales

En Análisis Numérico es frecuente el necesitar saber cuánto de próximo se encuentra una solución aproximada de un determinado problema con respecto a su solución exacta. Para poder dar respuesta a esto de manera cuantitativa cuando trabajamos con problemas que involucran a matrices necesitamos introducir el concepto de *norma matricial,* como una aplicación

$$\begin{aligned} \|\cdot\| : \ \mathscr{M}_n &\to \mathbb{R} \\ A &\to \|A\| \end{aligned}$$

que cumple, para toda $A, B \in \mathscr{M}_n$, $\lambda \in \mathbb{R}$ ($\mathbb{C}$), las condiciones

i) $\|A\| \geq 0$, ii) $\|A\| = 0 \Longleftrightarrow A = 0$, iii) $\|\lambda \cdot A\| = |\lambda| \cdot \|A\|$,
iv) $\|A + B\| \leq \|A\| + \|B\|$, v) $\|A \cdot B\| \leq \|A\| \cdot \|B\|$.

Con el fin de establecer conexiones entre normas vectoriales y matriciales establecemos la siguiente

Definición 1.6. *Sea $\|\cdot\|_V$ una norma vectorial y $\|\cdot\|_M$ una norma matricial. Se dice que ambas normas son* compatibles *si $\|A \cdot x\|_V \leq \|A\|_M \cdot \|x\|_V$, $\forall A \in \mathscr{M}_n$ y $\forall x \in \mathbb{C}^n$.*

Dada una norma vectorial, existen infinitas normas matriciales compatibles con ésta. Recíprocamente, se puede comprobar que dada una norma matricial, existen infinitas normas vectoriales compatibles con ella. De todas estas normas va a resultar de especial interés una en particular, que viene dada por la siguiente

Definición 1.7. *Dada una norma vectorial $\|\cdot\|_V$, definimos la norma matricial* inducida *por ésta según $\|A\|_M = \max_{\|x\|_V = 1} \|A \cdot x\|_V$.*

Es fácil comprobar que la norma matricial inducida por una norma vectorial es efectivamente una norma y que es compatible con la norma vectorial. Algunos ejemplos particulares son:

$$\|A\|_1 = \max_{1 \leq j \leq n} \sum_{i=1}^{n} |a_{i,j}|, \qquad \|A\|_\infty = \max_{1 \leq i \leq n} \sum_{j=1}^{n} |a_{i,j}|, \qquad \|A\|_2 = \sqrt{r_\sigma(A^H \cdot A)}.$$

Observación 1.8. Para la demostración del caso $p = 2$ es necesario hacer uso del Teorema 1.5.

Como consecuencia observamos que:

1. $\|A\|_2 = \|A^H\|_2$. Nótese que esto no se cumple para $\|\cdot\|_p$, con $p = 1, \infty$.
2. Si A es hermitiana, entonces $\|A\|_2 = r_\sigma(A)$.

El siguiente resultado justifica la importancia del cálculo de $r_\sigma(A)$, y es que aunque las diferentes normas estudiadas de una matriz dada nos pueden ofrecer cantidades bien distintas, el radio espectral será siempre una cota inferior de las mismas, y por lo tanto, viene a jugar en cierto sentido el papel de norma matricial mínima.

Teorema 1.9. *Sea $\|\cdot\|_M$ una norma matricial compatible con una norma vectorial $\|\cdot\|_V$ dada (en particular, la norma inducida). Entonces, $\forall A \in \mathscr{M}_n$ se cumple $0 \leq r_\sigma(A) \leq \|A\|_M$.*

Puede demostrarse además que siempre es posible encontrar una norma matricial que se aproxime tanto como uno quiera al radio espectral de una matriz dada (véase [4, Pág. 12])

Teorema 1.10. *Sea $\varepsilon > 0$. Entonces existe una norma matricial $\|\cdot\|_\varepsilon$ para la cual se cumple $\|A\|_\varepsilon - \varepsilon \leq r_\sigma(A)$, $\forall A \in \mathscr{M}_n$.*

La combinación de los Teoremas 1.9 y 1.10 permite escribir

$$r_\sigma(A) = \text{ínf}\{\|A\| \text{ tal que } \|\cdot\| \text{ es norma matricial}\}.$$

Hemos visto además que este ínfimo se alcanza en el caso particular $\|\cdot\| = \|\cdot\|_2$ y A matriz hermitiana, lo cual pone también de relevancia la importancia de trabajar en procesos numéricos con la norma $p = 2$.

Como combinación de los resultados anteriores podemos concluir con el siguiente

Corolario 1.11. *Dada $A \in \mathscr{M}_n$. Entonces*

$$r_\sigma(A) < 1 \Leftrightarrow \|A\|_M < 1, \quad \textit{para alguna norma matricial } \|\cdot\|_M.$$

Demostración. " $\Leftarrow$ "

Si existe $\|\cdot\|_M$ tal que $\|A\|_M < 1$, como $r_\sigma(A) \leq \|A\|_M$, para toda $\|\cdot\|_M$ norma matricial (Teorema 1.9), se sigue que $r_\sigma(A) < 1$.

" $\Rightarrow$ "

Supongamos que $r_\sigma(A) < 1$. Tomemos $\varepsilon/2$ donde $\varepsilon = 1 - r_\sigma(A)$. Entonces, por el Teorema 1.10 se sigue que $\exists\ \|\cdot\|_{\varepsilon/2}$ tal que

$$\|A\|_{\varepsilon/2} - \frac{\varepsilon}{2} \leq r_\sigma(A) \ \Rightarrow\ \|A\|_{\varepsilon/2} \leq r_\sigma(A) + \frac{\varepsilon}{2} < 1.$$

Esto prueba por tanto la existencia de una norma matricial $\|\cdot\|_{\varepsilon/2}$ para la cual $\|A\|_{\varepsilon/2} < 1$.

2

Transformaciones matriciales y factorizaciones

En la mayoría de las aplicaciones del Álgebra Lineal los problemas son resueltos mediante transformaciones de matrices. Una matriz dada que representa alguna transformación de un vector es transformada de manera que permita determinar un vector a partir de otro vector dado. El ejemplo más elemental es la resolución de un sistema lineal de ecuaciones $AX = B$. La matriz A es transformada a través de una sucesión de operaciones de manera que la solución X pueda determinarse fácilmente a partir del vector dado B y la matriz transformada. Otro ejemplo elemental y de gran importancia es el cálculo de los autovalores y autovectores de una matriz. La mayoría de estas transformaciones provienen como resultado de ciertas factorizaciones de la matriz A.

En el capítulo anterior ya hemos introducido algunas factorizaciones matriciales particulares. El objetivo de este capítulo es introducir algunas transformaciones matriciales generales (transformaciones ortogonales, geométricas y de reflexión) para introducir a continuación dos factorizaciones nuevas adicionales: la factorización $LU(LDU)$ y la factorización QR. Para nuestros objetivos en esta Memoria haremos especial énfasis en la factorización QR, dada su importancia en numerosos procesos numéricos y por no haber sido estudiada durante el Grado en Matemáticas.

2.1. Transformaciones ortogonales y geométricas. Reflexiones de Householder y rotaciones de Givens

Transformaciones por Matrices Ortogonales

En el capítulo previo hemos introducido algunas propiedades que tienen las matrices ortogonales ($Q^TQ = I$). Comenzamos esta sección añadiendo una nueva propiedad:

$$\langle Qx, Qy\rangle = (Qy)^T(Qx) = y^T Q^T Qx = y^T x = \langle x, y\rangle, \quad \forall x, y \in \mathbb{R}^n$$

y así, $\arccos\left(\frac{\langle Qx,Qy\rangle}{\|Qx\|_2\|Qy\|_2}\right) = \arccos\left(\frac{\langle x,y\rangle}{\|x\|_2\|y\|_2}\right)$. Vemos por tanto que las transformaciones ortogonales

$$\begin{aligned} q : \mathbb{R}^n &\to \mathbb{R}^n \\ x &\to q(x) = Q \cdot x \end{aligned}$$

conservan ángulos. A menudo usamos las transformaciones ortogonales que conservan longitudes y ángulos mientras rotamos ó reflejamos regiones de $\mathbb{R}^n$ en $\mathbb{R}^n$, por lo que es importante mencionar que nos referiremos a ellas apropiadamente como *rotadores* ó *reflectores*, respectivamente.

Transformaciones Geométricas

Las operaciones algebraicas son transformaciones geométricas que rotan, deforman, ó trasladan un objeto. Estas transformaciones, que son normalmente usadas en dos o tres dimensiones pues se corresponden con el espacio físico fácilmente percibido, tienen aplicaciones similares en dimensiones superiores. Pensar en Álgebra Lineal en términos de operaciones geométricas asociadas, a menudo proporciona una intuición útil. Para este tipo de transformaciones resulta importante conocer qué objetos permanecen inalterados mediante la transformación. Enumeramos en la siguiente tabla algunas de ellas que son lineales, dado que preservan líneas rectas.

Transformación	**Preservación**
lineal y afín	rectas
escalado	rectas y ángulos
traslación	rectas, ángulos y distancias
rotación	rectas, ángulos y distancias
reflexión	rectas, ángulos y distancias

Tabla 2.1. Invarianza de algunas transformaciones.

Las transformaciones que conservan longitudes y ángulos se llaman *transformaciones isométricas* (como por ejemplo, las transformaciones ortogonales). Éstas preservan también áreas y volúmenes. Otra transformación isométrica es una *traslación* $\tilde{x} = x + t$. Una transformación que conserva ángulos recibe el nombre de *transformación isotrópica*. Un ejemplo de ésta que no es isométrica, es una transformación uniforme de escalado o dilatación, $\tilde{x} = ax$, donde a es un escalar. La transformación $\tilde{x} = Ax$, donde A es una matriz diagonal con no todos sus elementos iguales, no conserva ángulos (*escalamiento anisotrópico*). Otra transformación anisotrópica

es la *transformación de corte*, $\tilde{x} = Ax$, donde A coincide con la matriz identidad, excepto por una sola fila o columna que tiene uno en la diagonal, pero posiblemente elementos distintos de cero en otras posiciones; por ejemplo

$$A = \begin{pmatrix} 1 & 0 & a_1 \\ 0 & 1 & a_1 \\ 0 & 0 & 1 \end{pmatrix} \quad , \quad a_1 \neq 0.$$

A pesar de que no conservan ángulos, ambas (escalamiento anisotrópico y transformación de corte) conservan lineas paralelas. Una transformación que conserva lineas paralelas recibe el nombre de *transformación afín*, mientras que una *transformación proyectiva*, la cual usa sistema de coordenadas homogéneo del plano proyectivo, conservará lineas rectas, pero no conservará lineas paralelas.

Analizamos a continuación algunas transformaciones que nos resultarán de especial interés.

Rotaciones

La rotación más simple de un vector es la rotación del plano definido por dos coordenadas sobre los otros ejes principales. Esta rotación cambia dos componentes de todos los vectores de ese plano y deja inalterados a todas las demás componentes. Esta rotación se puede describir en el espacio 2−dimensional que definen las coordenadas a cambiar, sin hacer referencia a las otras coordenadas.

Consideramos la rotación del vector x a través del ángulo θ en $\tilde{x}$. La longitud se conserva, así que tenemos $\|\tilde{x}\| = \|x\|$, véase la Figura 2.1.

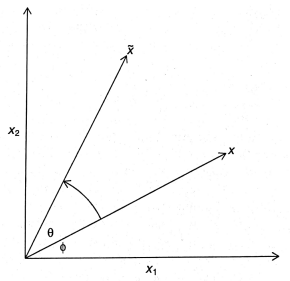

Figura 2.1. Rotación del vector x a través del ángulo θ.

Podemos escribir $x_1 = \|x\|\cos(\phi+\theta)$ y $x_2 = \|x\|\sin(\phi+\theta)$, donde $x = \|x\|_\phi$. Ahora, de la trigonometría elemental, tenemos que

$$\cos(\phi+\theta) = \cos\phi\cos\theta - \sin\phi\sin\theta, \qquad \sin(\phi+\theta) = \sin\phi\cos\theta + \cos\phi\sin\theta.$$

Como $\cos\phi = x_1/\|x\|$ y $\sin\phi = x_2/\|x\|$, podemos combinarlos para obtener $\tilde{x}_1 = x_1\cos\theta - x_2\sin\theta$ y $\tilde{x}_2 = x_1\sin\theta + x_2\cos\theta$. Por tanto, podemos realizar la rotación de x multiplicándolo por una matriz ortogonal:

$$\begin{pmatrix}\tilde{x}_1\\ \tilde{x}_2\end{pmatrix} = \begin{pmatrix}\cos\theta & -\sin\theta\\ \sin\theta & \cos\theta\end{pmatrix}\cdot\begin{pmatrix}x_1\\ x_2\end{pmatrix}. \tag{2.1}$$

Una rotación de cualquier hiperplano en un $n-$espacio se puede obtener como n rotaciones sucesivas de hiperplanos formados por dos ejes principales. En un espacio tridimensional, esto es conocido como el *Teorema de la rotación de Euler.*

Traslaciones

Las *traslaciones* son relativamente simples transformaciones envolviendo la suma de vectores. En aplicaciones donde varias transformaciones geométricas están combinadas, resulta conveniente que las traslaciones puedan también ser obtenidas mediante la multiplicación por una cierta matriz. Esto puede hacerse usando *coordenadas homogéneas*, las cuales forman el sistema de coordenadas natural para la Geometría Proyectiva. De este modo la traslación $\tilde{x} = x + t$ con $x = (x_1,\ldots,x_n)$, $t = (t_1,\ldots,t_n)$ puede llevarse a cabo representando primero el punto x como $(1, x_1,\ldots,x_n)$ y multiplicar a continuación por la matriz

$$T = \begin{pmatrix}1 & 0 & \ldots & 0\\ t_1 & 1 & \ldots & 0\\ \vdots & \vdots & \ddots & \vdots\\ t_n & 0 & \ldots & 1\end{pmatrix} \in \mathscr{M}_{n+1}$$

para obtener así el vector trasladado $\tilde{x} = (1, x_1 + t_1,\ldots,x_n + t_n)$.

Reflexiones

Sean u y v dos vectores ortonormales y $x = c_1 u + c_2 v$ para ciertos escalares c_1 y c_2. El vector

$$\tilde{x} = -c_1 u + c_2 v \tag{2.2}$$

es una *reflexión* de x a través de la recta definida por el vector v, o $u^{\perp}$. Podemos considerar una reflexión que transforma un vector $x = (x_1,\ldots,x_n)$ en un vector colineal con el vector unidad,

$$\tilde{x} = (0,\ldots,0,\tilde{x}_i,0,\ldots,0) = \pm\|x\|_2 e_i. \tag{2.3}$$

En la Figura 2.2 vemos una representación geométrica en dos dimensiones donde $i = 1$. El vector x se gira dependiendo de la elección de signo $\pm$ en la ecuación (2.3).

Reflexiones de Householder

En esta subsección estudiaremos una transformación de gran importancia que combina las transformaciones geométricas que acabamos de estudiar. Consideramos el problema de reflejar x a través del vector u. Como antes, asumimos que u y

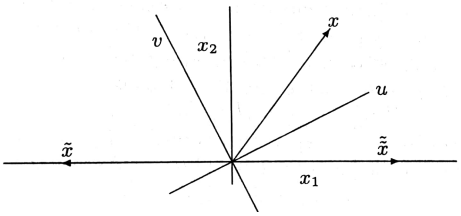

Figura 2.2. Reflexión de x sobre $u^{\perp}$.

v son vectores ortonormales y que $x = c_1 u + c_2 v$. Definamos la matriz $H = I - 2uu^T$. Como $u^T \cdot u = 1$, resulta

$$\begin{aligned} Hx &= c_1 u + c_2 v - 2c_1 uu^T u - 2c_2 uu^T v = c_1 u + c_2 v - 2c_1 u^T uu - 2c_2 u^T vu \\ &= -c_1 u + c_2 v = \tilde{x}, \end{aligned}$$

como en la ecuación (2.2). La matriz H es un *reflector*, pues ha transformado x en su reflexión sobre u. Esta reflexión recibe el nombre de *reflexión de Householder* o *transformación de Householder*, y la matriz H se llama *matriz de Householder* o *reflector de Householder*. Las siguientes propiedades son inmediatas:

(1) H es simétrica ($H = H^T$).
(2) H es ortogonal ($H^T = H^{-1}$).
(3) $Hu = -u$
(4) $Hv = v$, $\forall$ v ortogonal a u.

Al ser H ortogonal, si $Hx = \tilde{x}$, entonces $\|x\|_2 = \|\tilde{x}\|_2$, por tanto si $i = 1$, $\tilde{x}_1 = \pm\|x\|_2$, pues el vector x se ha reflejado en la dirección de x_1, $\tilde{x} = (\pm\|x\|_2, 0, \ldots, 0)$. La matriz uu^T es simétrica, idempotente, y de rango 1. A una transformación de una matriz de la forma $A - vw^T$ se le denomina actualización "rango 1"(por ser vw^T de rango 1.) Por tanto, una reflexión Householder es un caso particular de actualización de rango 1.

La utilidad de las reflexiones de Householder resulta del hecho de que es fácil construir una reflexión que transforma un vector x en un vector $\tilde{x}$, que tiene ceros en todas las posiciones menos una, como en la ecuación (2.3). Para construir el reflector de x en $\tilde{x}$, partimos del vector $v = x - \tilde{x}/\|\tilde{x}\|_2$. Elegimos el signo de $\|\tilde{x}\|_2$ de manera que evitemos cantidades con signos opuestos y magnitudes similares. Es decir, meramente por cuestiones de estabilidad numérica. Por tanto, consideramos

$$q = (x_1, \ldots, x_{i-1}, x_i + signo(x_i)\|x\|_2, x_{i+1}, \ldots, x_n)^T,$$

luego tomamos $u = q^T/\|q\|_2$, para definir finalmente $H = I - 2uu^T$. De manera ilustrativa, consideremos el siguiente

Ejemplo 2.1 *Sea el vector* $x = (3,1,2,1,1)^T$, *de norma* $\|x\|_2 = 4$, *el cual queremos transformar en* $\tilde{x} = (\pm 4,0,0,0,0)^T$. *Construimos así el vector* $q = (7,1,2,1,1)^T$, *con* $\|q\|_2 = \sqrt{56}$, *para obtener* $u = \frac{1}{\sqrt{56}}(7,1,2,1,1)^T$ *y el reflector*

$$H = I - 2uu^T = \begin{pmatrix} 1&0&0&0&0\\ 0&1&0&0&0\\ 0&0&1&0&0\\ 0&0&0&1&0\\ 0&0&0&0&1 \end{pmatrix} - \frac{1}{28}\begin{pmatrix} 49&7&14&7&7\\ 7&1&2&1&1\\ 14&2&4&2&2\\ 7&1&2&1&1\\ 7&1&2&1&1 \end{pmatrix} = \frac{1}{28}\begin{pmatrix} -21&-7&-14&-7&-7\\ -7&27&-2&-1&-1\\ -14&-2&24&-2&-2\\ -7&-1&-2&27&-1\\ -7&-1&-2&-1&27 \end{pmatrix},$$

el cual verifica $Hx = (-4,0,0,0,0)^T$.

Rotaciones de Givens

Anteriormente vimos transformaciones geométricas que rotan un vector de manera que una componente específica del vector se convierte en 0 y sólo otra componente del vector queda modificada. Tal método puede ser particularmente útil si solo una parte de la matriz a transformar está disponible. Estas transformaciones se llaman *transformaciones de Givens*, *rotaciones de Givens*, o también son conocidas como *transformaciones de Jacobi*. La idea básica de esta rotación (que es un caso particular de las estudiadas en la sección anterior) podemos verla en un vector de dimensión 2. Dado un vector $x = (x_1, x_2)$, queremos rotarlo en $\tilde{x} = (\tilde{x}_1, 0)$. Como con un reflector, $\tilde{x}_1 = \|x\|_2$, véase la Figura 2.3.

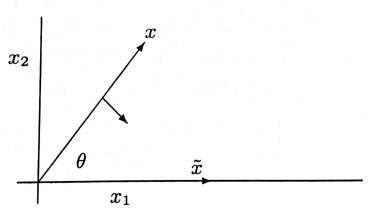

Figura 2.3. Rotación de x a los ejes de coordendas.

Es fácil ver que la matriz ortogonal

$$Q = \begin{pmatrix} \cos\theta & \sin\theta \\ -\sin\theta & \cos\theta \end{pmatrix} \tag{2.4}$$

realizará esta rotación de x si $\cos\theta = x_1/r$ y $\sin\theta = x_2/r$, donde $r = \|x\|_2 = \sqrt{x_1^2 + x_2^2}$. Nótese que la matriz Q es la misma que en (2.1) solo que en la dirección opuesta, y que θ no es realmente relevante dado que solo necesitamos números reales c y s tales que $Q = \begin{pmatrix} c & s \\ -s & c \end{pmatrix}$ verificando $c^2 + s^2 = 1$. Tenemos así que $\tilde{x}_1 = \frac{x_1^2}{r} + \frac{x_2^2}{r} = \|x\|_2$ y $\tilde{x}_2 = -\frac{x_2 x_1}{r} + \frac{x_1 x_2}{r} = 0$, esto es, $Q\begin{pmatrix} x_1 \\ x_2 \end{pmatrix} = \begin{pmatrix} \|x\|_2 \\ 0 \end{pmatrix}$.

Al igual que las reflexiones de Householder, que transforman el vector $x = (x_1,\ldots,x_n)$ en $\tilde{x}_H = (\tilde{x}_{H1},0,\ldots,0)$, las rotaciones de Givens permiten también conseguir transformar x en $\tilde{x}_G = (\tilde{x}_{G1},0,x_3,\ldots,x_n)$, es decir, conseguir anular en este caso una de las entradas del vector transformado. Más generalmente, el vector $x = (x_1,\ldots,x_p,\ldots,x_q,\ldots,x_n)$ se puede transformar en $\tilde{x} = G_{pq}x$ (cumpliendo $\tilde{x}_p = 0$ ó $\tilde{x}_q = 0$), siendo G_{pq} la matriz identidad de orden n que tiene las cuatro entradas que ocupan las posiciones $(p,p),(p,q),(q,p)$ y (q,q) sustituidas por las cuatro entradas de la matriz (2.4). Esto es,

$$G_{pq} = \begin{pmatrix}
1 & 0 & \cdots & 0 & 0 & 0 & \cdots & 0 & 0 & 0 & \cdots & 0 \\
0 & 1 & \cdots & 0 & 0 & 0 & \cdots & 0 & 0 & 0 & \cdots & 0 \\
\vdots & & \ddots & \vdots & \vdots & \vdots & & \vdots & \vdots & \vdots & & \vdots \\
0 & 0 & \cdots & 1 & 0 & 0 & \cdots & 0 & 0 & 0 & \cdots & 0 \\
0 & 0 & \cdots & 0 & c & 0 & \cdots & 0 & s & 0 & \cdots & 0 \\
0 & 0 & \cdots & 0 & 0 & 1 & \cdots & 0 & 0 & 0 & \cdots & 0 \\
\vdots & \vdots & & \vdots & \vdots & \vdots & \ddots & \vdots & \vdots & \vdots & & \vdots \\
0 & 0 & \cdots & 0 & 0 & 0 & \cdots & 1 & 0 & 0 & \cdots & 0 \\
0 & 0 & \cdots & 0 & -s & 0 & \cdots & 0 & c & 0 & \cdots & 0 \\
0 & 0 & \cdots & 0 & 0 & 0 & \cdots & 0 & 0 & 1 & \cdots & 0 \\
\vdots & \vdots & & \vdots & \vdots & \vdots & & \vdots & \vdots & \vdots & \ddots & \vdots \\
0 & 0 & \cdots & 0 & 0 & 0 & \cdots & 0 & 0 & 0 & \cdots & 1
\end{pmatrix} \in \mathcal{M}_n$$

donde las entradas en las filas y columnas p-ésima y q-ésima son $c = \frac{x_p}{r}$ y $s = \frac{x_q}{r}$ con $r = \sqrt{x_p^2 + x_q^2}$.

2.2. Factorizaciones matriciales

La idea de representar $A = BC$ con B y C matrices estructuradas, tiene como particular objetivo la resolución numérica de sistemas de ecuaciones o el cálculo numérico de autovalores y autovectores. Ya hemos visto en el capítulo anterior algunas factorizaciones particulares. Para finalizar este capítulo veremos dos factorizaciones más de gran importancia: $LU(LDU)$ y QR.

2.2.1. Factorización LU (LDU)

Cualquier matriz cuadrada puede expresarse como $A = LU$, donde L es triangular inferior y U es triangular superior. El producto LU recibe el nombre de *factorización LU*. Una factorización LU se logra como consecuencia de la eliminación Gaussiana que consiste en la premultiplicación y postmultiplicación de la matriz A

por matrices elementales de manera recursiva con el fin de conseguir ceros en las entradas por debajo de la diagonal principal de cada columna.

En el proceso de la eliminación Gaussiana resulta conveniente mantener almacenados los multiplicadores. Las entradas de las matrices $A^{(k)}$ que se van obteniendo durante este proceso, con $k = 1,\ldots,n$, se calculan mediante

$$m_{i,k} = \frac{a_{i,k}^{(k)}}{a_{k,k}^{(k)}}, \quad \forall\ i = k{+}1, k{+}2, \ldots, n, \qquad a_{i,j}^{(k+1)} = a_{i,j}^{(k)} - m_{i,k} \cdot a_{k,j}^{(k)}, \quad \forall\ i,j = k{+}1, \ldots, n. \tag{2.5}$$

La razón se justifica en el siguiente resultado junto con las consideraciones posteriores.

Teorema 2.1 (Descomposición LU). *Sea $A \in \mathscr{M}_n$ una matriz no singular, U la matriz triangular superior obtenida en el proceso de la eliminación Gaussiana, y L la matriz triangular inferior definida de la siguiente forma, siendo sus entradas los multiplicadores obtenidos en el proceso de la eliminación Gaussiana:*

$$L = \begin{pmatrix} 1 & 0 & 0 & \cdots & 0 \\ m_{2,1} & 1 & 0 & \cdots & 0 \\ m_{3,1} & m_{3,2} & 1 & \cdots & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ m_{n,1} & m_{n,2} & m_{n,3} & \cdots & 1 \end{pmatrix}.$$

Entonces se verifica la relación $A = LU$.

A esta descomposición se le conoce como *descomposición de Doolittle*. Sin embargo, dicha descomposición no es única. En efecto, existen muchas otras posibles descomposiciones en la forma $A = LU$, no necesariamente verificando $l_{i,i} = 1$. La terminología *métodos compactos* hace referencia a las posibles descomposiciones en la forma LU para una matriz dada A, haciendo que éstas puedan expresar la información de las matrices L y U de manera compacta. Supongamos que tenemos dos descomposiciones $A = L_1U_1 = L_2U_2$. Se sigue entonces la relación matricial

$$A = L_1U_1 = L_2U_2 \ \Rightarrow\ L_1U_1U_1^{-1} = L_2U_2U_1^{-1} \ \Rightarrow\ L_2^{-1}L_1 = U_2U_1^{-1}.$$

Como la inversa de una matriz triangular inferior (superior) es de nuevo triangular inferior (superior), el producto de dos matrices triangular inferior (superior) es de nuevo triangular inferior (superior), y la inversa de una matriz diagonal D es otra matriz diagonal $\tilde{D}$ cuyas entradas son las inversas de las entradas de D $\left(\tilde{d}_{i,i} = d_{i,i}^{-1}, i = 1,\ldots,n\right)$, tenemos que $L_2^{-1}L_1$ es una matriz triangular inferior, mientras que la matriz $U_2U_1^{-1}$ es una matriz triangular superior. Debe cumplirse pues

$$L_2^{-1}L_1 = U_2U_1^{-1} = D, \text{ siendo D una matriz diagonal,}$$

de lo que se concluye que

$$L_2^{-1}L_1 = D \;\Rightarrow\; L_1 = L_2 \cdot D \quad \text{y} \quad U_2U_1^{-1} = D \;\Rightarrow\; U_1 = D^{-1}U_2.$$

La elección particular de una matriz diagonal D está ligada a la elección particular de los elementos de la diagonal principal de las matrices L ó U. Un ejemplo particular consiste en imponer que sean los elementos de la diagonal principal de la matriz U los que valgan 1, es decir $u_{i,i} = 1,\ i = 1,\ldots,n$. Esta elección particular recibe el nombre de *descomposición de Crout*. Obsérvese que podemos pasar de la descomposición de Doolittle a la de Crout y viceversa de manera inmediata. En efecto, si $A = LU$ denota la descomposición de Doolittle (A no singular, por lo que $u_{i,i} \neq 0,$ para todo $i = 1,\ldots,n$), y $A = L_cU_c$ denota la descomposición de Crout, entonces

$$A = LU = \begin{pmatrix} 1 & 0 & 0 & \cdots & 0 \\ l_{2,1} & 1 & 0 & \cdots & 0 \\ l_{3,1} & l_{3,2} & 1 & \cdots & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ l_{n,1} & l_{n,2} & l_{n,3} & \cdots & 1 \end{pmatrix} \cdot \begin{pmatrix} u_{1,1} & u_{1,2} & u_{1,3} & \cdots & u_{1,n} \\ 0 & u_{2,2} & u_{2,3} & \cdots & u_{2,n} \\ 0 & 0 & u_{3,3} & \cdots & u_{3,n} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & 0 & \cdots & u_{n,n} \end{pmatrix}$$

donde

$$U = \begin{pmatrix} u_{1,1} & 0 & 0 & \cdots & 0 \\ 0 & u_{2,2} & 0 & \cdots & 0 \\ 0 & 0 & u_{3,3} & \cdots & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & 0 & \cdots & u_{n,n} \end{pmatrix} \cdot \begin{pmatrix} 1 & \frac{u_{1,2}}{u_{1,1}} & \frac{u_{1,3}}{u_{1,1}} & \cdots & \frac{u_{1,n}}{u_{1,1}} \\ 0 & 1 & \frac{u_{2,3}}{u_{2,2}} & \cdots & \frac{u_{2,n}}{u_{2,2}} \\ 0 & 0 & 1 & \cdots & \frac{u_{3,n}}{u_{3,3}} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & 0 & \cdots & 1 \end{pmatrix} = D \cdot U_c.$$

De este modo, $A = LU = LDU_c = L_cU_c \quad$ donde $\quad L_c = LD$.

2.2.2. Factorización QR

En Álgebra Lineal, la *factorización QR* de una matriz es una descomposición de la misma como producto de una matriz ortogonal Q, ($Q^TQ = I$), por una matriz triangular superior R. La factorización QR es la base del algoritmo QR utilizado para el cálculo de los vectores y valores propios de una matriz que veremos posteriormente en el Capítulo 4.

En cuanto a la unicidad de la descomposición QR de una matriz, al igual que hicimos con la descomposición LU, supongamos que tenemos $A = Q_1R_1 = Q_2R_2$. R_1 y R_2 deben ser no singulares y $Q_2^TQ_1 = R_2R_1^{-1}$. Por la misma razón que en el caso LU, $R_2R_1^{-1}$ es triangular superior, y $Q_2^TQ_1$ es también una matriz ortogonal. Tenemos por tanto que $R_2R_1^{-1}$ es triangular superior y ortogonal, lo que implica que debe ser una matriz diagonal D, que al ser ortogonal, ha de cumplir $D^2 = I$. Se sigue por tanto que $Q_2 = Q_1D$ y $R_2 = DR_1$, con D una matriz diagonal que contiene en sus entradas de

la diagonal principal elementos iguales a 1 ó -1. Lo que nos quiere decir esto es que los signos de los elementos de la diagonal de R pueden elegirse arbitrariamente, y entonces la descomposición QR será única.

Asumiremos que $A \in \mathscr{M}_n$, $|A| \neq 0$ (y por tanto, $|R| \neq 0$).

Obtención de la factorización QR

Hay básicamente tres métodos para obtener la factorización QR de una matriz A: mediante transformaciones de Householder (o reflexiones), transformaciones de Givens (o rotaciones) o mediante el proceso de Gram-Schmidt. Las transformaciones Householder son probablemente el método más comúnmente usado.

Reflexiones Householder para formar la factorización QR

Este método se basa en lo explicado en la Sección 2.1, es decir, en usar los reflectores para calcular la factorización QR, formando en secuencia el reflector para la columna i-ésima que producirá ceros debajo del elemento que ocupa la posición (i,i). Podemos ilustrar el procedimiento considerando el siguiente ejemplo, que hemos resuelto haciendo uso del software Matlab.

Ejemplo 2.2 *Sea la matriz*

$$A = \begin{pmatrix} 3 & 2 & -4 & 3 & 2 \\ 1 & -1 & 1 & 2 & 1 \\ 2 & 4 & 6 & 4 & 1 \\ 1 & 3 & 2 & 1 & 1 \\ 1 & 2 & 8 & 5 & 4 \end{pmatrix}.$$

Comenzamos transformando el vector $(3,1,2,1,1)^T$ *en* $(\pm 4,0,0,0,0)^T$. *Usando las ecuaciones del Ejemplo 2.1 construimos la primera matriz de Householder* P_1, *que en este caso coindice con* H_1. *Obtenemos así la matriz* $P_1 A$*:*

$$\begin{pmatrix} -3.9999999999 & -4.4999999999 & -2.7500000000 & -6.2499999999 & -3.4999999999 \\ 0.0000000000 & -1.9285714285 & 1.1785714285 & 0.6785714285 & 0.2142857142 \\ 0.0000000000 & 2.1428571428 & 6.3571428571 & 1.3571428571 & -0.5714285714 \\ 0.0000000000 & 2.0714285714 & 2.1785714285 & -0.3214285714 & 0.2142857142 \\ 0.0000000000 & 1.0714285714 & 8.1785714285 & 3.6785714285 & 3.2142857142 \end{pmatrix}.$$

Ahora, elegimos un reflector para transformar el vector

$$x_2 = (-1.9285714285,\ 2.1428571428,\ 2.0714285714,\ 1.0714285714)^T$$

donde $\|x_2\|_2 = \frac{\sqrt{55}}{2}$ *en* $(\pm\times,0,0,0)^T$. *Recordar que no queremos perturbar la primera columna en* $P_1 A$, *la segunda matriz de Householder* P_2 *ha de ser de la forma*

$$P_2 = \begin{pmatrix} 1 & 0 & 0 & 0 & 0 \\ 0 & & & & \\ 0 & & H_2 & & \\ 0 & & & & \\ 0 & & & & \end{pmatrix}.$$

Sea el vector

$$u_2 = \frac{q_2^T}{\|q_2\|_2} = \frac{(-1.9285714285 - \frac{\sqrt{55}}{2},\, 2.1428571428,\, 2.0714285714,\, 1.0714285714)^T}{\sqrt{\frac{55}{2} + \frac{27\sqrt{55}}{14}}}.$$

Procediendo del mismo modo obtenemos el reflector $H_2 = I - 2u_2 u_2^T$*:*

$$\begin{pmatrix} -0.5200970367 & 0.5778855963 & 0.5586227431 & 0.2889427981 \\ 0.5778855963 & 0.7803089181 & -0.2123680457 & -0.1098455409 \\ 0.5586227431 & -0.2123680457 & 0.7947108890 & -0.1061840228 \\ 0.2889427981 & -0.1098455409 & -0.1061840228 & 0.9450772295 \end{pmatrix}.$$

Tenemos así que $P_2 P_1 A$ *viene dada por*

$$\begin{pmatrix} -3.999999999 & -4.499999999 & -2.750000000 & -6.249999999 & -3.499999999 \\ 0.000000000 & 3.708099243 & 6.640868645 & 1.314689731 & 0.606779876 \\ 0.000000000 & -0.000000000 & 4.280576158 & 1.115313818 & -0.720640573 \\ 0.000000000 & 0.000000000 & 0.171223620 & -0.555196642 & 0.070047445 \\ 0.000000000 & 0.000000000 & 7.140288079 & 3.557656909 & 3.139679713 \end{pmatrix}.$$

El proceso continúa buscando un reflector para transformar el vector

$$x_3 = (4.280576158,\, 0.171223620,\, 7.140288079)^T, \quad \textit{para el cual} \quad \|x_3\|_2 = \frac{70595}{8478},$$

en $(\pm\times, 0, 0)^T$*. Construimos así la tercera matriz de Householder* P_3*, de la forma*

$$P_3 = \begin{pmatrix} 1 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & & & \\ 0 & 0 & & H_3 & \\ 0 & 0 & & & \end{pmatrix}.$$

Sea

$$u_3 = \frac{q_3^T}{\|q_3\|_2} = \frac{(4.280576158 + \frac{70595}{8478},\, 0.171223620,\, 7.140288079)^T}{\frac{1449}{100}}.$$

Procediendo del mismo modo, obtenemos el reflector $H_3 = I - 2u_3 u_3^T$*:*

$$\begin{pmatrix} -0.514069334239251 & -0.020562842282816 & -0.857502121928608 \\ -0.020562842282816 & 0.999720732417475 & -0.011645887339273 \\ -0.857502121928608 & -0.011645887339273 & 0.514348601821776 \end{pmatrix}.$$

Tenemos así que $P_3P_2P_1A$ viene dada por

$$\begin{pmatrix} 3.999999999 & -4.499999999 & -2.750000000 & -6.249999999 & -3.499999999 \\ 0.000000000 & 3.708099243 & 6.640868645 & 1.314689731 & 0.606779876 \\ -0.000000000 & -0.000000000 & -8.326845959 & -3.612630559 & -2.323263170 \\ 0.000000000 & 0.000000000 & 0.000000000 & -0.619407687 & 0.048281945 \\ -0.000000000 & 0.000000000 & 0.000000000 & 0.879957648 & 2.232024927 \end{pmatrix}.$$

Finalmente, $x_4 = (-0.619407687,\ 0.879957648)^T$, con $\|x_4\|_2 = \frac{7141}{6636}$ ha de transformarse en $(\pm\times, 0)$. La cuarta matriz de Householder P_4 será

$$P_4 = \begin{pmatrix} 1 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & & H_4 \\ 0 & 0 & 0 & & \end{pmatrix}.$$

Tomando

$$u_4 = \frac{q_4^T}{\|q_4\|_2} = \frac{(-0.619407687 - \frac{7141}{6636},\ 0.879957648)^T}{\frac{12388}{6485}}$$

obtenemos el reflector

$$H_4 = I - 2u_4u_4^T = \begin{pmatrix} -0.575604173291874 & 0.817728460852976 \\ 0.817728460852976 & 0.575604173291875 \end{pmatrix}.$$

Obtenemos así finalmente la expresión para $R = P_4P_3P_2P_1A$:

$$R = \begin{pmatrix} -3.999999999 & -4.499999999 & -2.750000000 & -6.249999999 & -3.499999999 \\ 0.000000000 & 3.708099243 & 6.640868645 & 1.314689731 & 0.606779876 \\ -0.000000000 & -0.000000000 & -8.326845959 & -3.612630559 & -2.323263170 \\ -0.000000000 & -0.000000000 & 0.000000000 & 1.076100063 & 1.797399019 \\ -0.000000000 & 0.000000000 & 0.000000000 & 0.000000000 & 1.324244383 \end{pmatrix}.$$

La expresión para Q se obtiene de $Q^T = P_4P_3P_2P_1$:

$$Q = \begin{pmatrix} -0.750000000 & -0.370809924 & 0.432336567 & 0.336281269 & -0.000000000 \\ -0.250000000 & -0.573069883 & -0.494566830 & -0.553645684 & 0.240771706 \\ -0.500000000 & 0.471939903 & -0.179048477 & -0.364548391 & -0.601929265 \\ -0.250000000 & 0.505649896 & 0.245645776 & -0.315811975 & 0.722315118 \\ -0.250000000 & 0.235969951 & -0.689991693 & 0.589710632 & 0.240771706 \end{pmatrix}.$$

Como comprobación, el producto QR da por resultado la matriz A, salvo errores de redondeo:

$$A = QR = \begin{pmatrix} 2.999999999 & 1.999999999 & -3.999999999 & 2.999999999 & 1.999999999 \\ 1.000000000 & -1.000000000 & 0.999999999 & 1.999999999 & 0.999999999 \\ 1.999999999 & 3.999999999 & 5.999999999 & 3.999999999 & 0.999999999 \\ 1.000000000 & 2.999999999 & 2.000000000 & 1.000000000 & 1.000000000 \\ 1.000000000 & 1.999999999 & 7.999999999 & 4.999999999 & 3.999999999 \end{pmatrix}.$$

Se puede demostrar que el número de operaciones para la factorización QR de una matriz en $\mathscr{M}_n$ usando los reflectores de Householder es $2n^3/3$ multiplicaciones y $2n^3/3$ sumas, véase [2, Sección 5.7.1] ó [1, Sección 9.3].

Rotaciones de Givens para formar la factorización QR

Las rotaciones de Givens están basadas en la transformación introducida en la Sección 2.1. Aunque estas transformaciones también permiten la obtención de la factorización QR de una matriz dada, presentan el siguiente inconveniente: cada vez que se multiplica por una matriz de rotación conseguimos hacer un cero, a diferencia de las reflexiones Householder que consiguen hacer ceros en los elementos de cada columna por debajo de la diagonal principal en cada iteración. De este modo, se puede comprobar que el número de operaciones que requiere este procedimiento es aproximadamente el doble que el de Householder, por lo que en la práctica se suele proceder mediante este último.

Ejemplo 2.3 *Hallaremos la descomposición QR de*

$$A = \begin{pmatrix} 2 & 0 & 1 \\ 6 & 2 & 0 \\ -3 & 1 & -1 \end{pmatrix}.$$

El primer paso será obtener el valor 0 *en la posición* $(2,1)$ *de A, utilizando la matriz de Rotación* $G_{2,1}$. *Como*

$$\begin{pmatrix} \cos\theta & -\sin\theta \\ \sin\theta & \cos\theta \end{pmatrix} \begin{pmatrix} 2 & 0 \\ 6 & 2 \end{pmatrix} = \begin{pmatrix} \times & \times \\ 2\sin\theta + 6\cos\theta & \times \end{pmatrix},$$

bastará tomar θ *tal que* $\tan\theta = -3$, *por lo que*

$$G_{2,1} = \begin{pmatrix} \sqrt{10}/10 & 3\sqrt{10}/10 & 0 \\ -3\sqrt{10}/10 & \sqrt{10}/10 & 0 \\ 0 & 0 & 1 \end{pmatrix}, \quad \textit{y de ahí,} \quad G_{2,1}A = A_1 = \begin{pmatrix} 2\sqrt{10} & 3\sqrt{10}/5 & \sqrt{10}/10 \\ 0 & \sqrt{10}/5 & -3\sqrt{10}/10 \\ -3 & 1 & -1 \end{pmatrix}.$$

Para conseguir el valor 0 *en la posición* $(3,1)$ *procedemos de manera similar para obtener la matriz de Givens*

$$G_{3,1} = \begin{pmatrix} \cos\theta & 0 & -\sin\theta \\ 0 & 1 & 0 \\ \sin\theta & 0 & \cos\theta \end{pmatrix} = \begin{pmatrix} 2\sqrt{10}/7 & 0 & -3/7 \\ 0 & 1 & 0 \\ 3/7 & 0 & 2\sqrt{10}/7 \end{pmatrix},$$

y de ahí,

$$A_2 = G_{3,1}A_1 = G_{3,1}G_{2,1}A = \begin{pmatrix} 7 & 9/7 & 5/7 \\ 0 & \sqrt{10}/5 & -3\sqrt{10}/10 \\ 0 & 19\sqrt{10}/35 & -17\sqrt{10}/70 \end{pmatrix}.$$

Para conseguir finalmente el valor 0 *en la posición* $(3,2)$ *se obtiene*

$$G_{3,2} = \begin{pmatrix} 1 & 0 & 0 \\ 0 & \cos\theta & -\sin\theta \\ 0 & \sin\theta & \cos\theta \end{pmatrix} = \begin{pmatrix} 1 & 0 & 0 \\ 0 & 7/\sqrt{4}10 & 19/\sqrt{4}10 \\ 0 & -19/\sqrt{4}10 & 7\sqrt{4}10 \end{pmatrix},$$

y de ahí,

$$R = A_3 = G_{3,2}G_{3,1}G_{2,1}A = \begin{pmatrix} 7 & 9/7 & 5/7 \\ 0 & 2\sqrt{41}/7 & -47\sqrt{41}/287 \\ 0 & 0 & 4\sqrt{41}/41 \end{pmatrix},$$

por lo que $A = QR$*, con*

$$Q^T = G_{3,2}G_{3,1}G_{2,1} = \begin{pmatrix} 2/7 & 6/7 & -3/7 \\ -9/7\sqrt{41} & 22/7\sqrt{41} & 38/7\sqrt{41} \\ 6/\sqrt{41} & -1/\sqrt{41} & 2/\sqrt{41} \end{pmatrix}.$$

Transformaciones Gram-Schmidt para formar la factorización QR

Estas transformaciones están basadas en el proceso de ortogonalización de vectores (1.1) introducidos en el Capítulo 1.

Las transformaciones Gram-Schmidt producen un conjunto de vectores ortonormales que expanden el mismo espacio que el dado por un conjunto de vectores linealmente independientes, $\{x_1, \ldots, x_m\}$. La aplicación de estas transformaciones es llamada *ortogonalización Gram-Schmidt*. Si los vectores linealmente independientes dados son las columnas de una matriz A, estas transformaciones producen finalmente la factorización QR de A.

3

Resolución numérica de sistemas lineales de ecuaciones

En este tercer capítulo abordaremos el condicionamiento y la resolución numérica de sistemas lineales de ecuaciones lineales. El condicionamiento y error de un sistema permite conocer el efecto que puede producir en la solución de éste una perturbación en las entradas de la correspondiente matriz o vector de términos independientes. Abordaremos también la resolución numérica de sistemas con un elevado número de incógnitas mediante métodos directos e iterativos. Como estos conceptos fueron ya abordados en la asignatura Métodos Numéricos *I* del Grado en Matemáticas, haremos especial énfasis en dos apartados que no han sido abordados hasta la fecha: el uso de la descomposición QR en la resolución numérica de un sistema lineal de ecuaciones y el método iterativo de sobre-relajación. En lo que sigue asumiremos que $A \in \mathscr{M}_n$ con $|A| \neq 0$.

3.1. Condicionamiento y error

Antes de comenzar, presentamos a modo de motivación el siguiente ejemplo. Planteamos resolver el sistema $AX = B$, donde

$$A = \begin{pmatrix} 0.9352 & 0.0261 \\ 8.51 & 0.2375 \end{pmatrix}, \quad X = \begin{pmatrix} x_1 \\ x_2 \end{pmatrix} \quad B = \begin{pmatrix} 0.9091 \\ 8.2725 \end{pmatrix}.$$

La solución viene dada por $X = (1,-1)^T$. Supongamos ahora que el vector B queda ligeramente modificado de la siguiente forma:

$$B = \begin{pmatrix} 0,9091 \\ 8.2725 \end{pmatrix} \quad \rightarrow \quad \tilde{B} = \begin{pmatrix} 0.9091 \\ 8.2724 \end{pmatrix}$$

Esa pequeña modificación de una milésima en una de las componentes del vector B hace que la solución $\tilde{X}$ del sistema $A\tilde{X} = \tilde{B}$ cambie drásticamente: $\tilde{X} = (-1.61, 92.52)^T$. Desde un punto de vista numérico, este ejemplo es catastrófico, en

el sentido de que una mínima perturbación en una entrada del sistema (en el vector de términos independientes, en este caso) provoca un cambio drástico en la solución. Nos podríamos preguntar si esto será una problemática generalizada para cualquier sistema lineal de ecuaciones. La respuesta, afortunadamente, será que no. Caracterizamos las matrices *mal condicionadas*, es decir, aquellas en las que ocurre este fenómeno. Las matrices *bien condicionadas* serán por tanto aquellas para las cuales una pequeña perturbación en las entradas de ésta, o del vector de términos independientes, producirán pequeñas perturbaciones en la solución, y por lo tanto, darán lugar a sistemas lineales de ecuaciones estables desde un punto de vista numérico.

Comenzamos suponiendo que se realiza una perturbación únicamente en el vector de términos independientes. Partimos pues del sistema lineal de ecuaciones $AX = B$, y supongamos que el vector B queda perturbado por $\tilde{B}$. Consideremos el sistema perturbado $A\tilde{X} = \tilde{B}$, y sea $\|\cdot\|_v$ una norma vectorial y $\|\cdot\|_M$ la norma matricial inducida. El error relativo cometido en el vector B vendrá dado por $Rel(\tilde{B}) = \frac{\|B-\tilde{B}\|_v}{\|B\|_v}$, y nuestro objetivo será obtener una estimación para $Rel(\tilde{X}) = \frac{\|X-\tilde{X}\|_v}{\|X\|_v}$. Obsérvese que se cumple la relación $\|I_n\|_M = \max_{\|x\|_v=1} \|I_n x\|_v = 1$ y por tanto $1 = \|I_n\|_M = \|A \cdot A^{-1}\|_M \leq \|A\|_M \cdot \|A^{-1}\|_M$. Se define como *número de condición* de una matriz, con respecto a la norma matricial $\|\cdot\|_M$ a la cantidad

$$Cond_M(A) = \|A\|_M \cdot \|A^{-1}\|_M \geq 1. \tag{3.1}$$

De lo anterior obsérve la relación $\|A^{-1}\|_M \geq (\|A\|_M)^{-1}$. El siguiente resultado da respuesta a nuestro problema. En él se establecen cotas para $Rel(\tilde{X})$ en términos de $Rel(\tilde{B})$ y del número de condición de la matriz A.

Teorema 3.1. *En las condiciones anteriores se cumple la relación*

$$\frac{1}{Cond_M(A)} \cdot Rel(\tilde{B}) \leq Rel(\tilde{X}) \leq Cond_M(A) \cdot Rel(\tilde{B}).$$

Los números de condición usuales se corresponden con $\|\cdot\|_p$, tomando $p = 1, 2, \infty$. Tal y como se vió en el Capítulo 1, $r_\sigma(A)$ juega un papel similar al de norma matricial, cumpliéndose $r_\sigma(A) \leq \|A\|_M$ para toda $\|\cdot\|_M$ norma matricial.

Establecemos pues la siguiente

Definición 3.2. *Se define el número de condición de una matriz según la cantidad*

$$cond(A) = r_\sigma(A) \cdot r_\sigma(A^{-1}).$$

De esta definición, junto a las consideraciones anteriores, se deduce que $cond(A) \leq cond_M(A)$, para toda norma matricial $\|\cdot\|_M$. Tenemos pues, del Teorema 3.1, que $cond(A)$ vuelve a ser una medida fiable en la estimación de $Rel(\tilde{X})$, que es independiente de la norma matricial elegida. Dado que

$$\lambda \in \mathbb{C} \setminus \{0\} \quad \text{es un autovalor de A} \Leftrightarrow \lambda^{-1} \in \mathbb{C} \setminus \{0\} \quad \text{es un autovalor de} A^{-1},$$

podemos establecer la siguiente regla para el cálculo de $cond(A)$:

$$cond(A) = \left(\max_{\lambda \in \sigma(A)} |\lambda| \right) \cdot \left(\max_{\tilde{\lambda} \in \sigma(A^{-1})} |\tilde{\lambda}| \right) = \frac{\text{máx}_{\lambda \in \sigma(A)} |\lambda|}{\text{mín}_{\lambda \in \sigma(A)} |\lambda|} \geq 1.$$

Es decir, una matriz estará bien condicionada si sus autovalores están próximos entre sí en módulo, mientras que estará mal condicionada cuando la diferencia entre los módulos del autovalor más grande y el más pequeño es grande. Con todo esto establecemos formalmente la siguiente definición: Una matriz se dice que está *bien condicionada* si su número de condición es $\approx 1 (\geq 1)$. En caso contrario, se dirá que está *mal condicionada* ($cond(A) \gg 1$).

Tal y como hemos visto en el Capítulo 1, podemos afirmar que las matrices ortogonales o unitarias son óptimas desde el punto de vista de la estabilidad de sistemas de ecuaciones.

3.2. Métodos directos e iterativos

Métodos directos

Los sistemas de ecuaciones lineales aparecen en problemas relacionados con un largo número de áreas: Matemáticas, Física, Biología, Química, Economía, Ciencias Sociales, etc. Aparecen de manera directa, por ejemplo, en problemas de modelización física, y de manera indirecta, en la solución numérica de otros problemas de modelización matemática. El tipo de problema más común es resolver un sistema lineal de n ecuaciones con n incógnitas $AX = B$, donde n puede ser un valor relativamente grande.

La mayoría de los algoritmos conocidos para matrices que son en general densas se basan en la *eliminación Gaussiana*. Como hemos visto en el Capítulo 1, dicho método está basado en emplear transformaciones elementales a la matriz A. Es fácil comprobar (véase [1, Sección 8.1]) que el coste computacional que conlleva este algoritmo es de aproximadamente $n^3/3$ multiplicaciones, una complejidad algorítmica mucho más económica si lo comparamos, por ejemplo, con el clásico método de Cramer, que requiere $(n+1)! \cdot (n-1)$ multiplicaciones.

Durante el proceso de la eliminación Gaussiana no es suficiente preguntarnos si el elemento pivote es o no, en cada paso, una entrada nula. Un elemento pivote podría estar próximo a cero, o ser cero salvo errores de redondeo. Los multiplicadores $m_{i,j}$ podrían ser en este caso cantidades muy grandes, y por tanto, los errores podrían propagarse durante el proceso. Para evitar este fenómeno se suelen emplear dos tipos de pivotación.

Pivotación parcial: en el paso $k-$ésimo de la eliminación Gaussiana (recuérdese 2.5), con $1 \leq k \leq n-1$, definamos

$$c_k = \max_{k \le i \le n} |a_{i,k}^{(k)}|.$$

Sea i el índice más pequeño por filas para el cual se alcanza este máximo, donde $i \ge k$. Si $i = k$, entonces no se realiza ninguna acción. Sin embargo, si $i > k$, se intercambian las filas i y $k-$ésimas. De este modo se garantiza que $|m_{i,k}| \le 1$, para todo $i = k+1, \ldots, n$. Esto previene que los elementos (entradas) de la matriz $A^{(k)}$ crezcan demasiado, disminuyendo por tanto la posibilidad de una gran pérdida de significación de errores.

Pivotación completa: en el paso $k-$ésimo de la eliminación Gaussiana, con $1 \le k \le n-1$, definamos

$$c_k = \max_{k \le i,j \le n} |a_{i,j}^{(k)}|.$$

El objetivo en este caso será conseguir que el elemento pivote pase a ser el elemento mayor en módulo de la submatriz formada por los elementos $(a_{i,j}^{(k)})_{i,j=k}^{n}$. En este caso se intercambiarían filas en la matriz A y en el vector B, pero también columnas en la matriz A y en el vector X que contiene a las incógnitas del problema.

En virtud de los problemas prácticos que conlleva la pivotación completa se concluye que el comportamiento de los errores viene a ser esencialmente el mismo que para la pivotación completa. Como el tiempo de ejecución el algoritmo de la eliminación con pivotación completa es elevado, siendo la estrategia menos económica en este sentido, en la mayoría de las situaciones prácticas se emplea pivotación parcial, y no completa.

Por otro lado, se ha observado experimentalmente que si las entradas de una matriz varían mucho en tamaño, se producirá una mayor pérdida de significación en los errores, empeorando la propagación de errores de redondeo. En la práctica se suele emplear eliminación Gaussiana con pivotación parcial y escalado implícito, cuya finalidad es escalar las entradas de la matriz, multiplicando filas y columnas por constantes apropiadas de manera que el elemento pivote sea el elemento de la columna más grande, pero en relación con los otros elementos no nulos de la fila. Debemos decir que el proceso de *escalamiento* no es entendido en la actualidad de manera rigurosa, en el sentido de que no se conoce cómo garantizar al 100 % que el efecto de los errores de redondeo en la eliminación Gaussiana disminuyan tras el escalado.

Si denotamos por $\tilde{A}$ al resultado de escalar una fila y una columna de la matriz A, entonces $\tilde{A} = D_1 A D_2$, con D_i matrices diagonales para $i = 1, 2$, que contienen las cantidades de escalado. Para resolver $AX = B$ obsérvese que $D_1 A D_2 (D_2^{-1} X) = D_1 B$, por lo que podemos obtener X resolviendo $\tilde{A} z = D_1 B$ y $X = D_2 z$.

Supongamos que realizamos solamente operaciones de escalamiento por filas. Podríamos plantear elegir los coeficientes de la matriz de manera que al escalarlos se cumpla la condición $\max_{1 \le j \le n} |\tilde{a}_{i,j}| \approx 1, \forall\, i = 1, \ldots, n$, donde $\tilde{A}$ es el resultado de escalar la matriz A. Esto podría realizarse definiendo

$$s_i = \max_{1 \le j \le n} |a_{i,j}|, \forall\, i = 1, \ldots, n \quad \text{y} \quad \tilde{a}_{i,j} = \frac{a_{i,j}}{s_i}, \forall\, j = 1, \ldots, n.$$

Pero esto tiene un inconveniente, y es que estamos introduciendo errores adicionales de redondeo en cada una de las entradas de la matriz A. Por esa razón se suele emplear usualmente la siguiente técnica. En el paso $k-$ésimo de la eliminación Gaussiana reemplazaremos en la pivotación parcial el valor c_k por el valor

$$c_k = \max_{k\le i\le n} \frac{|a_{i,k}^{(k)}|}{|s_i^{(k)}|}, \quad \text{donde} \quad s_i^{(k)} = \max_{k\le j\le n} |a_{i,j}^{(k)}|, \quad i = k,\dots,n.$$

De este modo, elegimos el índice más pequeño para el que se alcanza la cantidad c_k y reemplazamos esa fila por la $k-$ésima, si no fuera ésta.

Destaquemos la importancia de tener la factorización $A = LU$ en tres aplicaciones:

1. Resolución de sistemas mediante sustitución hacia adelante y hacia atrás. Una vez hemos obtenido la descomposición $A = LU$, la resolución del sistema $AX = LUX = B$, puede realizarse en dos pasos inmediatos:
   a) Resolver $LY = B$, obteniendo Y por sustitución hacia adelante.
   b) Resolver $UX = Y$, obteniendo así la solución X del sistema mediante sustitución hacia atrás.
2. Resolución de varios sistemas de ecuaciones con la misma matriz A en común. Supongamos que queremos resolver m sistemas de n ecuaciones con n incógnitas de de la forma $A\cdot X_l = B_l$, con $l = 1,2,\dots,m$. La factorización $A = LU$ será por tanto común, y podremos aprovecharla para resolver así cada uno mediante sustitución hacia adelante y sustitución hacia atrás en cada uno de ellos, reduciendo considerablemente el número de operaciones a realizar en la resolución de los m sistemas.
3. Variantes de la eliminación Gaussiana: si la matriz A tiene ciertas estructuras, la descomposición LU puede dar lugar a métodos especiales, como por ejemplo el algoritmo de Thomas (basado en la descomposición de Crout de la matriz A, siendo ésta tridiagonal) o el método de Cholesky (aplicable para matrices A definidas positivas, verificándose en este caso $A = L\cdot L^T$, con $l_{ii} > 0, \forall\ i = 1,\dots,n$).

Finalmente, nótese que si alternativamente, hemos obtenido la descomposición $A = QR$ estudiada en el Capítulo 2, entonces podemos resolver el sistema $AX = QRX = B$ en dos pasos, también de manera inmediata:

1. Resolvemos $QY = B$, obteniendo $Y = Q^T B$.
2. Resolvemos $RX = Y$, obteniendo X por sustitución hacia atrás.

Métodos iterativos

Como hemos visto en la sección anterior, el método directo de la eliminación Gaussiana requiere de aproximadamente $n^3/3$ multiplicaciones, siendo un algoritmo con un coste computacional mucho más económico que, por ejemplo, el clásico

método de Cramer. No obstante, para valores de n muy grandes, digamos, mayores que 10^6, este proceso sigue requiriendo un elevado coste computacional, y es por ello por lo que en estos casos se prefiere el uso de métodos indirectos o iterativos.

Sea $A = (a_{i,j})_{i,j=1}^n \in \mathscr{M}_n$ y presentemos a continuación el *esquema general* de los métodos iterativos. Queremos resolver el sistema lineal $AX = B$, con $A \in \mathscr{M}_n$, $|A| \neq 0$, $X, B \in \mathbb{C}^n$. La idea general es elegir una iteración inicial $X^{(0)} \in \mathbb{C}^n$ para a continuación plantear un esquema iterativo en la forma

$$X^{(\nu+1)} = g(X^{(\nu)}), \quad \forall \nu \geq 0, \quad \text{siendo} \quad g : \mathbb{C}^n \to \mathbb{C}^n.$$

El objetivo será obtener funciones g que permitan garantizar, bajo ciertas condiciones en la matriz A, que $\lim_{\nu\to\infty} X^{(\nu)} = X$, para todo $X^{(0)} \in \mathbb{C}^n$. Para obtener la función g, escribamos la matriz A en la forma $A = N - P$, donde N es una matriz tal que $|N| \neq 0$. Tenemos pues que

$$N \cdot X = P \cdot X + B, \tag{3.2}$$

lo que permite establecer el siguiente esquema iterativo:

$$N \cdot X^{(\nu+1)} = P \cdot X^{(\nu)} + B. \tag{3.3}$$

Los diferentes esquemas iterativos se diferenciarán en las diversas elecciones particulares que se hagan para las matrices N y P. Definamos el vector error en la iteración n-ésima según

$$e^{(\nu)} = X - X^{(\nu)} \in \mathbb{C}^n, \quad \text{para todo} \quad \nu = 0, 1, 2, \ldots$$

Las ecuaciones (3.2) y (3.3) permiten escribir

$$N \cdot e^{(\nu+1)} = P \cdot e^{(\nu)} \;\Rightarrow\; e^{(\nu+1)} = M \cdot e^{(\nu)}, \quad \text{donde} \quad M = N^{-1} \cdot P.$$

La matriz M en la relación juega un papel clave, nótese que ésta es conocida de antemano, antes de iniciar el proceso. A través de ella podremos realizar un control del error, dado que

$$e^{(\nu)} = M \cdot e^{(\nu-1)} = M^2 \cdot e^{(\nu-2)} = \cdots = M^\nu \cdot e^{(0)}.$$

Sea $\|\cdot\|_\nu$ una norma vectorial y $\|\cdot\|_M$ una norma matricial compatible con $\|\cdot\|_\nu$. Se tiene entonces que

$$\|e^{(\nu)}\|_\nu = \|M^\nu \cdot e^{(0)}\|_\nu \leq \|M^\nu\|_M \cdot \|e^{(0)}\|_\nu \leq \|M\|_M^\nu \cdot \|e^{(0)}\|_\nu,$$

donde en la primera desigualdad hemos usado la compatibilidad, y en la segunda desigualdad la quinta propiedad en la definición de norma matricial. Haciendo ahora uso del Corolario 1.11, tenemos que

$$\lim_{\nu\to\infty} M^\nu = 0 \quad \Leftrightarrow \quad r_\sigma(M) < 1.$$

Hemos probado por tanto el siguiente

Teorema 3.3. *El esquema iterativo (3.3) convergerá a la solución X del sistema $AX = B$ con $|A| \neq 0$, para cualquier iteración inicial $X^{(0)}$ que se considere, si y sólo si, $r_\sigma(M) < 1$ con $M = N^{-1} \cdot P$ siendo $A = N - P$ con $|N| \neq 0$.*

Los dos métodos iterativos más conocidos en la literatura son los procesos de *Jacobi* y de *Gauss-Seidel*, que se van a corresponder con las elecciones para la matriz N que parecen resultar más intuitivas o razonables. Consideremos primero como matriz N la diagonal principal de A:

$$N = \text{diag}(a_{1,1}, \cdots, a_{n,n}) = \left(a_{i,j} \cdot \delta_{i,j}\right)_{i,j=1}^{n},$$

donde $\delta_{i,j}$ es el símbolo delta de Kronecker, definido según $\delta_{i,j} = \begin{cases} 1 \text{ si } i = j \\ 0 \text{ si } i \neq j \end{cases}$. La matriz P tal que $A = N - P$ vendrá dada pues por $P = N - A = a_{i,j} \cdot (\delta_{i,j} - 1)$. Asumamos que $a_{i,i} \neq 0$ para todo $i = 1, \ldots, n$ (en caso contrario, la matriz N será singular; bastará pues con realizar previamente intercambio de filas en la matriz A par asegurar esta condición). Tenemos entonces que el esquema iterativo quedará de la forma

$$x_i^{(\nu)} = \frac{1}{a_{i,i}} \left\{ b_i - \sum_{j=1, j\neq i}^{n} a_{i,j} x_j^{(\nu-1)} \right\}, \quad \forall i = 1, \ldots, n \quad \forall \nu \geq 1, \tag{3.4}$$

donde

$$X^{(\nu)} = (x_1^{(\nu)}, \cdots, x_n^{(\nu)})^T \quad \text{y} \quad X^{(0)} = (x_1^{(0)}, \cdots, x_n^{(0)})^T.$$

Este método iterativo recibe el nombre de *método de Jacobi*, o de *iteraciones simultáneas*, dado que consiste en "despejar"la incógnita x_i en la fila $i-$ésima del sistema, y luego, partiendo de la iteración inicial, obtener de manera simultánea las componentes de un nuevo vector en la iteración sustituyendo las incógnitas x_j, $j = 1, \ldots, n$ $(j \neq i)$ por los valores de las mismas en el vector en la iteración anterior.

Caracterizar los sistemas para los cuales el método es convergente para cualquier iteración inicial $X^{(0)}$ que se considere es muy complicado. Podemos dar condiciones suficientes prácticas para garantizarlo, si procedemos analizando la estructura de la matriz M que resulta. Dado que la inversa de una matriz diagonal consiste en la matriz diagonal que contiene a los elementos inversos, se sigue que

$$M = N^{-1}P = \begin{pmatrix} 0 & -\frac{a_{1,2}}{a_{1,1}} & -\frac{a_{1,3}}{a_{1,1}} & \cdots & -\frac{a_{1,n}}{a_{1,1}} \\ -\frac{a_{2,1}}{a_{2,2}} & 0 & -\frac{a_{2,3}}{a_{2,2}} & \cdots & -\frac{a_{2,n}}{a_{2,2}} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ -\frac{a_{n,1}}{a_{n,n}} & -\frac{a_{n,2}}{a_{n,n}} & -\frac{a_{n,3}}{a_{n,n}} & \cdots & 0 \end{pmatrix}.$$

El Teorema 3.3 establece que el método será convergente, para cualquier iteración inicial que se considere, si y sólo si, $r_\sigma(M) < 1$. Haciendo uso del Corolario 1.11, tomando en particular la norma matricial $\|\cdot\|_\infty$ vemos que

$$\|M\|_\infty = \max_{1\le i\le n} \sum_{j=1}^{n} |m_{i,j}| = \max_{1\le i\le n} \sum_{j=1, j\neq i}^{n} \left|\frac{a_{i,j}}{a_{i,i}}\right|. \tag{3.5}$$

Podemos ahora detallar la siguiente

Definición 3.4. *Una matriz $A = (a_{i,j}) \in \mathcal{M}_n$ se dice que es (estrictamente) diagonal dominante si para todo $1 \le i \le n$ se cumple $|a_{i,i}| > \sum_{j=1, j\neq i}^{n} |a_{i,j}|$.*

Vemos pues que si la matriz A del sistema es diagonal dominante (algo que es inmediato comprobar), entonces el método de Jacobi será convergente, independientemente de la iteración inicial elegida (de ahí que digamos que es una condición suficiente práctica). Obviamente, estas condiciones para garantizar la convergencia del método de Jacobi son solamente suficientes. Es decir, podemos encontrar una matriz A que no sea diagonal dominante, para la cual la matriz M asociada al método de Jacobi cumpla $r_\sigma(M) < 1$.

El método de Jacobi (3.4) puede expresarse en la forma

$$x_i^{(\nu)} = g_i(x_1^{(\nu-1)}, \cdots, x_{i-1}^{(\nu-1)}, x_{i+1}^{(\nu-1)}, \cdots, x_n^{(\nu-1)}),$$

computando cada componente de la iteración $\nu-$ésima en términos de las componentes de la iteración $(\nu-1)-$ésima. Supongamos ahora que comenzamos el proceso en la iteración $\nu-$ésima. Tras computar la primera componente $x_1^{(\nu)}$, procedemos a calcular $x_2^{(\nu)}$ en términos de $x_i^{(\nu-1)}$, $\quad i = 1, 3, 4, \ldots, n$. Sin embargo, debemos observar que, si el método es convergente, entonces $x_1^{(\nu)}$ debe ser una mejor aproximación a x_1 que $x_1^{(\nu-1)}$, que es la información que está empleando el método de Jacobi en su codificación. Parece razonable pues, para el cálculo de $x_2^{(\nu)}$, hacer uso de $x_1^{(\nu)}$ en vez de $x_1^{(\nu-1)}$. Con el mismo razonamiento, en la componente $i-$ésima parecerá más razonable hacer uso de las cantidades $x_j^{(\nu)}$ con $j = 1, \ldots, i-1$, recién calculadas en la presente iteración, en vez de $x_j^{(\nu-1)}$. Es decir, plantear la misma idea del método de Jacobi pero expresándolo ahora en la forma

$$x_i^{(\nu)} = g_i(x_1^{(\nu)}, \cdots, x_{i-1}^{(\nu)}, x_{i+1}^{(\nu-1)}, \cdots, x_n^{(\nu-1)}).$$

Este método, conocido como de "iteraciones sucesivas", dado que aprovecha la propia información que el proceso está calculando en la misma iteración, componente a componente, se conoce como *método de Gauss-Seidel*. Con esta idea, el esquema iterativo (3.4) quedará en la forma [*]

$$x_i^{(\nu)} = \frac{1}{a_{i,i}} \left\{ b_i - \sum_{j=1}^{i-1} a_{i,j} x_j^{(\nu)} - \sum_{j=i+1}^{n} a_{i,j} x_j^{(\nu-1)} \right\}, \forall\, i = 1, \ldots, n \quad \forall \nu \ge 1, \tag{3.6}$$

y es fácil darse cuenta que en la formulación general lo que estamos considerando como matriz N es la matriz triangular inferior de la matriz A.

[*] Como es habitual, para $i = 1$ se entiende que $\sum_{j=1}^{0} a_{i,j} \cdot x_j^{(\nu)} = 0$.

Al igual que sucede con el método de Jacobi, es fácil comprobar (véase [1, Sección 8.6]) que si A es estrictamente diagonal dominante, entonces el método de Gauss-Seidel será también convergente, independientemente de la iteración inicial elegida. Además, estas condiciones son suficientes, pero no necesarias, siendo fácil encontrar ejemplos de matrices A que no son estrictamente diagonal dominantes para las cuales el método de Gauss-Seidel es convergente. Podemos resumir con el siguiente

Teorema 3.5. *Sea $A \in \mathscr{M}_n$ una matriz estrictamente diagonal dominante y consideremos el sistema lineal de ecuaciones $A \cdot X = B$. Entonces, para cualquier iteración inicial $X^{(0)}$ que se considere,*

- *El método de Jacobi converge a la solución X.*
- *El método de Gauss-Seidel converge a la solución X.*

Para finalizar con esta sección, podría deducirse, de la propia construcción de estos dos métodos iterativos, que el método de Gauss-Seidel convergerá siempre que lo haga el de Jacobi. Esto resulta ser falso, pues puede comprobarse que no existe relación alguna sobre la convergencia de ambos procesos, como podemos ver con el siguiente

Ejemplo 3.1 *Consideramos las siguientes cuatro matrices no diagonalmente dominantes:*

$$A_1 = \begin{pmatrix} 5 & 3 & 3 \\ 2 & 6 & 4 \\ 4 & 1 & 10 \end{pmatrix}, \qquad A_2 = \begin{pmatrix} 1 & 3 & 5 \\ 2 & 4 & 6 \\ 10 & 1 & 10 \end{pmatrix}, \qquad A_3 = \begin{pmatrix} 4 & 2 & 3 \\ 5 & -2 & 1 \\ 3 & 2 & 5 \end{pmatrix}, \qquad A_4 = \begin{pmatrix} 1 & 2 & -2 \\ 1 & 1 & 1 \\ 2 & 2 & 1 \end{pmatrix}.$$

Entonces,

- *Para A_1 tanto el método de Jacobi como el de Gauss-Seidel son convergentes.*
- *Para A_2, ni el método de Jacobi ni el de Gauss-Seidel son convergentes.*
- *Para A_3, el método de Gauss-Seidel converge, mientras que el de Jacobi no.*
- *Para A_4, el método de Jacobi converge, mientras que el de Gauss-Seidel no.*

3.3. Métodos de sobre-relajación

La mayoría de los métodos iterativos tienen un comportamiento regular en el decrecimiento de errores. Esto puede usarse con frecuencia para acelerar la convergencia. Nos limitaremos a describir un proceso de aceleración para el método de Gauss-Seidel dado por (3.6), que es empleado en la mayoría de las situaciones prácticas. Consideremos la siguiente modificación del método, introduciendo un parámetro de aceleración ω :

$$z_i^{(\nu+1)} = \frac{1}{a_{i,i}} \left\{ b_i - \sum_{j=1}^{i-1} a_{i,j} x_j^{(\nu+1)} - \sum_{j=i+1}^{n} a_{i,j} x_j^{(\nu)} \right\}, \forall i = 1, \ldots, n, \quad \nu \geq 0,$$
$$x_j^{(\nu+1)} = \omega z_j^{(\nu+1)} + (1-\omega) x_j^{(\nu)}. \tag{3.7}$$

Es decir, como iteración $x_j^{(\nu+1)}$ estaremos considerando una combinación lineal entre el valor $z_j^{(\nu+1)}$ que se obtendría directamente del método de Gauss-Seidel y el valor $x_j^{(\nu)}$ correspondiente al valor obtenido para esa componente en la iteración anterior. Cuando tomamos valores entre $0 < \omega < 1$ el resultado es un promedio que se conoce como *subrelajación* (convergencia más lenta). Si tomamos el valor $\omega = 1$ recuperamos el método de Gauss-Seidel. Y, si el parámetro $\omega \in (1,2]$ nos dejará elegirlo de manera adecuada de modo que nos permita acelerar la convergencia, método conocido como *sobre-relajación sucesiva* ó *SOR(ω)* (convergencia más rápida).

Para comprender cómo elegir el parámetro ω re-escribimos el método (3.7) en forma matricial, descomponiendo la matriz A en la forma $A = D + L + U$, donde $D = diag(a_{1,1}, \ldots, a_{n,n})$ y L y U son las matrices triangular inferior y superior de A, respectivamente. El método (3.7) queda por tanto re-escrito en la forma

$$\begin{aligned} Z^{(\nu+1)} &= D^{-1} \left[B - L X^{(\nu+1)} - U X^{(\nu)} \right], \quad \forall \nu \geq 0, \\ X^{(\nu+1)} &= \omega Z^{(\nu+1)} + (1-\omega) X(\nu). \end{aligned} \tag{3.8}$$

Eliminando $Z^{(\nu+1)}$ y resolviendo para $X^{(\nu+1)}$ se deduce

$$\left[I_n + \omega D^{-1} L \right] X^{(\nu+1)} = \omega D^{-1} B + \left[(1-\omega) I_n - \omega D^{-1} U \right] X^{(\nu)},$$

y para la estimación del error,

$$e^{(\nu+1)} = M(\omega) e^{(\nu)}, \quad \forall \nu \geq 0, \quad M(\omega) = \left[I_n + \omega D^{-1} L \right]^{-1} \cdot \left[(1-\omega) I_n - \omega D^{-1} U \right].$$

La elección óptima del parámetro ω será aquella que haga que se minimice $r_\sigma(M(\omega))$, de manera que se consiga que $X^{(\nu)}$ converja más rápidamente a X. El valor óptimo ω^* no es fácil de calcular, salvo en las situaciones más simples. Normalmente, este valor solo se obtiene de manera estimada, basándose en realizar varios intentos para diversas elecciones de ω, y observando el efecto en la velocidad de convergencia.

A continuación estableceremos algunos criterios de convergencia detallados para los métodos de Jacobi, Gauss-Seidel y $SOR(\omega)$. El primero de ellos coincide con el Teorema 3.5, donde además se establece que bajo tales hipótesis, el método de Gauss-Seidel converge más rápidamente que el de Jacobi. En los dos resultados siguientes se imponen condiciones más fuertes a la matriz A, pero nos dan más información acerca de la convergencia. Más concretamente,

- Si A es estrictamente diagonal dominante los métodos de Jacobi y Gauss-Seidel convergen, y el método de Gauss-Seidel es más rápido (Teorema 3.6).

- Respecto a $SOR(\omega)$, demostraremos que $0 < \omega < 2$ es una condición necesaria para la convergencia (Teorema 3.7). Si A es además definida positiva, $0 < \omega < 2$ también es una condición suficiente para la convergencia (Teorema 3.8).

En lo que sigue vamos a emplear la siguiente notación alternativa: $M_J = L+U$, donde $A = D+\tilde{L}+\tilde{U} = D\cdot(I+L+U)$, siendo $\tilde{L} = D\cdot L$ la parte triangular inferior de A y $\tilde{U} = D\cdot U$ la parte superior de A. Igualmente, $M_{GS} = (I-L)^{-1}\cdot U$.

Teorema 3.6. *Si A es estrictamente diagonal dominante, los métodos de Jacobi y Gauss-Seidel convergen. En efecto $\|M_{GS}\|_\infty \le \|M_J\|_\infty < 1$.*

Demostración. Queremos probar que

$$\|M_{GS}\|_\infty = \||M_{GS}|e\|_\infty \le \||M_J|e\|_\infty = \|M_J\|_\infty, \tag{3.9}$$

donde $e = (1,\dots,1)^T$ es el vector unidad. La desigualdad (3.9) será cierta si demostramos la siguiente desigualdad más fuerte

$$|(I-L)^{-1}U|\cdot e = |M_{GS}|\cdot e \le |M_J|\cdot e = (|L|+|U|)\cdot e. \tag{3.10}$$

En efecto, de la desigualdad triangular y dado que $\|L\|_2 < 1$, se sigue que

$$|(I-L)^{-1}U|\cdot e \le |(I-L)^{-1}|\cdot|U|\cdot e = \left|\sum_{i=0}^{\infty} L^i\right|\cdot|U|\cdot e \le \sum_{i=0}^{\infty}|L|^i\cdot|U|\cdot e = (I-|L|)^{-1}\cdot|U|\cdot e.$$

Ahora, la desigualdad (3.10) será cierta si probamos que $(I-|L|)^{-1}\cdot|U|\cdot e \le (|L|+|U|)\cdot e$. Como todas las entradas de $(I-|L|)^{-1} = \sum_{i=0}^{\infty}|L|^i$ son no negativas, esto equivale a probar $|U|\cdot e \le (I-|L|)\cdot(|L|+|U|)\cdot e = (|L|+|U|-|L|^2-|L|\cdot|U|)\cdot e$, es decir,

$$0 \le (|L|-|L|^2-|L|\cdot|U|)\cdot e = |L|\cdot(I-|L|-|U|)\cdot e.$$

Como las entradas de $|L|$ son no negativas, esta desigualdad es cierta si $0 \le (I-|L|-|U|)\cdot e$ ó $|M_J|\cdot e = (|L|+|U|)e \le e$, lo cual se cumple por la hipótesis $\||M_J|\cdot e\|_\infty = \|M_J\|_\infty = r_\sigma(M_J) < 1$. □

Definamos la matriz correspondiente para la convergencia del método $SOR(\omega)$ como $M_{SOR(\omega)} = (I-\omega L)^{-1}((1-\omega)I+\omega U)$.

Teorema 3.7. $r_\sigma(M_{SOR(\omega)}) \ge |\omega-1|$. *Por tanto, $SOR(\omega)$ converge si $0 < \omega < 2$.*

Demostración. Escribimos el polinomio característico de $M_{SOR(\omega)}$ como $\varphi(\lambda) = |\lambda I - M_{SOR(\omega)}| = |(\lambda+\omega-1)I-\omega\lambda L-\omega U|$ por lo que

$$\varphi(0) = \pm\prod_{i=1}^{n}\lambda_i(R_{SOR(\omega)}) = \pm|(\omega-1)I| = \pm(\omega-1)^n,$$

implicando que $\max_i|\lambda_i(M_{SOR(\omega)})| \ge |\omega-1|$. □

Teorema 3.8. *Si A es definida positiva, entonces $r_\sigma(M_{SOR(\omega)}) < 1$ para todo $0 < \omega < 2$, de este modo $SOR(\omega)$ converge para todo $0 < \omega < 2$. Tomando $\omega = 1$, vemos que el método de Gauss-Seidel también converge.*

Demostración. Se prodece mediante dos pasos, reescribimos $M_{SOR(\omega)} = R$. Sea $M = \omega^{-1}(D - \omega\tilde{L})$, donde $\tilde{L} = DL$. Entonces

1. Definimos $Q = A^{-1}(2M - A)$ y demostramos que $\Re(\lambda_i(Q)) > 0$ para todo i.
2. Se demuestra que $R = (Q - I)(Q + I)^{-1}$, implicando que $|\lambda_i(R)| < 1$ para todo i.

Para el paso 1, nótese que $Qx = \lambda x$ implica $(2M - A)x = \lambda Ax$ ó $x^H(2M - A)x = \lambda x^H Ax$. Tomando traspuesta conjugada a esta última relación obtenemos $x^H(M + M^H - A)x = \Re(\lambda)(x^H Ax)$, y así $\Re(\lambda) = x^H(M + M^H - A)x/x^H Ax = x^H(\frac{2}{\omega} - 1)Dx/x^H Ax > 0$ por ser A y $(\frac{2}{\omega} - 1)D$ matrices definidas positivas.

El paso 2 se sigue directamente de la relación $(Q - I)(Q + I)^{-1} = (2A^{-1}M - 2I)(2A^{-1}M)^{-1} = I - M^{-1}A = R$.

□

Juntos, los Teoremas 3.7 y 3.8 implican que si A es simétrica y definida positiva, entonces $SOR(\omega)$ converge, si y solo si, $0 < \omega < 2$ (Teorema de Ostrowski-Reich). Ilustramos estos resultados presentando el siguiente

Ejemplo 3.2 *Resolvamos por el método $SOR(\omega)$ el sistema de ecuaciones lineales*

$$\begin{cases} 4x_1 + 3x_2 = 24, \\ 3x_1 + 4x_2 - x_3 = 30, \\ -x_2 + 4x_3 = -24. \end{cases}$$

Aplicando la relación general de recurrencia del método, partiendo del punto inicial $x^{(0)} = (1,1,1)^T$, con $\omega = 1.25$ se obtienen los resultados de la siguiente tabla Desde el

k	0	1	2	3	4	5	6	7
$x_1^{(k)}$	1.0000	6.3125	2.6223	3.1333	2.9570	3.0037	2.9963	3.0000
$x_2^{(k)}$	1.0000	3.5195	3.9585	4.0102	4.0074	4.0029	4.0009	4.0002
$x_3^{(k)}$	1.0000	−6.6501	−4.6004	−5.0966	−4.9734	−5.0057	−4.9982	−5.0003

mismo punto de partida con $\omega = 2.25$, el proceso diverge:

k	0	1	2	3	4	5	6	7
$x_1^{(k)}$	1.0000	10.5625	3.0588	1.3328	−10.8367	12.2136	10.9919	18.5938
$x_2^{(k)}$	1.0000	−1.6367	4.9442	13.4344	8.7895	−7.5608	−11.1607	11.9961
$x_3^{(k)}$	1.0000	−15.6706	8.8695	−17.0300	12.7316	−33.6674	22.3064	−34.6352

4

Cálculo numérico de autovalores y autovectores

El objetivo de este último capítulo es el de presentar una breve introducción al problema de calcular numéricamente el conjunto de autovalores y autovectores de una matriz dada $A \in \mathscr{M}_n$. Este problema fue abordado en los contenidos de la última parte de la asignatura *Métodos numéricos I* del Grado en Matemáticas, de manera muy superficial. Más concretamente, fueron estudiados el conocido *Teorema de Gershgorin*, como método de localización de autovalores, y el *método de potencias*, como algoritmo iterativo para la determinación de los mismos.

El método de potencias tiene ciertos inconvenientes. En primer lugar, debemos decir que solo es aplicable si la matriz A es diagonalizable. Un teorema de caracterización para matrices diagonalizables fue demostrado en el Capítulo 1, Teorema 1.3. Si la dimensión de la matriz A es grande, comprobar esto puede convertirse en un problema verdaderamente costoso. Si la matriz A fuera hermitiana, esta condición la tendríamos garantizada, y es cierto que las matrices hermitianas aparecen con suma frecuencia en problemas de la Matemática Aplicada, pero no cabe duda que esta hipótesis de partida es una restricción verdaderamente importante. Además, la matriz A debe poseer un autovalor dominante con multiplicidad algebraica igual a 1, es decir,

$$\sigma(A) = \{\lambda_i\}_{i=1}^n, \quad \text{con} \quad |\lambda_1| > |\lambda_2| \geq |\lambda_3| \geq \cdots \geq |\lambda_n|.$$

Este método tiene además el inconveniente de que, aún cumpliéndose las hipótesis comentadas, podríamos tener la mala suerte de arrancar el proceso con una iteración inicial para la cual el método no converge, lo que requeriría en esta situación el volver a probar suerte, partiendo de otro vector, hasta que se observe empíricamente la convergencia del proceso. Debemos comentar también que el método de potencias es un proceso que permite la obtención aproximada del autovalor dominante λ_1 y un autovector asociado a éste, por lo que, para poder obtener el espectro completo $\sigma(A)$ debemos proceder, tras haber estimado λ_1, y bajo la suposición de que λ_2 es un autovalor subdominante de A, es decir,

$$|\lambda_1| > |\lambda_2| > |\lambda_3| \geq \cdots \geq |\lambda_n|,$$

mediante un *proceso de deflación* (deflación de Wielandt) para reducir el problema a uno de dimensión $n-1$, volver a aplicar el método de potencias para estimar λ_2 y un autovector asociado a éste, y repetir así el proceso sucesivamente hasta completar el cálculo de $\sigma(A)$.

Tras todas estas consideraciones, parece más que razonable el motivar el estudio de algún otro procedimiento que permita obtener de manera aproximada el espectro $\sigma(A)$ de manera conjunta. Abordamos pues en la siguiente sección el *algoritmo QR*, basado en la *descomposición QR* estudiada en el Capítulo 2 (y que no debemos confundir con ésta), el cual nos va a permitir nuestro objetivo.

4.1. Algoritmo QR

El gran avance en el problema del cálculo numérico de autovalores y autovectores de una matriz $A \in \mathscr{M}_n$ se produjo tras la invención del *algoritmo QR* en 1961, de manera independiente por J.G.F. Francis y V.N. Kublanovskaya. El comando eig de Matlab implementa una versión algo más sofisticada del mismo, que puede verse en [3]. Existen en la literatura otros procedimientos alternativos, como puede ser por ejemplo el *proceso de Arnoldi*, ejecutado en Matlab mediante la sentencia eigs, o el procedimiento *divide-and-conquer*, introducido en 1981 por Cuppen (véase [5, Lecture 26]). No obstante, por las limitaciones de este trabajo, nos centraremos en estudiar exclusivamente el algoritmo QR.

Para describir el algoritmo, consideremos la iteración inicial $A_0 = A$ y realicemos la descomposición QR de la misma (por ejemplo, mediante transformaciones de Householder o rotaciones de Givens): $A_0 = Q_0 R_0$. A continuación definimos $A_1 = R_0 Q_0$, y procederemos del mismo modo. Más concretamente, supuesta encontrada A_k, realizamos $A_k = Q_k R_k$ para a continuación definir $A_{k+1} = R_k Q_k$. Tenemos por tanto,

$$\begin{aligned}
A_0 &= A = Q_0 R_0 \\
A_1 &= R_0 Q_0 = Q_0^H A Q_0 = Q_1 R_1 \\
A_2 &= R_1 Q_1 = Q_1^H Q_0^H A Q_0 Q_1 = Q_2 R_2 \\
&\vdots \\
A_{k+1} &= R_k Q_k = Q_k^H \cdots Q_0^H A Q_0 \cdots Q_n = Q_{k+1} R_{k+1} \\
&\vdots
\end{aligned}$$

Dado que $A_k = (Q_0 \cdots Q_k)^H A (Q_0 \cdots Q_k)$, las matrices A_k son ortogonalmente semejantes a A, y por lo tanto, todas las matrices de la sucesión poseen el mismo espectro: $\sigma(A_k) = \sigma(A)$, para todo $k = 0, 1, \ldots$. Bajo ciertas suposiciones, este algoritmo converge a la descomposición de Schur de la matriz A (Capítulo 1, Teorema 1.2), $A = Q^H U Q$, siendo $Q = \lim_{k\to\infty} (Q_0 \cdots Q_k)^H$ una matriz unitaria, y U una matriz que

es triangular superior, para cualquier matriz A, y diagonal si la matriz A es simétrica. Tenemos pues en cualquier caso, que los autovalores de A serán los elementos que ocupan la diagonal principal de la matriz U.

El algoritmo QR puede ser relativamente costoso debido a la factorización QR que se debe llevar a cabo en cada iteración. Usualmente se suele proceder reduciendo la matriz A, antes de la primera iteración, en una forma más sencilla para la cual la factorización QR es mucho menos costosa.

Definición 4.1. *Una matriz A se dice que tiene estructura de Hessenberg si $a_{i,j} = 0$ para todo $i > j+1$. Es decir, es triangular superior excepto por la primera subdiagonal principal inferior, que contiene elementos no nulos:*

$$\begin{pmatrix} \times & \times & \times & \cdots & \times & \times \\ \times & \times & \times & \cdots & \times & \times \\ 0 & \times & \times & \cdots & \times & \times \\ 0 & 0 & \times & \cdots & \times & \times \\ \vdots & \vdots & \vdots & \ddots & \vdots & \vdots \\ 0 & 0 & 0 & \cdots & \times & \times \end{pmatrix}.$$

Si la matriz A es real y simétrica, ésta se puede reducir previamente a una matriz equivalente tridiagonal simétrica, y si no es simétrica, entonces podrá reducirse a una matriz de Hessenberg. La explicación de este procedimiento se sale fuera de nuestros objetivos, pero puede verse de manera detallada por ejemplo en [1, Págs. 615-618]. Recuérdase que en este caso tenemos, del Lema 1.4 que $\sigma(A) \subset \mathbb{R}$ y del Teorema 1.5 que podemos considerar una base de $\mathbb{R}^n$ formada por autovectores ortogonales. La razón de realizar este paso previo es que tanto si A tiene estructura tridiagonal como si tiene estructura de Hessenberg, la factorización QR admite una expresión simple, tanto si se emplean transformaciones de Householder como rotaciones de Givens, reduciendo el número de operaciones del algoritmo de $\mathcal{O}(n^3)$ a $\mathcal{O}(n^2)$ operaciones. El hecho de que los vectores columnas a los que se quieren hacer ceros tengan ya ceros en prácticamente todas sus entradas, hace que el coste computacional del proceso se reduzca considerablemente. El objetivo de realizar como paso previo la reducción a una matriz de Hessenberg A es que si $A = QR$ entonces puede comprobarse que tanto Q como RQ tendrán también estructura de Hessenberg, lo que asegura para todas las iteraciones del algoritmo el trabajar con este tipo de matrices estructuradas.

La demostración del siguiente resultado puede verse en [6, Capítulo 8].

Teorema 4.2. *Sea $A \in \mathcal{M}_n$ una matriz real, y supongamos que sus autovalores $\{\lambda_i\}_{i=1}^n$ verifican*

$$|\lambda_1| > |\lambda_2| > \cdots > |\lambda_n| > 0. \tag{4.1}$$

Entonces, las iteraciones R_m del algoritmo QR convergen a una matriz triangular superior D que contiene a los autovalores de A en las entradas de su diagonal principal. Si la matriz A es además simétrica, entonces la sucesión $\{A_m\}_{m\geq 0}$ converge a una

matriz diagonal, que contiene también a los autovalores de A en las entradas de su diagonal principal. En cuanto a la velocidad de convergencia, existe $c > 0$ para la cual

$$\| D - A_m \|_2 \leq c \cdot \max_i \left| \frac{\lambda_{i+1}}{\lambda_i} \right|.$$

Para matrices cuyo espectro no cumple la condición (4.1), las iteraciones podrían no converger a una matriz triangular. Si A es simétrica, la sucesión $\{A_m\}_{m\geq 0}$ convergerá a una matriz tridiagonal por bloques

$$A_m \underset{m\to\infty}{\longrightarrow} \begin{pmatrix} B_1 & 0 & \cdots & 0 \\ 0 & B_2 & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & B_r \end{pmatrix} \quad \text{donde} \quad B_i \in \mathcal{M}_1 \text{ ó } B_i \in \mathcal{M}_2, \quad i = 1, \ldots, r.$$

De este modo, los autovalores de A pueden ser fácilmente calculados a partir de D. Obsérvese que es muy fácil encontrar ejemplos donde $\{A_m\}_{m\geq 0}$ no siempre converge a una matriz diagonal, aún siendo A simétrica. Basta considerar el ejemplo elemental

$$A = \begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix}, \qquad \text{para la cual} \quad A_m = A, \quad R_m = I_n, \quad \forall m \geq 0.$$

Si la matriz A es real y no simétrica, la situación es algo más complicada, pero aceptable. Pueden verse los detalles en [6, Capítulo 8].

Con el objetivo de ganar velocidad de convergencia se puede establecer una variante del algoritmo QR, conocido como "algoritmo QR con desplazamiento"(shift). El proceso consiste en no realizar la factorización $Q_k R_k$ de A_k, sino de la matriz $A_k - \mu_k I_n$, donde μ_k es una estimación de alguno de los aultovalores de A: $A_k - \mu_k I_n = Q_k R_k$. Definimos así el proceso recursivo: $A_{k+1} = R_k Q_k + \mu_k I_n$. Con esta versión del algoritmo se consigue deflactar el proceso cuando un autovalor de A es encontrado, dado que la matriz A_k quedará estructurada en la correspondiente iteración en dos bloques,

$$A_k = \begin{pmatrix} A_1 & 0 \\ 0 & A_2 \end{pmatrix},$$

reduciendo así el problema a aplicar el algoritmo QR a las submatrices A_1 y A_2.

Podemos resumir el algoritmo QR de la siguiente forma:

1. Reducir la matriz A a estructura de Hessenberg $\mathcal{H}$, con un coste computacional de $\mathcal{O}(n^3)$ operaciones.
2. Aplicar la factorización QR a la matriz de Hessenberg resultante, con un coste computacional de $\mathcal{O}(n^2)$ operaciones, o a matriz tridiagonal simétrica resultante, si A es simétrica, con un coste computacional de $\mathcal{O}(n)$ operaciones.
3. Repetimos el Paso 2 a la matriz actualizada $\mathcal{H} = RQ$, que seguirá teniendo estructura de Hessenberg, o tridiagonal simétrica. El proceso se detendrá, de acuerdo con el Teorema 4.2, cuando todas las entradas de la diagonal inferior de R sean, en módulo, inferior a una tolerancia τ prefijada.

Para acelerar la convergencia del proceso:

1. Realizar la descomposición QR de $H-\mu I_n$. Aquí, μ debe ser una estimación de algún autovalor de A, que inicialmente podemos estimar aplicando criterios de localización (como por ejemplo, el Teorema de Gershgorin), y que a medida que avanza el proceso, podemos ir modificando en virtud de las expresiones de las matrices R que vayamos obteniendo. No obstante, existen en la literatura procesos para la deteminación de las cantidades μ. Para el caso particular de matrices tridiagonales simétricas, véase [1, Págs. 627-628].
2. Actualizar $H = RQ + \mu I$ y repetir el proceso con el mismo criterio de parada que en el caso anterior.

4.2. Dos ejemplos ilustrativos

Ejemplo 4.1 *Sea*

$$A_1 = \begin{pmatrix} 2 & 1 & 0 \\ 1 & 3 & 1 \\ 0 & 1 & 4 \end{pmatrix}.$$

Los autovalores son $\lambda_1 = 3+\sqrt{3} = 4.7321$, $\lambda_2 = 3.0$ *y* $\lambda_3 = 3-\sqrt{3} = 1.2679$. *En este caso se comprueba que las matrices iterativas* A_m *no convergen rápidamente a una matriz tridiagonal (obsérvse que los autovalores están muy próximos entre sí en módulo). Mostramos a continuación algunas iteraciones para observar este comportamiento cualitativamente, haciendo uso del código Matlab del Apéndice A.1.*

$$A_2 = \begin{pmatrix} 3.0000 & 1.0954 & 0 \\ 1.0954 & 3.0000 & -1.3416 \\ 0 & -1.3416 & 3.0000 \end{pmatrix}, \quad A_3 = \begin{pmatrix} 3.7059 & 0.9558 & 0 \\ 0.9558 & 3.5214 & 0.9738 \\ 0 & 0.9738 & 1.7727 \end{pmatrix},$$

$$A_7 = \begin{pmatrix} 4.6792 & 0.2979 & 0 \\ 0.2979 & 3.0524 & 0.0274 \\ 0 & 0.0274 & 1.2684 \end{pmatrix}, \quad A_8 = \begin{pmatrix} 4.7104 & 0.1924 & 0 \\ 0.1924 & 3.0216 & -0.0115 \\ 0 & -0.0115 & 1.2680 \end{pmatrix},$$

$$A_9 = \begin{pmatrix} 4.7233 & 0.1229 & 0 \\ 0.1229 & 3.0087 & 0.0048 \\ 0 & 0.0048 & 1.2680 \end{pmatrix}, \quad A_8 = \begin{pmatrix} 4.7285 & 0.0781 & 0 \\ 0.0781 & 3.0035 & -0.0020 \\ 0 & -0.0020 & 1.2680 \end{pmatrix}.$$

Los elementos en la posición $(1,2)$ *disminuyen geométricamente con un radio de aproximadamente* 0.64 *por iteración y los de la posición* $(2,3)$ *disminuyen con un radio de aproximadamente* 0.42 *por iteración. El valor en la posición* $(3,3)$ *de* A_{15} *será de* 1.2679, *alcanzando una precisión de cinco cifras significativas.*

Los cálculos del siguiente ejemplo se han realizado también haciendo uso del programa Matlab.

Ejemplo 4.2 *Consideremos la misma matriz A del ejemplo anterior y apliquemos el algoritmo QR con desplazamiento considerando* $\mu_m = a_{3,3}^{(m)}$. *Las cuatro matrices que se obtienen en la iteración son*

$$A_2 = \begin{pmatrix} 1.4000 & 0.4899 & 0 \\ 0.4899 & 3.2667 & 0.7454 \\ 0 & 0.7454 & 4.3333 \end{pmatrix}, \qquad A_3 = \begin{pmatrix} 1.2915 & 0.2017 & 0 \\ 0.2017 & 3.0202 & 0.2724 \\ 0 & 0.2724 & 4.6884 \end{pmatrix},$$

$$A_4 = \begin{pmatrix} 1.2737 & 0.0993 & 0 \\ 0.0993 & 2.9943 & 0.0072 \\ 0 & 0.0072 & 4.7320 \end{pmatrix}, \qquad A_5 = \begin{pmatrix} 1.2694 & 0.0498 & 0 \\ 0.0498 & 2.9986 & 0 \\ 0 & 0 & 4.7321 \end{pmatrix}.$$

Se aprecia claramente en esta modificación del algoritmo QR que se acelera notablemente la convergencia del proceso. Obsérvese que en la iteración 5 ya tenemos la matriz A_5 *separada en dos bloques disjuntos, por lo que el problema del cálculo de autovalores se reduce a problemas de dimensión menor. Ya en la quinta iteración hemos obtenido el autovalor* λ_1 *con cinco cifras significativas.*

A

Apéndice: Programación en Matlab

A.1. Factorización QR

Introducimos el siguiente código implementado en Matlab que computa la descomposición QR mediante transformaciones de Householder. Aunque Matlab ya tiene realmente una sentencia interna (qr) que calcula la descomposición QR de una matriz A, se ha implementado el código para la correcta comprensión del proceso y para mostrar por pantalla las sucesivas operaciones que éste conlleva:

```
function [Q,R]=qrtfg(A)
format long
n=length(A); Q=eye(n);
for k=1:n-1
    x=A(k:n,k)
    m=length(x);
    p=norm(x);
    q=x;
    q(1)=q(1)+sign(q(1))*p;
    u=q/norm(q);
    H2=eye(m)-2*u*u'
    if k==1
        H=H2;
    else
    H(1:k-1,1:k-1)=eye(k-1);
    H(1:k-1,k:n)=zeros(k-1,n-k+1);
    H(k:n,1:k-1)=zeros(n-k+1,k-1);
    H(k:n,k:n)=H2;
    end
    H
```

```
    Q=H*Q;
    A=H*A
end
R=A; Q=Q';
disp('Comprobación: ') Q*R
end
```

A.2. Algoritmo QR

El siguiente código computa el algoritmo QR sin desplazamiento. Los parámetros de entrada serán la matriz A, el número máximo nmax de iteraciones a realizar y una tolerancia tol. El proceso se detendrá, o bien si se alcanza el número máximo de iteraciones, o bien si antes de alcanzarlo, las entradas por debajo de la diagonal principal de las matrices que se van obteniendo son todas inferiores, en módulo, a la cantidad tol.

```
function qralgortfg(A,nmax,tol)
aux=0; m=length(A);
for i=1:nmax
    [Q,R]=qrtfg(A);
    A=R*Q
    for j=2:m
        for k=1:j-1
      if abs(A(j,k))>=tol
           aux=1;
      end
    end
    end
    if aux==0
        disp('Detenemos el proceso, hemos alcanzado la tolerancia
        deseada en la iteración '), i
        A
        return
    else
        aux=0
    end
end
disp('Número máximo de iteraciones máximo alcanzado') A
end
```

Bibliografía

[1] K.E. ATKINSON, *An Introduction to Numerical Analysis, John Wiley and Sons,* New York, 1989.

[2] J. E. GENTLE, *Matrix Algebra: Theory, computations, and applications in statistics*, Springer, 2007.

[3] G.H. GOLUB AND C.F. VAN LOAN, *Matrix Computations*, Third Edition, The John Hopkins University Press, 1992.

[4] E. ISAACSON AND H. KELLER, *Analysis of Numerical Methods,* Wiley, New York, 1966.

[5] LL. N. TREFETHEN AND D. BAU III, *Numerical Linear Algebra*, SIAM, 1997.

[6] J. WILKINSON, *The algebraic eigenvalue problem*, Oxford Univ. Press, Oxford, England, 1965.

Numerical Linear Algebra

Cristina González Marrero
Facultad de Ciencias · Sección de Matemáticas
Universidad de La Laguna
alu0100763065@ull.edu.es

Numerical Linear Algebra is the study of algorithms for performing linear algebra computations, most notably matrix operations, on computers. It is often a fundamental part of engineering and computational science problems, such as image and signal processing, telecommunication, computational finance, materials science simulations, structural biology, data mining, bioinformatics, fluid dynamics, and many other areas. Such software relies heavily on the development, analysis, and implementation of state-of-the-art algorithms for solving various numerical linear algebra problems, in large part because of the role of matrices in finite difference and finite element methods.
The aim of this Project is to delve in some on the contents that were already studied in the subject Numerical Methods I *of the Degree of Mathematics. It has been structures in four chapters and an Appendix.*

1. Preliminary results on Linear Algebra

We start by introducing some results about vectors and matrices that will be needed throughout the work. In particular, we consider norm spaces, orthogonal vectors and matrices, equivalent canonical forms, orthogonal similarity transformations, Schur factorization, spectral projections and the spectral decomposition of a given matrix.

2. Matrix transformations and factorizations

We introduce in Chapter 2 some new general matrix factorizations, like orthogonal and geometric transformations, Householder reflections and Givens rotations. Thus, we present two additional factorizations of special relevance in Numerical Linear Algebra: LU (Gaussian elimination) and QR factorizations. The construction A=QR is especially described in details: from Householder reflections, from Givens rotations and from the use of Gram-Schmidt transformations. Some illustrative examples are carried out.

3. Linear systems of equations

As an application of the previous results, we consider in Chapter 3 the problem to find numerically the solution of a linear system of equations. We introduce first the concept of conditioning, we describe direct and iterative (Jacobi and Gauss-Seidel) methods, and we place special emphasis on the method of successive over-relaxation.

4. Numerical calculus of eigenvalues and eigenvectors

We consider in the last chapter the problem of finding numerically the eigenvalues and eigenvectors of a given matrix by means of the QR algorithm. Some illustrative examples are carried out.

5. Appendix

This work contains an Appendix with two codes that have been implemented in MATLAB language: the first one let us to obtain the QR factorization of a given matrix, whereas the second one is an implementation of the QR algorithm.

References

[1] K.E. Atkinson, *An Introduction to Numerical Analysis, John Wiley and Sons*, New York, 1989.
[2] J. E. Gentle, *Matrix Algebra: Theory, computations, and applications in statistics*, Springer, 2007.
[3] G.H. Golub and C.F. Van Loan, *Matrix Computations*, Third Edition, The John Hopkins University Press, 1992.
[4] E. Isaacson and H. Keller, *Analysis of Numerical Methods*, Wiley, New York, 1966.
[5] Ll. N. Trefethen and D. Bau III, *Numerical Linear Algebra*, SIAM, 1997.
[6] J. Wilkinson, *The algebraic eigenvalue problem*, Oxford Univ. Press, Oxford, England, 1965.

TRABAJO FIN DE GRADO, Convocatoria de Septiembre, 2018