Análisis Numérico Matricial
Universidad de Murcia
curso 2016-2017

Unidad 2: Métodos Directos para Resolver Ecuaciones Lineales

Antonio José Pallarés Ruiz

Índice general

Capítulo 2

Métodos directos para resolver sistemas de ecuaciones

Interrogantes centrales del capítulo

- Analizar técnicas de resolución de sistemas de ecuaciones lineales.
- Aprender métodos de resolución :
  - Factorización LU y Choleski.
  - Método de Gauss.
  - Factorización QR, método de Householder.
- Analizar sistemas con matrices de coeficientes "especiales" (diagonal dominate, tridiagonales, definidas positivas, ...).
- Aproximar por mínimos cuadrados. Resolución aproximada de sistemas sobredeterminados.

Destrezas a adquirir en el capítulo

- Resolver sistemas de ecuaciones lineales utilizando los métodos directos del álgebra lineal.
- Implementar los métodos en el ordenador.
- Comparar su estabilidad y eficacia, y elegir el más adecuado a cada problema.
- Aplicar los programas construidos de forma efectiva en la búsqueda de la solución de sistemas lineales concretos.
- Describir algoritmos correspondientes a al método de aproximación por mínimos cuadrados.
- Plantear y resolver los sistemas de ecuaciones normales en problemas de aproximación por mínimos cuadrados, en particular los correspondientes a sistemas de ecuaciones lineales sobredeterminados. polinomios.

En esta unidad se estudian distintos **métodos directos** de resolución de sistemas de ecuaciones lineales

$$Ax = b.$$

Un método se dice directo cuando el número de operaciones a realizar es finito y hay que hacerlas todas para obtener la solución. En contraposición veremos en el siguiente tema **métodos iterativos** en donde se plantea la construcción de una sucesión que aproxime a x; en este caso no sabremos de antemano cuantas operaciones hay que hacer para conseguir la solución.

Si sabemos que A es regular (no singular), esto es, $det(A) \neq 0$, entonces existe inversa y por lo tanto el sistema lineal propuesto posee solución única

$$x = A^{-1}b.$$

Desde el punto de vista teórico parece que el problema ha terminado pero es el aspecto práctico el que no hace sino comenzar. Nos enfrentamos aquí a dos grandes dificultades

(I) **El coste computacional:** puede ser excesivo e imprácticable en tiempo y/o memoria.

(II) **Estabilidad del resultado:** influencia de los errores de representación en el cálculo que lo puede hacer muy inestable.

La complejidad numérica de cada uno de los métodos que se presentan se va a medir en función del número de operaciones que requiere y la estabilidad en los cálculos se va a analizar en función del tipo de operaciones que se realizan.

En los tres métodos propuestos el número de operaciones para un sistema $n \times n$ es del mismo orden, $O(n^3)$. La elección entre uno u otro dependerá de la naturaleza del sistema y del control que tengamos sobre la estabilidad de los cálculos.

En los tres métodos propuestos se simplifica el problema de resolver un sistema de ecuaciones, reduciéndolo a uno o dos sistemas (triangulares) más fácil de resolver. Usando estas reducciones, cualquiera de los métodos descritos permite calcular el determinante y la inversa de la matriz.

2.1. Sistemas fáciles de resolver

2.1.1. Sistemas diagonales

Si la matriz de coeficientes A es diagonal y no singular,

$$\begin{pmatrix} a_{11} & 0 & ... & 0 \\ 0 & a_{22} & ... & 0 \\ . & . & ... & . \\ 0 & 0 & ... & a_{\mathrm{nn}} \end{pmatrix} \begin{pmatrix} x_1 \\ x_2 \\ . \\ x_n \end{pmatrix} = \begin{pmatrix} b_1 \\ b_2 \\ . \\ b_n \end{pmatrix}$$

La solución se reduce a

$$\begin{pmatrix} x_1 \\ x_2 \\ . \\ x_n \end{pmatrix} = \begin{pmatrix} \frac{b_1}{a_{11}} \\ \frac{b_2}{a_{22}} \\ . \\ \frac{b_n}{a_{\mathrm{nn}}} \end{pmatrix}$$

y la complejidad del problema se puede medir contando las n operaciones (divisiones) necesarias.

2.1.2. Sistemas triangulares superiores. Método ascendente

$$\begin{pmatrix} a_{11} & a_{12} & ... & a_{1n} \\ 0 & a_{22} & ... & a_{2n} \\ . & . & ... & . \\ 0 & 0 & ... & a_{\mathrm{nn}} \end{pmatrix} \begin{pmatrix} x_1 \\ x_2 \\ . \\ x_n \end{pmatrix} = \begin{pmatrix} b_1 \\ b_2 \\ . \\ b_n \end{pmatrix}$$

Cuando la matriz de coeficientes es triangular superior (**U «upper»**) y no singular ($a_k k \neq 0, \forall k$), podemos resolver el sistema por el **método ascendente**, comenzando por encontrar la última coordenada en la ecuación de abajo y **ascendiendo** por el sistema, calculando una nueva coordenada con cada ecuación:

- $x_n = \frac{b_n}{a_{n,n}}$
- $x_k = (b_k - \sum_{j=k+1}^{n}(a_{k,j} * x_j))/a_{k,k}$ para $k = n-1, n-2, ..., 1$

Contando las operaciones nos encontramos con

- sumas: $1 + 2 + ... + n - 1 = n(n-1)/2$
- productos: $1 + 2 + ... + n - 1 = n(n-1)/2$
- divisiones: n

que hacen un máximo de n^2 operaciones necesarias para calcular la solución.

Algoritmo 2.1 **Método Ascendente.**

Datos de entrada: $A[n][n]$ (Matriz triangular sup. de coeficientes del sistema, sin ceros en la diagonal); $b[n]$ (vector término independiente.);
n (dimensión de A y b)
Variables: $x[n]$;// un vector donde escribir la solución.
Fujo del programa:

```
// Resolvemos el sistema por el método ascendente.
for(k=n;k>=1;k-){
  //x_k = (b_k - sum_{j=k+1}^{n}(A_{k,j} * x_j))/A_{k,k}
  x_k = b_k ;
    for(j=k+1;j<=n;j++){
      x_k = x_k - A_{k,j} * x_j ;
      }
  x_k = x_k/A_{k,k} ;
  }
```

Datos de salida: Solución x del sistema.

Ejercicio 2.1 Resuelve el sistema

$$\begin{cases} 3x + 2y + z = 14 \\ 6y + 3z = 15 \\ 2z = 2 \end{cases}$$

2.1.3. Sistemas triangulares inferiores. Método descendente

$$\begin{pmatrix} a_{11} & 0 & ... & 0 \\ a_{21} & a_{22} & ... & 0 \\ . & . & ... & . \\ a_{\mathrm{n1}} & a_{\mathrm{n2}} & ... & a_{\mathrm{nn}} \end{pmatrix} \begin{pmatrix} x_1 \\ x_2 \\ . \\ x_n \end{pmatrix} = \begin{pmatrix} b_1 \\ b_2 \\ . \\ b_n \end{pmatrix}$$

Si la matriz de coeficientes es triangular inferior (**L «lower»**) y no singular ($a_k k \neq 0, \forall k$), podemos resolver el sistema por el **método descendente**, comenzando por encontrar primera coordenada en la ecuación de arriba y **descendiendo** por el sistema, calculando una nueva coordenada con cada ecuación.

- $x_1 = \frac{b_1}{a_{n,n}}$
- $x_k = (b_k - \sum_{j=1}^{k-1}(a_{k,j} * x_j))/a_{k,k}$ para $k = 2, 3, ..., n$

Igual que antes, el número de operaciones necesarias para encontrar la solución es n^2.

Algoritmo 2.2 **Método Descendente.**

Datos de entrada: $A[n][n]$ (Matriz triangular inf. de coeficientes del sistema, sin ceros en la diagonal); $b[n]$ (vector término independiente.);
n (dimensión de A y b)
Variables: $x[n]$;// un vector donde escribir la solución.
Fujo del programa:

```
// Resolvemos el sistema por el método descendente.
for(k=1;k<=n;k++){
  //x_k = (b_k - sum_{j=k+1}^{n}(A_{k,j} * x_j))/A_{k,k}
  x_k = b_k ;
    for(j=1;j<k;j++){
      x_k = x_k - A_{k,j} * x_j ;
      }
  x_k = x_k/A_{k,k} ;
  }
```

Datos de salida: Solución x del sistema.

Ejercicio 2.2 Resuelve el sistema

$$\begin{cases} 2x & = 2 \\ 3x + 6y + & = 15 \\ x + 2y + 3z & = 14 \end{cases}$$

Ejercicio 2.3

(I) Comprobar que el producto de matrices triangular superiores (respectívamente inferiores) también es una matriz triangular superior (resp. inferior).

(II) Comprobar que el producto de matrices triangular superiores (respectívamente inferiores) con unos en la diagonal también es una matriz triangular superior (resp. inferior) con unos en la diagonal.

(III) Comprobar que la inversa de una matriz triangular superior (respectívamente inferior) también es una matriz triangular superior (resp. inferior).

(IV) Comprobar que la inversa de una matriz triangular superior (respectívamente inferior) con unos en la diagonal también es una matriz triangular superior (resp. inferior) con unos en la diagonal.

2.2. Factorización LU

Supongamos ahora que la matriz $A = (a_{\text{ij}})_{i,j=1,...,n}$ admite una factorización como el producto de una matriz triangular inferior L por una matriz triangular superior U, esto es $A = LU$

$$\begin{pmatrix} a_{11} & a_{12} & ... & a_{1n} \\ a_{21} & a_{22} & ... & a_{2n} \\ . & . & ... & . \\ a_{n1} & a_{n2} & ... & a_{\text{nn}} \end{pmatrix} = \begin{pmatrix} l_{11} & 0 & ... & 0 \\ l_{21} & l_{22} & ... & 0 \\ . & . & ... & . \\ l_{n1} & l_{n2} & ... & l_{\text{nn}} \end{pmatrix} . \begin{pmatrix} u_{11} & u_{12} & ... & u_{1n} \\ 0 & u_{22} & ... & u_{2n} \\ . & . & ... & . \\ 0 & 0 & ... & u_{\text{nn}} \end{pmatrix} \tag{2.1}$$

Para resolver el sistema lineal de ecuaciones:

$$A.x = b \iff LUx = b$$

resolvemos consecutivamente los sistemas :

$$Ly = b \quad \text{y} \quad Ux{=}y$$

con los métodos descendente y ascendente descritos en los apartados de arriba.

Cuando se pueden hacer este tipo de factorizaciones (2.1), se dice que A tiene una factorización LU.

Observación 2.2.1 Si A tiene una factorización LU, $A = LU$, los menores principales de A se factorizan como producto de los menores principales de L y de U:

$$A_k = \begin{pmatrix} a_{11} & a_{12} & ... & a_{1k} \\ a_{21} & a_{22} & ... & a_{2k} \\ . & . & ... & . \\ a_{k1} & a_{k2} & ... & a_{\text{kk}} \end{pmatrix} = \begin{pmatrix} l_{11} & 0 & ... & 0 \\ l_{21} & l_{22} & ... & 0 \\ . & . & ... & . \\ l_{k1} & l_{k2} & ... & l_{\text{kk}} \end{pmatrix} . \begin{pmatrix} u_{11} & u_{12} & ... & u_{1k} \\ 0 & u_{22} & ... & u_{2k} \\ . & . & ... & . \\ 0 & 0 & ... & u_{\text{kk}} \end{pmatrix}$$

Proposición 2.2.2 (Condición necesaria para existencia de LU) *Si $A \in \mathcal{M}_n(\mathbb{K})$ es una matriz no singular ($\det(A) \neq 0$) y existe una factorización $A = LU$ como en (2.1), entonces se cumple que*

$$\det(A) = \det(L)\det(U) = \prod_{i=1}^{n} l_{ii} \prod_{i=1}^{n} u_{ii} \neq 0,$$

o lo que es equivalente:

$$l_{ii} \neq 0 \ y \ u_{ii} \neq 0 \quad para \quad i = 1, 2, ..., n,$$

y en consecuencia todos los menores principales de A tienen determinante no nulo:

$$\det(A_k) == \prod_{i=1}^{k} l_{ii} \prod_{i=1}^{k} u_{ii} \neq 0.$$

La factorización $A = LU$ si existe, no tiene porqué ser única, aunque tal y como acabamos de observar, si A es no singular, los productos de los elementos de las diagonales de L y de U están determinados de forma única por los determinantes de los menores principales:

$$l_{ii}u_{ii} = \det(A_i)/\det(A_{i-1}), \tag{2.2}$$

conveniando que $\det(A_0) = 1$.

En la sección siguiente, vamos a describir un algorítmo de factorización, que nos vendrá a demostrar que el que los menores principales sean no singulares, también es una condición suficiente para la existencia de factorizaciones LU.

2.2.1. Algoritmos de factorización

Supongamos que A es una matriz cuadrada con todos sus menores principales A_k no singulares.

Para deducir la factorización LU comenzamos múltiplicando las matrices en (2.1), obteniendo las ecuaciones

$$a_{i,j} = \sum_{s=1}^{n} l_{i,s}u_{s,j} = \sum_{s=1}^{\text{mín}(i,j)} l_{i,s}u_{s,j}$$

Con estas ecuaciones, por etapas, podemos ir determinando las filas de U y las columnas de L. Supongamos que tenemos determinados las (k-1) primeras filas de U y las (k-1) primeras columnas de L. La ecuación correspondiente al término $a_{k,k}$:

$$a_{k,k} = \sum_{s=1}^{k-1} l_{k,s}u_{s,k} + l_{k,k}u_{k,k} \qquad \text{y} \qquad p_k = l_{k,k}u_{k,k} = a_{k,k} - \sum_{s=1}^{k-1} l_{k,s}u_{s,k} \tag{2.3}$$

Sabemos por 2.2 que el producto $p_k = l_{k,k}u_{k,k}$ está definido de forma única y es no nulo. Ahora podemos seguir distintos criterios para determinar los valores de $l_{k,k}$ y de $u_{k,k}$. Señalaremos los dos más usados (en la sección siguiente estudiaremos más cada uno de los casos, de momento podemos verlos cómo posibilidades de obtener distintas factorizaciones):

- Criterio de Dootlittle .- $l_{k,k} = 1 \quad u_{k,k} = p_k$
- Criterio de Crout .- $u_{k,k} = 1 \quad l_{k,k} = p_k$

Una vez determinados los coeficientes $l_{k,k}$ y $u_{k,k}$, variando $j > k$ e $i > k$ completamos la fila k de U y la columna k de L:

$$a_{k,j} = \sum_{s=1}^{k-1} l_{k,s}u_{s,j} + l_{k,k}u_{k,j} \qquad \text{y} \qquad u_{k,j} = \left(a_{k,j} - \sum_{s=1}^{k-1} l_{k,s}u_{s,j}\right) l_{k,k}^{-1} \tag{2.4}$$

$$a_{i,k} = \sum_{s=1}^{k-1} l_{i,s}u_{s,k} + l_{i,k}u_{k,k} \qquad \text{y} \qquad l_{i,k} = \left(a_{i,k} - \sum_{s=1}^{k-1} l_{i,s}u_{s,k}\right) u_{k,k}^{-1} \tag{2.5}$$

Como $p_k = l_{k,k}u_{k,k} \neq 0$, una vez elegidos los elementos de las diagonales $l_{k,k}$ y $u_{k,k}$, el resto de los elementos de la fila k ($u_{k,j}$) de U y de la columna k ($l_{i,k}$) de L están definidos de forma única.

Esta construcción prueba que el que todos los menores principales de A tengan determinante no nulo (que implica el que $p_k \neq 0$ para $k = 1, ..., n$) también es condición suficiente para que exista la factorizacion $A = LU$.

Teorema 2.2.3 *Si $A \in \mathcal{M}_n(\mathbb{K})$ es una matriz no singular, las siguientes afirmaciones son equivalentes:*

(I) existe una factorización $A = LU$ como en (2.1);

(II) todos los menores principales de A tienen determinante no nulo: $\det(A_k) \neq 0$.

El algoritmo 2.3 describe el método de Dootlittle de factorización LU.

2.2.2. Complejidad de las factorizaciones LU

Para medir el número máximo de operaciones necesario para obtener factorizaciones LU observamos que

- En (2.3) se necesitan $(k-1)$ productos y $(k-1)$ sumas, e.d. $2(k-1)$ operaciones para cada valor de $k = 1, 2, ..., n$.
- En cada uno de los $(n-k)$ procesos (2.4) y (2.5) se necesitan $(k-1)$ productos, $(k-1)$ sumas y una división para cada valor de $k = 1, 2, ..., n$ y cada $j = (k+1), ..., n$. En total $4(k-1)+1$ operaciones en el caso de Dootlittle (sólo hay división en (2.5)).

En total se necesitan[1]

$$\begin{aligned}\sum_{k=1}^{n}(2(k-1)+(n-k)(4(k-1)+1)) &= 2\sum_{k=1}^{n-1}k + 4n\sum_{k=1}^{n-1}k - 4\sum_{k=1}^{n}(k^2-k) + n - \sum_{k=1}^{n}k \\ &= n(n-1) + 2n^2(n-1) - \frac{2}{3}n(n+1)(2n+1) + 3\frac{n(n+1)}{2} + n \\ &= \frac{2}{3}n^3 - \frac{3}{2}n^2 + \frac{5}{6}n\end{aligned}$$

operaciones.

Una vez obtenida la factorización para resolver sistemas $Ly = b$ y $Ux = y$ se necesitan $2n^2$ operaciones adicionales.

Es ilustrativo comparar la resolución de sistemas con factorización LU, con el uso de la regla de Cramer que genera un coste aproximado de $n!$. Para valores moderados de n ya la diferencia de $n!$ con $\frac{2}{3}n^3$ es abismal.

Ejercicio 2.4 Demuestra que la factorización $A = LU$ conserva la estructura de las matrices banda, es decir, si $a_{ij} = 0$ para $|i-j| > p$, entonces $l_{ij} = 0$ si $i - j > p$ y $u_{ij} = 0$ si $j - i > p$.

[1] Recuerda que $\sum_{k=1}^{n-1} k = \frac{n(n-1)}{2}$ y $\sum_{k=1}^{n} k^2 = \frac{n(n+1)(2n+1)}{6}$.

Ejercicio 2.5 Modifica el algoritmo de factorización LU para calcular el determinante de una matriz.

Algoritmo 2.3 Método de factorización LU (Dootlittle)

Datos de entrada: $A[n][n]$ (Matriz de coeficientes del sistema.);
n (dimensión de A)
Variables: $L[n][n]$; $U[n][n]$// matrices para escribir las matrices triangulares superiores e inferiores.
aux;// una variable auxiliar para sumatorios y productos escalares.
Fujo del programa:

```
aux = 0. ;
for(k=0;k<n;k++){
  aux = A[k][k];
  for(s=0;s<k;s++){ // de Fila k de L por Columna k de U.
    aux = aux − L[k][s]U[s][k]; }
  if(aux==0){
    Parada: no hay factorización LU;}
  L[k][k] = 1.;
  U[k][k] = aux;
  for(j=k+1;j<n;j++){// de Fila k de L por Columna j de U.
    aux = A[k][j];
    for(s=0;s<k;s++){ aux = aux − L[k][s] ∗ U[s][j];}
    U[k][j] = aux; }
  for(i=k+1;i<n;i++){// de Fila i de L por Columna k de U.
    aux = A[i][k];
    for(s=0;s<k;s++){ aux = aux − L[i][s] ∗ U[s][k];}
    L[i][k] = aux/U[k][k];}
  }
```

Datos de salida:L y U(Factorización LU)o mensaje de error

Ejercicio 2.6 Considera el sistema lineal

$$\begin{pmatrix} 2 & -1 & & \\ -1 & 2 & -1 & \\ & -1 & 2 & -1 \\ & & -1 & 2 \end{pmatrix} \begin{pmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \end{pmatrix} \begin{pmatrix} 19 \\ 19 \\ -3 \\ -12 \end{pmatrix}$$

¿Admite una factorización LU? Si la respuesta es afirmativa, resuelve el sistema calculando y utilizando la factorización LU.

2.2.3. Factorizaciones LDU

A partir de la factorización de Dootlittle: $A = LU$ con $l_{ii} = 1$, se tiene que $det(A) = u_{11}u_{22}....u_{nn}$ donde u_{ii} son los elementos en la diagonal de U. Evidentemente, si $det(A) \neq 0$

entonces $u_{ii} \neq 0$ para cualquier i. En este caso, si tomamos $D = diag\{u_{11}, u_{22}, ..., u_{nn}\}$ nos encontramos con la descomposición $A = LD\tilde{U}$ dada por

$$\begin{pmatrix} a_{11} & a_{12} & ... & a_{1n} \\ a_{21} & a_{22} & ... & a_{2n} \\ . & . & \ddots & . \\ a_{n1} & a_{n2} & ... & a_{\mathrm{nn}} \end{pmatrix} = \begin{pmatrix} 1 & 0 & ... & 0 \\ l_{21} & 1 & ... & 0 \\ . & . & \ddots & . \\ l_{n1} & l_{n2} & ... & 1 \end{pmatrix} \cdot \begin{pmatrix} u_{11} & 0 & ... & 0 \\ 0 & u_{22} & ... & 0 \\ 0 & 0 & \ddots & . \\ 0 & 0 & ... & u_{\mathrm{nn}} \end{pmatrix} \begin{pmatrix} 1 & \tilde{u}_{12} & ... & \tilde{u}_{1n} \\ 0 & 1 & ... & \tilde{u}_{2n} \\ . & . & \ddots & . \\ 0 & 0 & ... & 1 \end{pmatrix}$$

donde $\tilde{u}_{ij} = u_{ij}/u_{ii}$. Denotando por $M = \tilde{U}^t$ esta factorización toma ahora la forma $A = LDM^t$ en donde tanto L como M son triangulares inferiores con unos en la diagonal.

Observad que esta factorización es única, pues si $A = L_1 D M_1^t = L_2 D_2 M_2^t$, con L_1, L_2, M_1 y M_2 triangulares inferiores con unos en la diagonal, se tendrá que

$$L_2^{-1} L_1 D_1 = D_2 M_2^t (M_1^t)^{-1},$$

donde la matriz de la izquierda es triangular inferior y la de la derecha es triangular superior, por lo tanto $L_2^{-1} L_1 D_1$ también $L_2^{-1} L_1$ son matrices diagonales, esta última con unos en la diagonal sólo puede ser la identidad y $L_1 = L_2$. Análogamente se concluye que $M_1 = M_2$ y por último que $D_1 = D_2$.

Teorema 2.2.4 *Si A es una matriz cuadrada que admite una factorización LU, entonces existen dos matrices triangulares inferiores con unos en la diagonal, L y M, y una matriz diagonal D sin ceros en la diagonal de forma que*

$$A = LDM^t.$$

Además esta factorización es única.

Si agrupamos L y D, obtenemos la factorización de Crout, esto es

$$\begin{pmatrix} a_{11} & a_{12} & ... & a_{1n} \\ a_{21} & a_{22} & ... & a_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ a_{n1} & a_{n2} & ... & a_{\mathrm{nn}} \end{pmatrix} = \begin{pmatrix} u_{11} & 0 & ... & 0 \\ \tilde{l}_{21} & u_{22} & ... & 0 \\ \vdots & \vdots & \ddots & \vdots \\ \tilde{l}_{n1} & \tilde{l}_{n2} & ... & u_{\mathrm{nn}} \end{pmatrix} \cdot \begin{pmatrix} 1 & \tilde{u}_{12} & ... & \tilde{u}_{1n} \\ 0 & 1 & ... & \tilde{u}_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & ... & 1 \end{pmatrix}$$

Volviendo a la factorización $A = LDM^t$ en donde tanto L como M son triangulares inferiores con unos en la diagonal, en el caso en el que A es simétrica resulta que además $L = M$

Teorema 2.2.5 *Si $A^t = A$ con $det(A) \neq 0$ y existe la factorización LU entonces existe una única factorización de la forma $A = LDL^t$ donde L es una matriz triangular con unos en la diagonal y D es una matriz diagonal.*

DEMOSTRACIÓN:
Ideas que intervienen

- *La demostración se basa en que tenemos la factorización única $A = LDM^t$ con L y M triangulares inferiores con unos en la diagonal.*
- *Como A es simétrica, $A = A^t = MDL^t$. La unicidad de la factorización LDM prueba que $L = M$.*

□

2.2.4. Transformaciones de Gauss sin permutar filas

La factorización LU se puede considerar como una descripción algebraica del método de Gauss sin permutaciones de filas (método de eliminación) que será útil para futuras aplicaciones.

¿Cómo podemos convertir en ceros algunas componentes de un vector al multiplicarlo por una matriz?

Recordemos que el producto de una matriz por un vector genera un nuevo vector mediante rotaciones y/o cambios de tamaño. Por lo tanto, hacer cero en componentes de un vector equivale a moverlo de tal forma que no tenga componente en determinadas direcciones.

Por ejemplo, dado el vector columna $a = (a_1, a_2)^t$, con $a_1 \neq 0$, si usamos $\tau = a_2/a_1$ resulta que

$$\begin{pmatrix} 1 & 0 \\ -\tau & 1 \end{pmatrix} \begin{pmatrix} a_1 \\ a_2 \end{pmatrix} = \begin{pmatrix} a_1 \\ 0 \end{pmatrix}$$

De forma general, si $a = (a_1, a_2, ..., a_n)^t$ y $a_k \neq 0$ entonces podemos anular a_j para $j = k+1, ..., n$ siguiendo esta misma idea y usando la matriz triangular inferior

$$M_k = \begin{pmatrix} 1 & 0 & ... & 0 & ... & 0 \\ 0 & 1 & ... & 0 & ... & 0 \\ 0 & 0 & . & . & ... & . \\ 0 & 0 & & 1 & ... & 0 \\ 0 & 0 & & -\tau_{k+1} & ... & 0 \\ \vdots & \vdots & . & \vdots & ... & . \\ 0 & 0 & ... & -\tau_n & ... & 1 \end{pmatrix}$$

para

$$\tau_j = a_j/a_k, \quad j = k+1, ..., n,$$

tenemos que

$$\begin{pmatrix} 1 & 0 & ... & 0 & ... & 0 \\ 0 & 1 & ... & 0 & ... & 0 \\ 0 & 0 & . & . & ... & . \\ 0 & 0 & & 1 & ... & 0 \\ 0 & 0 & & -\tau_{k+1} & ... & 0 \\ \vdots & \vdots & . & \vdots & ... & . \\ 0 & 0 & ... & -\tau_n & ... & 1 \end{pmatrix} \begin{pmatrix} a_1 \\ a_2 \\ \vdots \\ a_k \\ a_{k+1} \\ \vdots \\ a_n \end{pmatrix} = \begin{pmatrix} a_1 \\ a_2 \\ \vdots \\ a_k \\ 0 \\ \vdots \\ 0 \end{pmatrix}$$

Podemos expresar la transformación de Gauss M_k en la forma

$$M_k = I_n - \tau(k) \cdot e_k^t$$

usando la matriz identidad Id_n y los vectores columna $\tau(k) = (0, 0, .., 0, \tau_{k+1}, ..., \tau_n)^t$ y e_k el vector de la base canónica $e_k = (0, 0, ..., 0, 1, 0, ...0)^t$ donde 1 está en la coordenada k-ésima. Simplemente multiplicando, se comprueba que $M_k^{-1} = I_n + \tau(k) \cdot e_k^t$ (triangular inferior):

$$(I_n - \tau(k) \cdot e_k^t)(I_n + \tau(k) \cdot e_k^t) = I_n.$$

El uso de estas transformaciones de Gauss para obtener ceros debajo de la diagonal es ahora evidente. Dada la matriz A si $a_{11} \neq 0$ tomamos $\tau^1 = (0, \tau_2^1, ..., \tau_n^1)^t$ con $\tau_j^1 = a_{j1}/a_{11}$, $j = 2, ..., n$

y usamos $M_1 = I_n + \tau^1 \cdot e_1^t$. Si denotamos por $a_{\cdot,j}$ la columna j de A y por $a_{i,\cdot}$ la fila i de A entonces resulta que se puede expresar el producto en términos de columnas y

$$M_1 A = [M_1 a_{\cdot,1}, M_1 a_{\cdot,2}, ..., M_1 a_{\cdot,n}]$$

esto es

$$M_1 A = A^{(2)} = \begin{pmatrix} a_{11} & a_{12} & ... & a_{1n} \\ 0 & a_{22}^{(2)} & ... & a_{2n}^{(2)} \\ 0 & a_{32}^{(2)} & ... & a_{3n}^{(2)} \\ \vdots & \vdots & \vdots & \vdots \\ \vdots & \vdots & \vdots & \vdots \\ 0 & a_{n2}^{(2)} & a_{n3}^{(2)} & a_{nn}^{(2)} \end{pmatrix}$$

Suponiendo que $a_{22}^{(2)} \neq 0$, si repetimos este proceso de anular debajo de $a_{22}^{(2)}$ podremos hacerlo de manera similar multiplicando por una matriz triangular inferior M_2, consiguiendo una nueva matriz $A^{(3)} = M_2 A^{(2)}$ con ceros bajo la diagonal en las dos primeras columnas.

Cuando se puede reiterar este procedimiento haciendo $A^{(j)} = M_{j-1} A^{(j-1)}$ obtenemos el método de eliminación de Gauss sin permutaciones de filas. Si podemos reiterar será porque cada elemento que vamos encontrando en la diagonal es no nulo, $a_{jj}^{(j)} \neq 0$, para $j = 2, 3, ..., n-1$. Así, construiremos matrices: $M_1, M_2, .., M_{n-1}$, de forma que

$$M_{n-1} M_{n-2} .... M_2 M_1 A = U$$

donde U es triangular superior.

Además, tenemos que

$$L = M_1^{-1} M_2^{-1} ... M_{n-2}^{-1} M_{n-1}^{-1}$$

es triangular inferior con unos en la diagonal, de acuerdo a los ejercicios vistos antes, y que $A = LU$. Por lo tanto tenemos una factorización LU de A con L triangular inferior y con unos en la diagonal (Factorización de Dootlittle):

$$\begin{pmatrix} a_{11} & a_{12} & ... & a_{1n} \\ a_{21} & a_{22} & ... & a_{2n} \\ . & . & ... & . \\ a_{n1} & a_{n2} & ... & a_{\mathrm{nn}} \end{pmatrix} = \begin{pmatrix} 1 & 0 & ... & 0 \\ l_{21} & 1 & ... & 0 \\ . & . & ... & . \\ l_{n1} & l_{n2} & ... & 1 \end{pmatrix} \cdot \begin{pmatrix} u_{11} & u_{12} & ... & u_{1n} \\ 0 & u_{22} & ... & u_{2n} \\ . & . & ... & . \\ 0 & 0 & ... & u_{\mathrm{nn}} \end{pmatrix}$$

Tenemos más información: se ve con facilidad que con la notación anterior:

$$L = \begin{pmatrix} 1 & 0 & ... & ... & ... & 0 \\ \tau_2^1 & 1 & ... & ... & ... & 0 \\ \tau_3^1 & \tau_3^2 & 1 & ... & ... & \vdots \\ \tau_4^1 & \tau_4^2 & \tau_4^3 & 1 & ... & \vdots \\ \vdots & \vdots & \vdots & \ddots & \ddots & \vdots \\ \tau_n^1 & \tau_n^2 & \tau_n^3 & ... & \tau_n^{n-1} & 1 \end{pmatrix}$$

Ejercicio 2.7 Comprueba que si A es no singular y existe la factorización de Dootlittle $A = LU$, entonces A se puede triangular por el método de eliminación de Gauss sin permutación de filas.

2.3. Sistemas con matrices especiales

2.3.1. Matrices con diagonal estrictamente dominante

Definición 2.3.1 *Se dice que la* ***diagonal*** *de una matriz cuadrada de dimensión n* $A = (a_{ij})$ *es* ***estrictamente dominante*** *cuando*

$$|a_{ii}| > \sum_{j=1;j\neq i}^{n} |a_{ij}|$$

para toda fila $i = 1, ...n$*. Esto es, en cada fila el elemento de la diagonal domina sobre toda la fila.*

Ejemplo 2.3.2 Si consideramos las matrices

$$A = \begin{pmatrix} 7 & 2 & 0 \\ 3 & 5 & 1 \\ 0 & 5 & 6 \end{pmatrix} \quad \text{y} \quad B = \begin{pmatrix} 5 & 3 & -3 \\ 3 & -4 & 0 \\ 3 & 0 & 4 \end{pmatrix}$$

La diagonal de la matriz A es estrictamente dominante, A no simétrica y la diagonal de A^t no es estrictamente dominante. La matriz B es simétrica pero no tiene diagonal estrictamente dominante (tampoco lo es $B^t = B$.

No son raros los sistemas lineales con diagonal estrictamente dominante que aparecen en muchos modelos de ecuaciones en diferencias (elementos finitos) al discretizar ecuaciones en derivadas parciales y en métodos numéricos como en el caso de los problemas de interpolación con "splines" estudiados el pasado cuatrimestre. Las matrices con diagonal estrictamente dominante son no singulares y tienen buenas propiedades de estabilidad con relación al método de Gauss.

Teorema 2.3.3 *Toda matriz A con diagonal estrictamente dominante es no singular. Además, A se puede triangular con el método de eliminación de Gauss sin permutaciones de filas, equivalentemente, A admite una factorización* $A = LU$*, donde L es triangular inferior con unos en la diagonal y U es triangular superior con diagonal estrictamente dominante.*

Demostración:
Ideas que intervienen

- *Se puede razonar por contradicción: Si A fuese singular existiría* $x = (x_1, .., x_n)^t$ *tal que* $Ax = 0$.

  Tomamos k tal que $|x_k| = \|x\|_\infty = \max\{|x_1|, ...|x_n|\}$

  Como $\sum_j a_{ij}x_j = 0$ *para todo i, en particular para* $i = k$ *se tiene*

  $$a_{kk}x_k = -\sum_{j=1;j\neq k}^{n} a_{kj}x_j.$$

  La desigualdad triangular nos dice entonces que

  $$|a_{kk}||x_k| \leq \sum_{j=1;j\neq k}^{n} |a_{kj}||x_j|,$$

Lo que contradice la hipótesis de que A es estrictamente diagonal dominante porque

$$|a_{kk}| \leq \sum_{j=1;j\neq k}^{n} |a_{kj}| \frac{|x_j|}{|x_k|} \leq \sum_{j=1;j\neq k}^{n} |a_{kj}|.$$

- *Si A es tiene diagonal estrictamente dominante, todos los elementos de su diagonal son no nulos. En particular $a_{11} \neq 0$ y se puede utilizar la transformación de Gauss*

$$M_1 = \begin{pmatrix} 1 & 0 & 0 & ... & 0 \\ -\tau_2 & 1 & 0 & ... & 0 \\ -\tau_3 & 0 & 1 & ... & \vdots \\ \vdots & \vdots & \vdots & \ddots & . \\ -\tau_n & 0 & 0 & ... & 1 \end{pmatrix}$$

con $\tau_k = \frac{a_{k1}}{a_{11}}$. La matriz $B = M_1 A = (b_{ij})$ tiene la misma primera fila que A, ceros bajo la diagonal en la primera columna y el resto de elementos vienen dados por

$$b_{ij} = a_{ij} - \frac{a_{i1}}{a_{11}} a_{1j}$$

La desigualdad triangular nos permite probar que

$$\sum_{j=2;j\neq i}^{n} |b_{ij}| < |b_{ii}|.$$

En otras palabras, la matriz $B = M_1 A$ que proporciona la primera transformación de Gauss también tiene diagonal estrictamente dominante.

Repitiendo el proceso se triangula la matriz con transformaciones de Gauss sin permutaciones de filas y la matriz triangular que se obtiene tiene diagonal estrictamente dominante.

□

Si la matriz A tiene diagonal estrictamente dominante (no singular) y no tiene ninguna fila "casi" nula, los cálculos del método de Gauss sin hacer cambios de filas ni columnas serán estables ya que los pivotes no resultan demasiado pequeños.

Los métodos iterativos de la siguiente lección para sistemas de ecuaciones con este tipo de matrices son convergentes.

2.3.2. Matrices definidas positivas.

Definición 2.3.4 *Se dice que una matriz de dimensión n $A = (a_{ij})$ es **definida positiva** (PD por la expresión "positive definite" en inglés) cuando*

$$x^t A x = \sum_{i,j=1}^{n} a_{ij} x_i x_j > 0$$

para todo vector columna $x \in \mathbb{R}^n \setminus \{0\}$.

Es fácil observar que si A es PD entonces A no es singular, porque si $Ax = 0$ entonces $x^t Ax = 0$ y la única posibilidad para esto es que $x = 0$.

En el caso de matrices PD de dimensión 2, 2×2,

$$A = \begin{pmatrix} a_{11} & a_{12} \\ a_{12} & a_{22} \end{pmatrix}$$

se puede ver que

$$\begin{aligned} x = (1,0)^t &\Rightarrow x^t Ax = a_{11} > 0 \\ x = (0,1)^t &\Rightarrow x^t Ax = a_{22} > 0 \\ x = (1,1)^t &\Rightarrow x^t Ax = a_{11} + 2a_{12} + a_{22} > 0 \\ x = (1,-1)^t &\Rightarrow x^t Ax = a_{11} - 2a_{12} + a_{22} > 0 \end{aligned}$$

entonces tenemos que

$$|a_{12}| \leq (a_{11} + a_{22})/2.$$

Aunque no es una dominancia estricta, si que podemos ver que los elementos de la diagonal son positivos y que es en la diagonal donde se concentra el mayor valor.

A continuación vamos a ir observando algunas propiedades de las matrices PD que nos van a llevar a la existencia de factorizaciones LU.

Teorema 2.3.5 *Si $A \in \mathbb{K}^{n\times n}$ es PD y $X \in \mathbb{K}^{n\times k}$ tiene rango k entonces $B = X^t AX \in \mathbb{K}^{k\times k}$ también es PD.*

Demostración: Simplemente usar $y = Xz \neq 0$ para cualquier $z \in \mathbb{K}^k \setminus \{0\}$. □

Corolario 2.3.6 *Si $A \in \mathbb{K}^{n\times n}$ es PD, A es no singular, todos sus menores principales tienen determinante no nulo y todos los elementos de la diagonal son positivos.*

Demostración: Usando como X la matriz de las primeras k columnas de la matriz identidad en el teorema anterior, se tiene que el menor principal $A_k = X^t AX$ es PD y en consecuencia no singular (con determinante no nulo).

Si tomamos como X el vector columna k-ésimo de la matriz identidad en el teorema, tenemos que $(a_{kk}) = X^t AX$ es positivo. □

Corolario 2.3.7 *Si $A \in \mathbb{K}^{n\times n}$ es PD entonces existe la factorización $A = LDM^t$ con L y M triangulares inferiores con unos en la diagonal y $D = diag\{d_1, d_2, ..., d_n\}$ es diagonal con $d_i > 0$ para todo $i = 1, 2, ..., n$.*

Demostración: Del Corolario 2.3.6 sabemos que existe la factorización LU de A y por lo tanto una única factorización de la forma $A = LDM^t$ con L y M triangulares inferiores con unos en la diagonal ty D una matriz diagonal. El Teorema 2.3.5 con $X = L^{-1t}$ asegura que $L^{-1}AL^{-1t} = DM^t L^{-t}$ es PD. Pero al ser $M^t L^{-t}$ triangular superior con unos en la diagonal resulta que $DM^t L^{-t}$ y D poseen la misma diagonal, sus términos $d_i > 0$ son positivos por ser PD. □

Observación 2.3.8 La existencia de la factorización $A = LDM^t$ no quiere decir que su computación sea recomendable. Por ejemplo, si

$$A = \begin{pmatrix} \varepsilon & m \\ -m & \varepsilon \end{pmatrix}$$

es PD, nos encontramos con la descomposición

$$\begin{pmatrix} \varepsilon & m \\ -m & \varepsilon \end{pmatrix} = \begin{pmatrix} 1 & 0 \\ -m/\varepsilon & 1 \end{pmatrix} \begin{pmatrix} \varepsilon & 0 \\ 0 & \varepsilon + m^2/\varepsilon \end{pmatrix} \begin{pmatrix} 1 & m/\varepsilon \\ 0 & 1 \end{pmatrix}$$

pero si $m/\varepsilon >> 1$ resulta más recomendable pivotar intercambiando las filas.

2.3.3. Matrices simétricas definidas positivas (SPD).

Existen varias situaciones en las que nos podemos encontrar con matrices PD aunque lo normal es que además sean simétricas. Por ejemplo al considerar matrices simétricas (hermitianas en el caso complejo) de la forma A^*A

Definición 2.3.9 *Se dice que una matriz A es simétrica simétrica definida positiva cuando $A^t = A$ y es PD.*

Proposición 2.3.10 *Si A es una matriz simétrica definida positiva entonces:*

(I) A es no singular

(II) $a_{ii} > 0$ para cada $i = 1, ..., n$.

(III) $\max\{|a_{ij}| : 1 \leq i, j \leq n\} \leq \max\{|a_{ii}| : 1 \leq i \leq n\}$.

(IV) $a_{ij}^2 < a_{ii}a_{jj}$.

DEMOSTRACIÓN: Ver el Teorema 6.21 del libro de Burden-Faires [4]. □

Teorema 2.3.11 *Sea A una matriz simétrica. Son equivalentes:*

(I) A es definida positiva.

(II) Todos los valores propios de A son estrictamente positivos, $\sigma(A) \subset (0, +\infty)$.

(III) Todos los menores principales de A tienen determinante estrictamente positivo.

(IV) Existe una matriz triangular inferior L con unos en la diagonal, y una matriz diagonal D con elementos estrictamente positivos a lo largo de la diagonal, tal que $A = LDL^t$.

(V) Existe una matriz triangular inferior B con diagonal estrictamente positiva, tal que $A = B^tB$. (Método de Choleski).

DEMOSTRACIÓN: (I) $\Rightarrow$ (II): Si A es PD, los cocientes de Rayleigh $R_A(x) = (x^tAx)/(x^tx) > 0$, y el Teorema 1.2.10 nos dice que $\sigma(A) \subset (0, +\infty)$.

(II) $\Rightarrow$ (I): Sea $D = diag\{\lambda_1, ..., \lambda_n\}$ la matriz diagonal cuyos elementos son los valores propios de A (estrictamente positivos) y O una matriz ortogonal (cambio de base ortonormal) tal que $O^tAO = D$ ($A = ODO^t$), claramente D es PD y lo mismo ocurre con A por el Teorema 2.3.5.

(I) $\Rightarrow$ (III): Razonando como en la prueba del Corolario 2.3.6, los menores principales A_k de A son simétricos y PD con determinante no nulo. Tal y como acabamos de ver, A_k tiene todos sus valores propios positivos, y consecuentemente, tiene determinante estrictamente positivo.

(III) $\Rightarrow$ (IV): De (III) se deduce la existencia de la factorización LU de A, y si recordamos la construcción de la factorización de Dootlittle y el Teorema 2.2.5 comprobamos la existencia de una única matriz triangular inferior L con unos en la diagonal y una matriz diagonal $D = diag\{d_1, ..., d_n\}$ tal que $A = LDL^t$, donde con el convenio $\det(A_0) = 1$, los elementos d_k vienen dados por $d_k = \det(A_k)/\det(A_{k-1}) > 0$.

(IV) $\Rightarrow$ (V): Si existen una única matriz triangular inferior L y una matriz diagonal $D = diag\{d_1, ..., d_n\}$ tal que $A = LDL^t$ con $d_{i,i} > 0$ en la diagonal, consideramos $\sqrt{D}$ la matriz diagonal con elementos $\sqrt{d_{i,i}} > 0$ en la diagonal. Entonces se cumple que $\sqrt{D}.\sqrt{D} = D$.

Tomando $B = \sqrt{D}L^t$ se cumple que

$$B^t.B = L\sqrt{D}\sqrt{D}L^t = LDL^t = A$$

(V) $\Rightarrow$ (I): Como los elementos de la diagonal de B son positivos, se sigue que B es no singular y que $Bx \neq 0$ si $x \neq 0$. Así,

$$x^tAx = x^tB^tBx = (Bx)^t(Bx) = \|Bx\| > 0$$

si $x \neq 0$, i.e. A es definida positiva.

(Ver el teorema 4.4.1 de la sección 4.4 de [5].) □

Si B es la matriz triangular inferior de la factorización de Choleski, $A = B^tB$, para resolver el sistema $Ax = b$ bastará resolver primero el sistema $By = b$ y, una vez obtenida la solución y, continuar resolviendo el sistema $B^tx = y$.

Ejercicio 2.8 Realiza la factorización de Choleski de la matriz

$$A = \begin{pmatrix} 1 & 2 & 3 & 4 \\ 2 & 5 & 1 & 10 \\ 3 & 1 & 35 & 5 \\ 4 & 10 & 5 & 45 \end{pmatrix}$$

Calcula $det(A)$, y resuelve el sistema $Ax = b$ para $b = (1, 0, 0, 0)^t$.

Algoritmo 2.4 **Método de factorización de (Choleski)**

(sólo para matrices simétricas definidas positivas)
Datos de entrada: $A[n][n]$ (Matriz de coeficientes del sistema.);
n (dimensión de A)
Variables: $L[n][n]$; // matriz para escribir la matriz triangular superior.
aux;// una variable auxiliar para sumatorios y productos escalares.
Fujo del programa:

```
aux = 0. ;
for(k=0;k<n;k++){
  aux = A[k][k];
  for(s=0;s<k;s++){ // de Fila k de L por Columna k de L^t.
    aux = aux - L[k][s] * L[k][s];
    }
  if(aux<=0){
    Parada: no hay factorización de Choleski;
    }
  L[k][k] = sqrt(aux);
  for(i=k+1;i<n;i++){// de Fila i de L por Columna k de L^t.
    aux = A[i][k];
    for(s=0;s<k;s++){ aux = aux - L[i][s] * L[k][s];
      }
    L[i][k] = aux/L[k][k];
    }
  }
```

Datos de salida:L (Factorización de Choleski $A = LL^t$)o mensaje de error

Ejercicio 2.9 A la matriz real simétrica $H_n = (h_{i,j})_{n\times n}$, con $h_{i,j} = \frac{1}{i+j-1}$ se la llama matriz de Hilbert de orden n.

(I) Probar que esta matriz es definida positiva y por lo tanto es invertible.

(II) En http://en.wikipedia.org/wiki/Hilbert_matrix tenéis información sobre las matrices de Hilbert, en particular una fórmula para definir sus inversas. Comprobar que la función $cond(H_n)$ crece muy rápidamente con n. Convenceros calculando los números $cond(H_2)$, $cond(H_3)$, $cond(H_4)$ y $cond(H_5)$ (este último ya es muy grande).

2.3.4. Matrices tridiagonales

Definición 2.3.12 *Una matriz cuadrada A de dimensión n se dice que es una* ***matriz banda*** *cuando existen enteros p y q tales que $a_{ij} = 0$ si $i + p \leq j$ o $j + q \leq i$. El ancho de banda de este tipo se define como $w = p + q - 1$.*

¿ Cuáles son las matrices banda con $p = 1$ y $q = 1$ ($w = 1$)?

Las matrices banda que más suelen aparecer en la práctica tienen la forma $p = q = 2$ y $p = q = 4$. Las matrices de ancho de banda 3 con $p = q = 2$ se llaman **matrices tridiagonales** porque su forma es

Teorema 2.3.13 *Si A es una matriz tridiagonal*

$$A = \begin{pmatrix} b_1 & c_1 & & & \\ a_2 & b_2 & c_2 & & \\ & \ddots & \ddots & \ddots & \\ & & a_{n-1} & b_{n-1} & c_{n-1} \\ & & & a_n & b_n \end{pmatrix}$$

se define la sucesión $\delta_0 = 1$, $\delta_1 = 1$, $\delta_k = b_k\delta_{k-1} - a_k c_{k-1}\delta_{k-2}$ $(2 \leq k \leq n)$.

Entonces, $\delta_k = det(A_k)$ (A_k el menor principal de orden k) y si todos los $\delta_k \neq 0$, la factorización LU de la matriz A es

$$A = LU = \begin{pmatrix} 1 & & & & \\ a_2\frac{\delta_0}{\delta_1} & 1 & & & \\ & \ddots & \ddots & \ddots & \\ & & a_{n-1}\frac{\delta_{n-3}}{\delta_{n-2}} & 1 & \\ & & & a_n\frac{\delta_{n-2}}{\delta_{n-1}} & 1 \end{pmatrix} \begin{pmatrix} \frac{\delta_1}{\delta_0} & c_1 & & & \\ & \frac{\delta_2}{\delta_1} & c_2 & & \\ & \ddots & \ddots & \ddots & \\ & & & \frac{\delta_{n-1}}{\delta_{n-2}} & c_{n-1} \\ & & & & \frac{\delta_n}{\delta_{n-1}} \end{pmatrix}$$

Demostración: Basta recordar la construcción de la factorización LU de Dootlittle. (Ver Teorema 4.3.2 de [5]) □

Ejemplo 2.3.14 Si A es una matriz tridiagonal simétrica definida positiva

$$A = \begin{pmatrix} b_1 & a_2 & & & \\ a_2 & b_2 & a_3 & & \\ & \ddots & \ddots & \ddots & \\ & & a_{n-1} & b_{n-1} & a_n \\ & & & a_n & b_n \end{pmatrix}$$

Escribe una algoritmo que proporcione la factorización de Choleski $A = SS^t$

En la sección 4.3 del libro de Ciarlet [5] y en la sección 6.6 del libro de Burden-Faires [4] podéis estudiar la factorización de LU para sistemas tridiagonales. En esa misma sección tenéis información sobre resolución de sistemas con matrices de coeficientes de los distintos tipos que acabamos de presentar.

Ejercicio 2.10 Considera que un objeto puede estar en cualquiera de los $n+1$ puntos $x_0, x_1, ..., x_n$ de un segmento $[x_0, x_n]$, y que cuando el objeto esté situado en la posición x_i tendrá las mismas probabilidades de desplazarse hacia x_{i-1} o aa x_{i+1} sin poder desplazarse a otro lugar. Calcula las probabilidades p_i de que un objeto que parte de x_i llegue al extremo x_0 antes que a x_n. Por supuesto $p_0 = 1$ y $p_n = 0$, y el resto de probabilidades son las soluciones del sistema lineal

$$p_i = 0.5p_{i-1} + 0.5p_{i+1}, \quad i = 1, ...n-1.$$

Resuelve el sistema para n=10,50 y 100, utilizando cualquiera de los métodos directos o iterativos de resolución de ecuaciones [4, Ejercicio 14, pag 453].

2.4. Método de Gauss

Es evidente que al intentar construir una factorización LU, si encontramos un cero en la diagonal el proceso se interrumpe. En este caso es mejor buscar en las filas inferiores a la diagonal una entrada en la columna que sea no cero (pivote) y permutar la fila correspondiente con la fila donde esta el elemento diagonal. También es aconsejable evitar que el pivote sea pequeño en relación al resto de los elementos a anular. Esto es, conviene permutar filas de forma que el elemento de la diagonal que se use para pivotar sea el mayor posible en relación a los elementos a anular. Esto nos lleva a las estrategias de **pivote parcial** o **pivote total** que veremos mas adelante. En esta sección vamos a construir los algoritmos correspondientes a estas dos estrategias analizando su complejidad en términos del número de operaciones.

Tal y como recordaréis del curso de "Álgebra lineal", el método de Gauss para resolver sistemas de ecuaciones

$$Ax = b$$

consiste en pasar a un sistema equivalente donde la matriz de coeficientes es triangular superior $Ux = v$ y resolver este nuevo sistema por el método ascendente.

La triangulación de la matriz de coeficientes se consigue mediante un proceso de eliminación, que partiendo de la matriz $\mathrm{A}=A^{(1)}=\left(a_{\mathrm{ij}}^{(1)}\right)_{i,j=1,...,n}$ va obteniendo matrices $A^{(k)}=\left(a_{\mathrm{ij}}^{(k)}\right)_{i,j=1,...,n}$

$$A^{(k)} = \begin{pmatrix} a_{11}^{(k)} & a_{12}^{(k)} & ... & a_{1k}^{(k)} & ... & a_{nk}^{(k)} \\ 0 & a_{22}^{(k)} & ... & a_{2k}^{(k)} & ... & a_{2n}^{(k)} \\ 0 & 0 & . & . & ... & . \\ & & & a_{kk}^{(k)} & ... & a_{kn}^{(k)} \\ \vdots & \vdots & . & . & ... & . \\ 0 & 0 & ... & a_{nk}^{(k)} & ... & a_{nn}^{(k)} \end{pmatrix}$$

Con el objetivo de obtener la matriz triangular superior U en el último paso $U = A^{(n)}$. Evidentemente, para que el sistema resultante sea equivalente al inicial debemos hacer en la columna de los términos independientes $b^{(k)}$ (con $b = b^1$) las mismas operaciones que en las matrices $A^{(k)}$ y quedarnos con $v = b^{(n)}$. Los pasos a seguir para obtener $A^{(k+1)}$ y $b^{(k+1)}$ partiendo de $A^{(k)}$ y $b^{(k)}$ son los siguientes:

(I) (Elección parcial del pivote) Búsqueda del **pivote**: $|a_{kp}^{(k)}| = \text{máx}\{|a_{kj}^{(k)}| : j = k, ..., n\}$, (el mayor elemento de la columna), con localización de la fila, p, donde se alcanza. Observad que si el pivote es 0, la matriz de coeficientes es singular y el sistema no es compatible determinado.

¿Porqué hacemos esta elección, denominada "elección de pivote parcial", en lugar de detenernos en el primer elemento no nulo de la columna k?

(II) Cambio de fila: se permutan la fila k y p de $A^{(k)}$ y de $b^{(k)}$.

En la construcción del algoritmo se sustituye esta permutación por el uso de un puntero[2] *(permutación)*

$$fila : \{1, 2, .., n\} \to \{1, 2, .., n\}$$

de manera que $fila(k) = p$ apunta a la fila p donde está el puntero sin necesidad de permutar las dos filas.

(III) Eliminación: Se utiliza que el pivote es no nulo para ir anulando los elementos de la columna k bajo la diagonal, restando a cada fila $i = k + 1, ..., n$ de $A^{(k)}$ y de b^k, la correspondiente

[2]En Computación, un "puntero" es un tipo de dato que apunta a una dirección de la memoria física. En "WIKIPEDIA" podéis encontrar una descripción sencilla de puntero o "pointer" (en inglés).

fila k multiplicada por el cociente $\frac{a_{ik}^{(k)}}{a_{kk}^{(k)}}$, dejando invariantes las k primeras filas.

$$A_i^{(k+1)} = A_i^{(k)} - \left(\frac{a_{ik}^{(k)}}{a_{kk}^{(k)}}\right) * A_k^{(k)} \quad \text{y} \quad b_i^{(k+1)} = b_i^{(k)} - \left(\frac{a_{ik}^{(k)}}{a_{kk}^{(k)}}\right) * b_k^{(k)}.$$

El método de Gauss con elección del pivote parcial está descrito en el algoritmo 2.5.

Complejidad del método de Gauss

El número máximo de operaciones, $g(n)$, sin contar las permutaciones de filas o el uso del puntero para reducir el sistema a uno triangular superior dependiendo de la dimensión n del sistema (es el número de operaciones para resolver el sistema por eliminación de Gauss sin permutar filas y cabe esperar que sea del mismo tamaño que el que estimamos antes para hacer la factorización LU más la resolución de dos sistemas triangulares). Este número se puede calcular observando que:

(I) $g(1) = 0$.

(II) Teniendo en cuenta que para pasar de $A^{(1)} = A$ a $A^{(2)}$ y de $b^{(1)} = b$ a $b^{(2)}$, para cada fila $i = 2, ..., n$ de $A^{(1)}$ tenemos que: construir el multiplicador $tau_i = a_{i\,1}/a_{1\,1}$ **(1 división)**, luego para cada uno de los $n-1$ elementos en las columnas $j = 2, ...n$ de la fila hay que hacer una multiplicación y una resta $a_{i\,j}^{(}2) = a_{i\,j} - tau_i * a_{1,j}$ **$(2 * (n-1)$ operaciones)** y en la misma fila de $b^{(1)}$ hay que hacer una multiplicación y una resta $b_i^{(2)} = b_i - \tau_i * b_1$ **(2 operaciones)**. En total **$(2 * n + 1$ operaciones)** por cada fila **$((n-1)$veces)**.

El problema de reducir $A^{(2)}$ equivale a reducir una matriz de dimensión $n-1$ y necesita $g(n-1)$ operaciones. Con lo que tenemos:

$$g(n) = (n-1) * (2 * n + 1) + g(n-1) = 2 * n^2 - n - 1 + g(n-1),$$

(III) $g(n) = 2 * \sum_{i=1}^{n} n^2 - \sum_{i=1}^{n} n - n = \frac{2}{3}n^3 + \frac{1}{2}n^2 - \frac{7}{6}n$.

Ejercicio 2.11 Considera el sistema de ecuaciones

$$\begin{pmatrix} 1 & -1 & 1 & 0 \\ -1 & 2 & -1 & 2 \\ 1 & -1 & 5 & 2 \\ 0 & 2 & 2 & 6 \end{pmatrix} \begin{pmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \end{pmatrix} = \begin{pmatrix} 4 \\ -3 \\ 16 \\ 8 \end{pmatrix}$$

Resuelve el sistema por el método de Gauss, anotando el número de operaciones (sumas, restas, múltiplicaciones y divisiones) que hayas realizado.

Algoritmo 2.5 **Método de Gauss con elección de pivote parcial**

Datos de entrada: $A[n][n]$ (Matriz de coeficientes del sistema.); $b[n]$ (vector término independiente.);
n (dimensión de A y b)
Variables:$B[n][n]$;// una matriz donde ir haciendo todas las modificaciones de A.
$v[n]$;// un vector donde ir haciendo las modificaciones de b.
$fila[n]$;// un vector "puntero" donde anotar las permutaciones de las filas.
// $fila[k]$ "apunta" hacia la fila que ocuparía la posición k
$x[n]$;// un vector donde escribir la solución. **Fujo del programa:**

B=A ; v=b ;// Condiciones iniciales
`for(j=1;j<=n;j++){`
$fila(j) = j$;
}
// Vamos a hacer las etapas $k = 1, 2, ..., n-1$.
`for(k=1;k<n;k++){`
// 1. Elección del pivote (parcial).
p=k; //iniciamos la busqueda en la fila k
`for(i=k+1;i<=n;i++){`
`if(`$|B_{fila(i),k}| > |B_{fila(p),k}|$`){`
p=i;//apuntamos a la fila donde está el puntero
}
}
`if(`$B_{fila(p),k} == 0$`){`
Parada, Error: A es singular
}
// 2. Intercambio de filas (virtual). Cambios en el puntero.
$m = fila(k)$; $fila(k) = fila(p)$; $fila(p) = m$;
// 3. Eliminación.
`for(i=k+1;i<=n;i++){`
$mul = B_{fila(i),k}/B_{fila(k),k}$;
$B_{fila(i),k} = 0$;//Asignamos el valor 0 en lugar de hacer las operaciones
`for(j=k+1;j<=n;j++){`
$B_{fila(i),j} = B_{fila(i),j} - mul * B_{fila(k),j}$;
}
$v_{fila(i)} = v_{fila(i)} - mul * v_{fila(k)}$;
}
}
// Resolvemos el sistema por el método ascendente.
`for(k=n;k>=1;k-){`
//$x_k = (v_{fila(k)} - \sum_{j=k+1}^{n}(B_{fila(k),j} * x_j))/B_{fila(k),k}$
$x_k = v_{fila(k)}$;
$j = k + 1$;
`while(j<=n){`
$x_k = x_k - B_{fila(k),j} * x_j$;
j++ ; }
$x_k = x_k/B_{fila(k),k}$;
}

Datos de salida: Solución x del sistema o mensaje de error si A es singular.

Cálculo de determinantes e inversas

Ejercicio 2.12 Considera la matriz de coeficientes del sistema del ejercicio 2.11. Utiliza el método de triangulación de Gauss para calcular el determinante de A, anotando el número de operaciones (sumas, restas, múltiplicaciones y divisiones) que hayas realizado.

Tal y como hacíais en la asignatura de álgebra lineal, se puede modificar el algoritmo para calcular la matriz inversa de A. Aunque el cálculo de esta es equivalente a resolver los sistemas $Ax = e_i$ $(i = 1, ..., n)$ donde e_i es la base canónica de $\mathbb{K}^n$.

Otra variante que se puede realizar es el método de Gauss-Jordan, que consiste en realizar también el proceso de eliminación de los elementos que están sobre la diagonal, transformando el sistema de ecuaciones en uno equivalente donde la matriz de coeficientes va a ser diagonal.

Ejercicio 2.13 Considera otra vez la matriz de coeficientes del sistema del ejercicio 2.11. Utiliza el método de Gauss para calcular la matriz inversa de A.

2.4.1. Gauss con pivote total

El método de Gauss se puede optimizar al cambiar la estrategia seguida en la elección del pivote, buscando el elemento de mayor módulo entre las filas y columnas $i \geq k$ y $j \geq k$:

$$|a_{mp}^{(k)}| = \text{máx}\{|a_{ij}^{(k)}| : i = k, ..., n; j = k, ..., n\},$$

La finalidad es evitar hacer divisiones por números muy pequeños que pueden producir números muy grandes y cálculos inestables al operar con números de distinto tamaño. De esta manera construimos el algoritmo de Gauss de pivote total.

Como en el caso del pivote parcial, en lugar de permutar columnas en la matriz, lo que significaría permutar filas en el vector solución x, lo que se hace en el algoritmo 2.6 es utilizar un puntero donde señalar la posición de cada columna.

El método de Gauss con elección de pivote total está descrito en el algoritmo 2.6.

Ejercicio 2.14 Escribe un algoritmo especial de eliminación gausiana para resolver sistemas de ecuaciones lineales cuando la matriz de coeficientes A es tridiagonal (i.e. $A = (a_{i,j})$, $a_{i,j} = 0$ si $|i - j| > 1$).

Algoritmo 2.6 Método de Gauss con elección de pivote total

Datos de entrada: $A[n][n]$ (Matriz de coeficientes del sistema.); $b[n]$ (vector término independiente.);
n (dimensión de A y b)
Variables:$B[n][n]$;
$v[n]$;
fila$[n]$;// un vector "puntero" donde anotar las permutaciones de las filas.
colu$[n]$ // // un vector "puntero" donde anotar las permutaciones de las columnas.
$x[n]$;// un vector donde escribir la solución. **Fujo del programa:**

B=A ; v=b ;// Condiciones iniciales
`for(j=1;j<=n;j++){`
fila$(j) = j$; colu$(j) = j$
`}`
// Vamos a hacer las etapas $k = 1, 2, ..., n-1$.
`for(k=1;k<n;k++){`
// 1. Elección del pivote (total).
p=k; q=k //buscamos desde la fila k y la columna k
`for(i=k;i<=n;i++){`
`for(j=k;j<=n;j++){`
`if(`$|B_{\texttt{fila}(i),\texttt{colu}(j)}| > |B_{\texttt{fila}(p),\texttt{colu}(q)}|$`){`
p=i; q=j;
`}`
`}`
`}`
`if(`$B_{\texttt{fila}(p),\texttt{colu}(q)} == 0$`){`
Parada, Error: A es singular
`}`
// 2. Intercambio de filas y columnas. Cambios en los punteros.
$m =$ fila(k) ; fila$(k) =$ fila(p) ;fila$(k) =$ fila(p) ; fila$(p) = m$;
$m =$ colu(k) ; colu$(k) =$ colu(q) ;colu$(k) =$ colu(q) ; colu$(q) = m$;
// 3. Eliminación.
`for(i=k+1;i<=n;i++){`
$mul = B_{\text{fila}(i),\text{colu}(k)}/B_{\text{fila}(k),\text{colu}(k)}$;
$B_{\text{fila}(i),\text{colu}(k)} = 0$;//Asignamos el valor 0 en lugar de hacer las operaciones
`for(j=k+1;j<=n;j++){`
$B_{\text{fila}(i),\text{colu}(j)} = B_{\text{fila}(i),\text{colu}(j)} - mul * B_{\text{fila}(k),\text{colu}(j)}$;
`}`
$v_{\text{fila}(i)} = v_{\text{fila}(i)} - mul * v_{\text{fila}(k)}$;
`}`
`}`
// Resolvemos el sistema por el método ascendente.
`for(k=n;k>=1;k-){`//$x_{\text{colu}(k)} = (v_{\text{fila}(k)} - \sum_{j=k+1}^{n}(B_{\text{fila}(k),\text{colu}(j)} * x_{\text{colu}(j)}))/B_{\text{fila}(k),\text{colu}(k)}$
$x_{\text{colu}(k)} = v_{\text{fila}(k)}$;
$j = k + 1$;
`while(j<=n){`
$x_{\text{colu}(k)} = x_{\text{colu}(k)} - B_{\text{fila}(k),\text{colu}(j)} * x_{\text{colu}(j)}$;
j++ ;
`}`
$x_{\text{colu}(k)} = x_{\text{colu}(k)}/B_{\text{fila}(k),\text{colu}(k)}$;
`}`

Datos de salida: Solución x del sistema o mensaje de error si A es singular.

2.5. Factorización QR

2.5.1. Transformaciones de Householder

Definición 2.5.1 *Se llaman* **matrices de Householder** *a las matrices de la forma*

$$H(v) = Id - \frac{2}{v^t v} v v^t, \quad v \neq 0 \text{ un vector de } \mathbb{K}^n,$$

$$H(0) = Id.$$

Geométricamente, el producto $H(v)a$ representa a la "reflexión especular" de a con respecto a la dirección de v, es decir, el vector simétrico del vector a con respecto al vector normal a la dirección generada por v y dentro del plano determinado por los vectores a y v. En efecto:

(I) Observemos que si $y = H(v)a$ entonces

$$y = a - \frac{2}{v^t v} v v^t a = a - \frac{2v^t a}{\|v\|_2^2} v$$

por lo tanto, dada la dirección v la matriz $H(v)$ no se tiene que construir de forma explícita. Lo único que interesa es su acción sobre a que se calcula de esta forma

$$H(v)a = a - \frac{2v^t a}{\|v\|_2^2} v. \tag{2.6}$$

(II) Veamos el significado: como $v^t a = \|v\|\|a\|cos(\alpha)$ donde α es el ángulo entre a y v entonces tenemos que la proyección del vector a sobre la dirección dada por v es $P_v a = \|a\|cos(\alpha)u_v$ con $u_v = v/\|v\|_2$. Por lo tanto, $P_v a = \frac{v^t a}{\|v\|_2^2} v$ y

$$H(v)a = a - 2P_v a$$

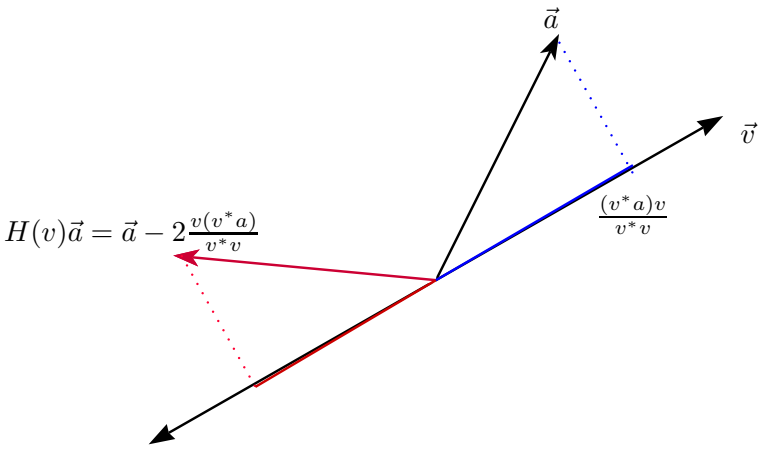


Figura 2.1: Reflexión especular $H(v)a$.

En otras palabras, al vector a se le resta dos veces su proyección sobre la dirección de v. Esto genera exactamente la reflexión de x con respecto a la ortogonal de la dirección generada por v (Figura 2.1).

Ejercicio 2.15 Demuestra que las matrices $H(v)$ son unitarias y simétricas.

Esto nos va a permitir reflejar un vector sobre una dirección adecuada para poder anular todas sus coordenadas menos la primera, ese sera nuestro resultado principal:

Teorema 2.5.2 *Sea a un vector de $\mathbb{K}^n$. Entonces existen dos matrices de Householder $H_\pm$ tales que $H_\pm a = (\mp\|a\|, 0, 0, ..., 0)^t$.*

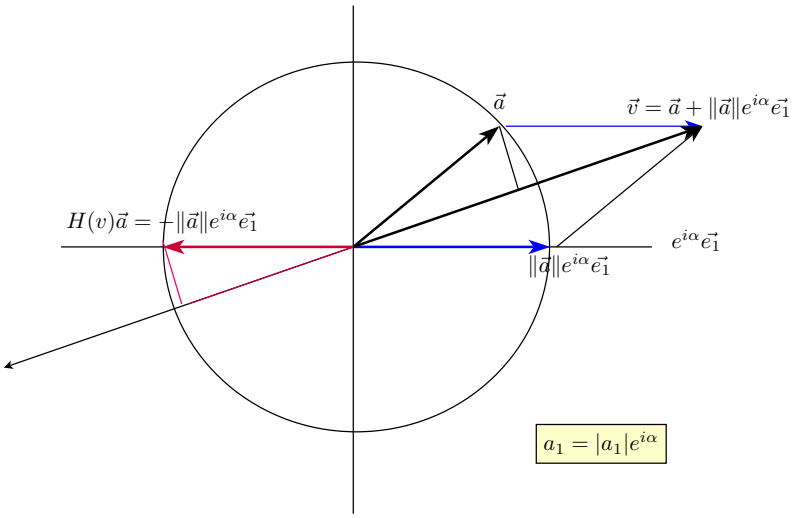


Figura 2.2: Prueba geométrica del Teorema 2.5.2

DEMOSTRACIÓN: Recordad que

$$H_v a = a - \frac{2v^t a}{\|v\|_2^2} v.$$

Si queremoa que $H_v a$ esté en el subespacio generado por el vector de la base canónica $e_1 = (1, 0, ..., 0)^t$ entonces tendremos que tomar v en el plano que generan los vectores a y e_1:

$$v \in span\{e_1, a\} \Rightarrow v = a + \alpha e_1$$

para algún α que hay que determinar. Observemos que si $a = (a_1, .., a_n)^t$ entonces $e_1^t a = a_1$ y

$$v^t a = \|a\|^2 + \alpha a_1, \quad v^t v = (a + \alpha e_1)^t (a + \alpha e_1) = \|a\|_2^2 + 2\alpha a_1 + \alpha^2.$$

Por lo tanto,

$$H_v a = a - 2\frac{\|a\|^2 + \alpha a_1}{\|a\|_2^2 + 2\alpha a_1 + \alpha^2}(a + \alpha e_1)$$

de donde

$$H_v a = \big(1 - 2\frac{\|a\|^2 + \alpha a_1}{\|a\|_2^2 + 2\alpha a_1 + \alpha^2}\big)a - 2\alpha\frac{\|a\|^2 + \alpha a_1}{\|a\|_2^2 + 2\alpha a_1 + \alpha^2} e_1.$$

Aquí resulta claro que si tomamos

$$\alpha = \pm\|a\|_2$$

entonces el coeficiente de a se anula y obtenemos

$$H_v a = -\alpha e_1 = \mp\|a\|_2 e_1.$$

Por lo tanto, con $\alpha = \pm\|a\|_2$ y $v = a + \alpha e_1$ tenemos que

$$H_v a = \mp\|a\|_2 e_1.$$

□

En la practica, para calcular $H(v)b$ no se calcula la matriz $H(v)$ sino su acción sobre b directamente mediante la expresión

$$H(v)b = b - 2\frac{v^t b}{\|v\|^2}v$$

esto es,

(I) se calcula la `norma` $\|v\|^2 = v^t v$,

(II) se calcula el `producto escalar` $v^t b$

(III) por último se calcula $b - 2\frac{v^t b}{\|v\|^2}v$.

Por ejemplo, si $x = (3, 1, 5, 1)^t$ y tomamos $\|x\|_2^2 = 36 = 6^2$. Entonces si tomamos

$$v_\pm = x \pm 6e_1$$

resulta que $H_\pm x = (\mp 6, 0, 0, 0)^t$ donde

$$H\pm = I - 2\frac{v_\pm v_\pm^t}{v_\pm^t v_\pm}$$

pero no necesitamos calcularla explícitamente.

Observación 2.5.3 Conviene tomar como α el valor $\alpha = -\|a\|_2$ para así obtener $H_- a = (\|a\|_2, 0, 0, .., 0)^t$. Pero entonces el vector asociado a la transformación de Householder tiene la expresión

$$v = a - \|a\|e_1 = (a_1 - \|a\|_2, a_2, .., a_n)^t.$$

Entonces, si $a \approx e_1$ podemos tener cancelaciones en el caso $a_1 > 0$ que originen pérdida de precisión. Es por ello aconsejable trabajar mejor con

$$a_1 - \|a\|_2 = \frac{a_1^2 - \|a\|_2^2}{a_1 + \|a\|_2} = -\frac{a_2^2 + a_3^2 + ... + a_n^2}{a_1 + \|a\|_2}.$$

Por otro lado, sabemos que si $H(v) = I - \beta v v^t$ con $\beta = \frac{2}{\|v\|_2^2}$, entonces para cualquier matriz $A \in \mathcal{M}_{nm}(\mathbb{K})$

$$H(v)A = A - vw^t, \quad \text{con } w = \beta A^t v,$$

y también para cualquier matriz $B \in \mathcal{M}_{mn}(\mathbb{K})$

$$BH(v) = B - zv^t \quad \text{con } z = \beta B v.$$

por lo tanto, para multiplicar $H(v)$ por una matriz por la derecha o por la izquierda tampoco es necesario construir la transformación de Householder de manera explícita.

2.5.2. Factorización QR usando las transformaciones de Householder

Dada $A \in \mathbb{K}^{m\times n}$ una factorización QR de A es una de la forma $A = QR$ donde $Q \in \mathbb{K}^{m\times m}$ es una matriz ortogonal y $R \in \mathbb{K}^{m\times n}$ es triangular superior.

Si $p = \text{mín}\{m, n\}$, el **método de Householder** para obtener la factorización consiste en encontrar p matrices de Householder $H_1, ..., H_p$ de manera que $R = H_p...H_2H_1A$ sea una matriz triangular superior. Tomando entonces la matriz ortogonal $Q = (H_p...H_2H_1)^t = H_1^tH_2^t...H_p^t$ se tiene $A = QR$.

- Pongamos $R_0 = A$ y $Q_0 = H_0 = Id$.
- Pongamos $Q_k = H_kQ_{k-1}$ $R_k = H_kR_{k_1}$, y supongamos que presenta la forma:

$$R_k = \begin{pmatrix} x & x & x & x & x & x & x & x \\ & x & x & x & x & x & x & x \\ & & x & x & x & x & x & x \\ & & & x & x & x & x & x \\ & & & x & x & x & x & x \\ & r_k^{\rightarrow} & & x & x & x & x & x \\ & & & x & x & x & x & x \\ & & & x & x & x & x & x \end{pmatrix} \leftarrow \text{fila } k$$

$$\uparrow \text{columna } k$$

- Sea $\widetilde{R}_k$ la matriz formada por los elementos r_{ij} de R_k que están en las filas $i \geq k$ y en las columna $j \geq k$. Sea $\vec{r_k} = \begin{pmatrix} r_{kk} \\ \vdots \\ r_{mk} \end{pmatrix} \in \mathbb{K}^{m-k+1}$, y $\widetilde{H}_{k+1} = H(\widetilde{v}_k)$ la matriz de Householder que da el teorema 2.5.2 de manera que $\widetilde{H}_{k+1}\vec{r_k} = \begin{pmatrix} 1 \\ 0 \\ \vdots \end{pmatrix}$.
- Sea

$$H_{k+1} = \left(\begin{array}{c|c} Id_{k-1} & 0 \\ \hline 0 & \widetilde{H}_{k+1} \end{array}\right),$$

  H_k es la matriz de Householder $H(v_k)$ con $v_k = \begin{pmatrix} 0 \\ \vdots \\ \widetilde{v}_k \end{pmatrix}$.
- Construimos $Q_{k+1} = H_{k+1}Q_k$, y $R_{k+1} = H_{k+1}R_k = Q_{k+1}A$ y .
- Reiteramos el proceso hasta obtener la matriz triangular $R = R_p = Q_pA$ y poniendo $Q = Q_p^t$ tendremos la factorización $A = QR$ que siempre existe.

Teorema 2.5.4 *Si $A = QR$ con $rango(A) = n$ con vectores columna en $A = (\vec{a_1}, ..., \vec{a_n})$ y $Q = (\vec{q_1}, ..., \vec{q_m})$ entonces*

$$span\{\vec{a_1}, ..., \vec{a_k}\} = span\{\vec{q_1}, ..., \vec{q_k}\}, \quad k = 1, .., n$$

Además, si Q_1 es el menor de Q formado por las n primeras columnas y Q_2 es el menor de Q formado por las columnas $n+1, n+2, ..., m$ entonces $Im(A) = \{Ax : x \in \mathbb{R}^n\} = Im(Q_1) = \{Q_1x : x \in \mathbb{R}^n\}$ e $Im(A)^{\perp} = Im(Q_2) = \{Q_2y : y \in \mathbb{R}^{m-n}\}$. Además, si R_1 es el menor de R formado por las n primeras filas, entonces $A = Q_1R_1$.

Algoritmo 2.7 **Método de Householder. Factorización QR**

Datos de entrada: $A[m][n]$ (Matriz de m filas y n columnas.)
m (numero de filas de A)
Variables:$R[m][n]$;// una matriz donde ir haciendo las modificaciones de A.
$Q[m][m]$; // matriz donde ir guardando el producto de las matrices de Householder.
aux;// una variable auxiliar para sumatorios y productos escalares.
$sign$;// una variable para el signo $\overline{R_{k,k}}/|R_{k,k}|$.
$norma$;// una variable real para la norma del vector $\vec{a}$.
$v[m]$;// un vector para definir las transformaciones de Householder. $norma2v$;// una variable real para $\|\vec{v}\|^2$.
Fujo del programa:
R=A; $Q[m][m] = Id$ // Condiciones iniciales
$N = \min\{n, m\}$ si $m > n$ y $N = N - 1$ en otro caso.
// Vamos a hacer las etapas $k = 1, 2, ..., N$.
`for(k=1;k<N;k++){`
// Vamos a hacer $\sum_{i=k+1}^{m} |R_{i,k}|$.
$aux = |R_{k+1,k}|$;
`for(i=k+2;i<=m;i++){` $aux = aux + |R_{i,k}|$; `}`
`if(`$aux == 0$`){ if(`$|R_{k,k}| == 0$`){ Error «Matriz no tiene rango máximo» Fin;}`
Continue; // Pasar a la siguiente etapa $k + 1$ del bucle. `}`
`if(`$|R_{k,k}| > 0$`){` $signo = \overline{R_{k,k}}/|R_{k,k}|$;`} else{` $signo = 1$;`}`
$norma = |R_{k,k}|^2$; // Vamos a hacer $\sum_{k}^{n} |R_{i,k}|^2$.
`for(i=k+1;i<=n;i++){` $norma = norma + |R_{i,k}|^2$;`}`
$norma = \sqrt{norma}$; $v[k] = R_{k,k} + norma * signo$; // 1. vector de Householder.
`for(i=k+1;i<=m;i++){`$v[i] = R_{i,k}$;`}`
$norma2V = 2(norma)^2 + 2 * norma * signo * R_{k,k}$;
$B_{k,k} = -norma * signo$; // 2. Acción de la simetría en columna k.
`for(i=k+1;i<=n;i++){` $B_{i,k} = 0$; `}`
`for(j=k+1;j<=n;j++){` // Acción en las demás columnas de R.
$aux = \overline{v[k]} * R_{k,j}$; //$\vec{v}.\vec{R^j}$.
`for(i=k+1;i<=n;i++){` $aux = aux + \overline{v[i]} * R_{i,j}$; `}`
$aux = 2 * aux/norma2V$
`for(i=k;i<=m;i++){` $R_{i,j} = R_{i,j} - aux * v[i]$; `}`
`}`

`for(j=1;j<=m;j++){` // Acción en las columnas de la matriz Q.
$aux = \overline{v[k]} * Q_{k,j}$; //$\vec{v}.\vec{Q^j}$.
`for(i=k+1;i<=n;i++){` $aux = aux + \overline{v[i]} * Q_{i,j}$; `}`
$aux = 2 * aux/norma2V$
`for(i=k;i<=m;i++){` $Q_{i,j} = Q_{i,j} - aux * v[i]$; `}`
`}`
`}`
$Q = Q^t$ **Datos de salida:** Matrices Q y R de la factorización $A = QR$.

2.5.3. Aplicación de QR a la resolución de sistemas lineales

El **método de Householder** para la resolución de un sistema $Ax = b$ consiste en encontrar $(n-1)$ matrices de Householder $H_1, ..., H_{n-1}$ de manera que $H_{n-1}...H_2H_1A$ sea una matriz triangular, y la solución del sistema es la solución de $H_{n-1}...H_2H_1Ax = H_{n-1}...H_2H_1b$ que se obtiene por el método ascendente.

Para escribir el algoritmo, bastará con seguir el algoritmo 2.7 de factorización QR, teniendo en cuenta que en este caso la matriz $A \in \mathcal{M}_n$ es cuadrada, que sólo necesitamos ir guardando en cada etapa las modificaciones de A, $R_k = H_kR_{k-1}$ con $R_0 = A$, y las modificaciones del vector b, $w_k = H_kw_{k-1}$ con $w_0 = b$. Al final se resuelve el sistema triangular $R_{n-1}x = w_{n-1}$ por el método ascendente.

Aunque este método es algo más costoso que el método LU o el método de Gauss, puede resultar interesante cuando tengamos problemas de estabilidad en los cálculos porque como las matrices de Householder son ortogonales (unitarias en el caso complejo)

$$cond_2(A) = cond_2(R_0) = cond_2(R_1) = ... = cond_2(R_{n-1})$$

En otras palabras. Al triangular la matriz con el método de Householder no varia el condicionamiento del problema.

Ejercicio 2.16 Repasa el algoritmo del método de Householder y calcula el número máximo de operaciones a realizar para triangular la matriz de coeficientes y para resolver el sistema lineal.

Ejercicio 2.17 Modifica el algoritmo de Householder para que dé como datos de salida las matrices de la factorización $A = QR$, en concreto: Q, Q^{-1} y R.

Ejercicio 2.18 También modificando el algoritmo de Householder calcula el determinante de una matriz

Ejercicio 2.19 *(No hay dos sin tres)* Resuelve el sistema de ecuaciones del ejercicio 2.6 por el método de Householder.

Algoritmo 2.8 **Método de Householder. Factorización QR**

Datos de entrada: $A[n][n]$ (Matriz de coeficientes del sistema.); $b[n]$ (vector término independiente.); n (dimensión de A y b)

Variables: $R[n][n]$;// una matriz donde ir haciendo las modificaciones de A.
$w[n]$;// un vector donde ir haciendo las modificaciones de b.
aux;// una variable auxiliar para sumatorios y productos escalares.
$sign$;// una variable para el signo $\overline{R_{k,k}}/|R_{k,k}|$.
$norma$;// una variable real para la norma del vector $\vec{a}$.
$v[n]$;// un vector para definir las transformaciones de Householder.
$norma2V$;// una variable real para $\|\vec{v}\|^2$.
$x[n]$;// un vector donde escribir la solución.

Fujo del programa:

R=A ; w=b ;// Vamos a hacer las etapas $k = 1, 2, ..., n-1$.
`for(k=1;k<n;k++){`
// Vamos a hacer $\sum_{k+1}^{n} |R_{i,k}|$.
$aux = |B_{k+1,k}|$;
`for(i=k+2;1<=n;i++){` $aux = aux + |B_{i,k}|$; `}`
`if(`$aux == 0$`){ if(`$|B_{k,k}| == 0$`){ Error «Matriz Singular» Fin;}`
Continue; // Pasar a la siguiente etapa $k+1$ del bucle. `}`
`if(`$|B_{k,k}| > 0$`){` $signo = \overline{B_{k,k}}/|B_{k,k}|$`;} else{` $signo = 1$`;}`
$norma = |B_{k,k}|^2$; // Vamos a hacer $\sum_{k}^{n} |B_{i,k}|^2$.
`for(i=k+1;i<=n;i++){` $norma = norma + |R_{i,k}|^2$`;}`
$norma = \sqrt{norma}$; $v[k] = R_{k,k} + norma * signo$; // 1. vector de Householder.
`for(i=k+1;i<=n;i++){`$v[i] = R_{i,k}$`;}`
$norma2V = 2(norma)^2 + 2 * norma * signo * R_{k,k}$;
$R_{k,k} = -norma * signo$; // 2. Acción de la simetría en columna k.
`for(i=k+1;i<=n;i++){` $R_{i,k} = 0$; `}`
`for(j=k+1;j<=n;j++){` // Acción en las demás columnas.
$aux = \overline{v[k]} * R_{k,j}$; //$\vec{v}.\vec{R^j}$.
`for(i=k+1;i<=n;i++){` $aux = aux + \overline{v[i]} * R_{i,j}$; `}`
$aux = 2 * aux/norma2V$
`for(i=k;i<=n;i++){` $R_{i,j} = R_{i,j} - aux * v[i]$; `}`
`}`
$aux = \overline{v[k]} * w[k]$; // Acción en el vector independiente.
`for(i=k+1;i<=n;i++){` $aux = aux + \overline{v[i]} * w[i]$;`}` //$\vec{v}.\vec{w}$
$aux = 2 * aux/norma2V$
`for(i=k;i<=n;i++){` $w[i] = w[i] - aux * v[i]$;`}`
`}`
// 3 Resolvemos el sistema por el método ascendente. $x =$ solución de Resolver por el método ascendente $Rx = w$ (Algoritmo 2.5).

Datos de salida: Solución x del sistema o mensaje de error si A es singular.

2.6. Problemas de mínimos cuadrados

En la unidad 2 de la asignatura anterior (Cálculo numérico) abordamos la cuestión de cómo aproximar funciones mediante funciones sencillas (polinomios) utilizando técnicas de **interpolación**, analizamos los errores cometidos y vimos cómo una elección adecuada de los nodos, utilizando polinomios de Tchevichev, proporciona la "mejor" aproximación que se puede hacer de una función con polinomios utilizando la "interpolación". Esta era una de las dos formas de enfocar el problema que allí anunciamos: fijar una familia de puntos de la función y encontrar el polinomio de grado más pequeño posible que pasa por dichos puntos.

El enfoque alternativo es fijar, con independencia del número de puntos disponible, el grado del polinomio para escoger a continuación, de entre todos los polinomio de grado a lo sumo el fijado, el que "mejor aproxima" (en términos de mínimos cuadrados o de aproximación uniforme) a la familia de puntos.

2.6.1. Modelo General de los problemas de aproximación

Comenzaremos enunciando el problema de aproximar una función por un polinomio de dos maneras diferentes, en función de la distancia utilizada para medir la aproximación

Ejemplo 2.6.1 (Aproximación uniforme y aproximación por mínimos cuadrados) Dada una función $f \in E = C[a,b]$, encontrar $p \in G = P_n$ (polinomios de grado $\leq n$) tal que:

$$\|f - p\| = \text{ínf}\{\|f - q\| : q \in P_n\}$$

Para la norma del supremo $\| \|_\infty$ se tiene

$$\max_{x\in[a,b]} \{|f(x) - p(x)|\} = \min_{q\in P_n} \max_{x\in[a,b]} \{|f(x) - q(x)|\}.$$

métodos de optimización **minmax**

Para la norma de L^2, $\| \|_2$ se tiene

$$\left(\int_a^b |f(x) - p(x)|^2 dx\right)^{1/2} = \min_{q\in P_n} \left(\int_a^b |f(x) - q(x)|^2 dx\right)^{1/2}.$$

métodos de optimización de **mínimos cuadrados**

Definición 2.6.2 (Modelo General de los Problemas de Aproximación) *Dado un espacio vectorial normado $(E, \| \|)$ un subespacio $G \subset E$ y un vector $f \in E$, Si se puede encontrar $g \in G$ tal que*

$$\|f - g\| = \text{ínf}\{\|f - h\| : h \in G\}$$

diremos que g ***es una MEJOR APROXIMACION de f en G***

Cuando estamos trabajando en dimensión finita el problema de la mejor aproximación siempre tiene solución:

Teorema 2.6.3 *Dado un espacio vectorial normado $(E, \| \|)$ y un subespacio $G \subset E$ de* ***dimensión finita****, entonces para cada $f \in E$ existe al menos una mejor aproximación en $g \in G$*

$$\|f - g\| = \text{ínf}\{\|f - h\| : h \in G\}$$

Demostración:
Ideas que intervienen

- *Para cada $f \in E$ se tiene que $0 \in \{g \in G : \|f - g\| \leq \|f\|\} =: K \neq \emptyset$ es un subconjuto compacto no vacío,*
- *K es compacto porque es cerrado y acotado y la dimensión $dim(G) < \infty$*
- *existe $g_n \in K$: $\|f - g_n\| \to \text{ínf}\{\|f - h\| : h \in G\}$.*
- *POR LA COMPACIDAD DE K existe una subsucesión convergente, $g_{n_k} \to g \in K$.*
- *g es una mejor aproximación de f.*

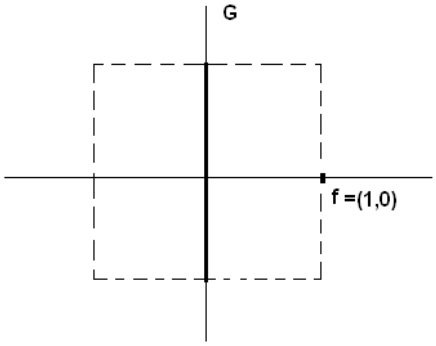
$$\|f - g\| = \lim_k \|f - g_{n_k}\| = \text{ínf}\{\|f - h\| : h \in G\}$$

□

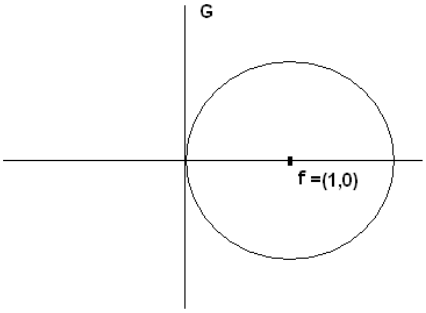
Observad en el ejemplo siguiente que el que la mejor aproximación sea única depende de la norma considerada.

Ejemplo 2.6.4 (Una o varias mejores aproximaciones)

Sean $E = \mathbb{R}^2$, $G = \{(0, y) : y \in \mathbb{R}\}$ y $f = (1, 0)$:
Con la norma $\| \|_\infty$,



todos los puntos de $[(0, -1), (0, 1)]$ son MA de f en G.
Con la norma euclidea $\| \|_2$,



sólo $(0, 0)$ es la MA de f en G

2.6.2. Aproximación por mínimos cuadrados

Se denominan problemas de aproximación por mínimos cuadrados a los problemas de aproximación en espacios normados completos, dónde la norma está definida por un producto escalar, en general, tienen solución única y se pueden resolver a través de sistemas de ecuaciones lineales.

Recordemos las definiciones:

Definición 2.6.5 (Espacio normado con producto escalar) *Sea E un espacio vectorial sobre $\mathbb{R}$. Un producto escalar en E es una aplicación bilineal $<,>: E \times E \to \mathbb{R}$ tal que*

(I) $< f, g >=< g, f >$ *para cada par* $f, g \in E$.

(II) $< f, ah + bg >= a < f, h > +b < f, g >$ *para* $f, h, g \in E$ *y* $a, b \in \mathbb{R}$

(III) $< f, f >> 0$ *para cada* $f \in E, f \neq 0$.

$$\|f\| = \sqrt{< f, f >} \textbf{ define una norma en } E$$

Un espacio de Hilbert es un espacio normado con producto escalar que es ***COMPLETO***

Ejemplo 2.6.6 (Espacios normados con producto escalar)

$E = \mathbb{R}^n$, con la norma euclídea es un espacio de Hilbert.

$$< (x_i), (y_i) >= \sum_{i=1}^{n} x_i y_i$$

$$\|(x_i)\|_2 = \sqrt{\sum_{i=1}^{n} x_i^2}$$

$E = C([a, b], w)$ donde $w \in C([a, b]), w \neq 0, w(x) \geq 0$

$$< f, g >= \int_a^b f(x)g(x)w(x)\,dx$$

$$\|f\|_2 = \sqrt{\int_a^b f(x)^2 w(x)\,dx}$$

$C([a, b], w)$ con la norma $\| \|_2$ no es completo.

Definición 2.6.7 (Ángulos y Ortogonalidad) *Sea E un espacio normado con producto escalar:*

Para cada par $f, g \in E$ se define el ángulo que forman, α, por la fórmula

$$< f, g >= \|f\| \, \|g\| \cos\alpha$$

y se dice que f y g son ortogonales $f \perp g$ cuando

$$< f, g >= 0$$

Proposición 2.6.8 (Propiedades del producto escalar) *Sea E un espacio normado con producto escalar:*

(I) $\|f+g\|^2 = \|f\|^2 + \|g\|^2 + 2 < f, g >$ *para cada par* $f, g \in E$.

(II) $f \perp g \Leftrightarrow \|f+g\|^2 = \|f\|^2 + \|g\|^2$ ***(Pitagoras)***.

(III) $| < f, g > | \leq \|f\| \, \|g\|$ ***(Cauchy-Schwartz)***.

(IV) $\|f+g\|^2 + \|f-g\|^2 = 2\|f\|^2 + 2\|g\|^2$ ***(Identidad del Paralelogramo)***.

La identidad del paralelogramo equivale a que la norma está asociada a un producto escalar

Identidad del Paralelogramo $\Longleftrightarrow$ producto escalar

$$< f, g >:= \frac{1}{4}(\|f+g\|^2 - \|f-g\|^2)$$

2.6.2.1. *Proyecciones ortogonales*

El problema de la mejor aproximación por mínimos cuadrados para subconjuntos convexos y completos siempre tiene solución:

Teorema 2.6.9 (Proyección ortogonal I) *Sean* $(E, \| \|)$ *un espacio normado con producto escalar, y* $C \subset E$ *un subconjunto* ***convexo y completo****. Entonces*

(I) *Existe un único elemento* $g \in C$ *con* $\|g\| = \min\{\|h\| : h \in C\}$.

(II) *Para cada* $f \in E$ *existe una única mejor aproximación* $g \in C$ *de* f *en* C*:*
$\|f-g\| = \min\{\|f-h\| : h \in C\} = \min\{\|t\| : t \in f - C\}$.

Observa que $I) \Rightarrow II)$ $\quad f - C = \{f - h : h \in C\}$ es convexo y completo. El elemento $f - g$ de norma mínima de $f - C$ determina el vector $g \in C$ mejor aproximación de f en C.

Demostración:

Ideas que intervienen

- *: Existe un único elemento* $g \in C$ *con* $\|g\| = \min_{h \in C}\{\|h\|\}$:
  - *Sea* $\alpha = \inf\{\|h\| : h \in C\}$. *Para cada par* $h, g \in C$, *por la Identidad del Paralelogramo:*
    $\frac{1}{4}\|g-h\|^2 = \frac{1}{2}\|g\|^2 + \frac{1}{2}\|h\|^2 - \frac{1}{4}\|g+h\|^2$.
    $\frac{1}{4}\|g-h\|^2 = \frac{1}{2}\|g\|^2 + \frac{1}{2}\|h\|^2 - \|\frac{g+h}{2}\|^2$
    $\frac{1}{4}\|g-h\|^2 \leq \frac{1}{2}\|g\|^2 + \frac{1}{2}\|h\|^2 - \alpha^2$
    porque $\frac{g+h}{2} \in C$ *(*C *es* ***convexo****)* $\Rightarrow \|\frac{g+h}{2}\| \geq \alpha$.
  - *Unicidad: Si* $\|g\| = \|h\| = \alpha \Rightarrow \|g-h\| = 0$.
  - *Existencia: Sea* $g_n \in C$ *tal que* $\|g_n\| \to \alpha$, *entonces:*
    $\frac{1}{4}\|g_n - g_m\|^2 \leq \frac{1}{2}\|g_n\|^2 + \frac{1}{2}\|g_m\|^2 - \alpha^2 \to 0$ *si* $n, m \to \infty$,
    $\Rightarrow$ g_n *es de Cauchy en* C ***(completo)***, *y por lo tanto convergente hacia* $g \in C$ *y tomando límites:* $\|g\| = \lim \|g : n\| = \alpha$

□

La mejor aproximación en mínimos cuadrados en un subespacio vectorial que proporciona el teorema anterior se puede caracterizar en términos de ortogonalidad:

Teorema 2.6.10 (Proyección Ortogonal II) *Sean $(E, \|\ \|)$ un espacio normado con producto escalar, y $G \subset E$ un subespacio cerrado. Entonces, para cada $f \in E$, son equivalentes:*

(I) $g \in G$ es la mejor aproximación de f en G.

(II) $f - g \perp G$ ($f - g \perp h$ para cada $h \in G$).

DEMOSTRACIÓN:
Ideas que intervienen

- 2) $\Rightarrow$ 1) *por el teorema de Pitagoras:* $(f - g) \perp (g - h)$

$$\|f - h\|^2 = \|f - g\|^2 + \|g - h\|^2 \geq \|f - g\|^2.$$

- 1) $\Rightarrow$ 2) *de la desigualdad:*
  $\|f - g + ah\|^2 = \|f - g\|^2 + a^2\|h\|^2 + 2a < f - g, h > \geq \|f - g\|^2$ *con* $a > 0$, *se sigue que* $< f - g, h > \geq -\frac{a}{2}\|h\|^2$.
  Haciendo $a \to 0$, $< f - g, h > \geq 0$. *Ahora, cambiando* h *por* $-h$, $- < f - g, h > \geq 0$. $\Rightarrow < f - g, h > = 0$.

  g *es la* ***proyección ortogonal*** *de* f *en* G.

□

Cuando conocemos una base de vectores o un sistema generador de G, pedir que $f - g$ sea ortogonal a todos los elementos de G equivale a pedir que sea ortogonal a los vectores que generan todo G, y esto se puede traducir en un sistema de ecuaciones:

2.6.2.2. *Ecuaciones normales*

Definición 2.6.11 *Si $G \subset E$ es un subespacio de dimensión finita, y $\{g_1, \ldots, g_n\}$ es un sistema de vectores que genera G, $g = \sum_{i=1}^{n} x_i g_i$ también viene dada como la solución del sistema de n ecuaciones lineales (compatible determinado, si los vectores forma una base):*

$$\left\{ \sum_{i=1}^{n} x_i < g_i, g_j > = < f, g_j > : \quad j = 1, \ldots, n \right.$$

A estos sistemas de ecuaciones se les llama sistemas de ***ECUACIONES NORMALES****.*

Si además, la base $\{g_1, \ldots, g_n\}$ estuviese formada por vectores 2 a 2 ortogonales de norma 1, la proyección ortogonal de f en G viene dada por la fórmula

$$f = \sum_{i=1}^{n} < f, g_i > g_i.$$

Así, el problema de aproximar por mínimos cuadrados en espacios de dimensión finita G se puede tratar como un sistema de ecuaciones lineales si se conoce una base de G, o simplemente como un problema de cálculo de productos escalares cuando se conoce una base ortonormal en G.

2.6.3. Sistemas sobredeterminados

Considérese la familia de puntos $\{(x_i, y_i)\}_{i=0}^m$ con abscisas distintas dos a dos. La teoría de la interpolación nos dice que existe un único polinomio $q(x)$ de grado a lo sumo m tal que $q(x_i) = y_i$ para cada i. Por tanto, en general, si $n < m$ no existirá para estos puntos un polinomio interpolador de grado menor o igual que n. En estas condiciones tiene sentido, no obstante, preguntarse si existirá algún polinomio $p(x) \in \Pi_n$, el espacio de los polinomios de grado menor o igual que n, que minimice la norma euclídea $\|(y_0 - p(x_0), \ldots, y_m - p(x_m))\|$ o, equivalentemente, la suma $\sum_{i=0}^m (y_i - p(x_i))^2$. La respuesta a esta pregunta es afirmativa pero para formularla adecuadamente necesitamos introducir los llamados *sistemas lineales sobredeterminados*.

Sean $n < m$, $A \in \mathcal{M}_{m\times n}(\mathbb{R})$ y $b \in \mathbb{R}^m$. En lo que sigue supondremos que A tiene rango n. En general el sistema

$$\begin{aligned} a_{11}x_1 + a_{12}x_2 + \cdots + a_{1n}x_n &= b_1, \\ a_{21}x_1 + a_{22}x_2 + \cdots + a_{2n}x_n &= b_2, \\ &\cdots \\ a_{m1}x_1 + a_{m2}x_2 + \cdots + a_{mn}x_n &= b_m, \end{aligned}$$

que en forma matricial puede escribirse como $Ax = b$ para el vector columna $x \in \mathbb{R}^n$ cuyas componentes son las incógnitas $x_1, \ldots, x_n$, carecerá de solución.

Ahora procedemos como sigue. Véase $A : \mathbb{R}^n \to \mathbb{R}^m$ como una aplicación lineal y pongamos $G = A(\mathbb{R}^n)$. El Teorema 2.6.9 nos dice que existe un único $g \in G$ tal que $\|g - b\| < \|y - b\|$ para cada $y \in G$ distinto de g o, equivalentemente, por ser A inyectiva (recordemos que su rango es n), que existe $x \in \mathbb{R}^n$ (el único con la propiedad $Ax = g$) tal que $\|Ax - b\| < \|Au - b\|$ para cada $u \in \mathbb{R}^n$, $x \neq u$.

Más aún, sabemos que g está caracterizado por la propiedad de que $g - b$ es ortogonal a G (Teorema 2.6.10), es decir, $\langle Ax - b, Au\rangle = 0$ para cada $u \in \mathbb{R}^n$. En particular con $u = A^tAx - A^tb$ se tiene,

$$\|A^tAx - A^tb\|_2^2 = \langle A^tAx - A^tb, u\rangle = \langle Ax - b, Au\rangle = 0,$$

y x es solución de

$$A^tAx = A^tb$$

.

Otra forma de hacer este razonamiento es la siguiente:

Sean $\vec{A_j} = (a_{k,j})$ los vectores columna de $A = (a_{i,j})$, y sea $b = (b_k)$, entonces $G = span\{\vec{A_i} : i = 1, ..., n\} \subset \mathbb{R}^m$.
Sea $x = (x_1, x_2, ..., x_n)^t$ un vector columna de $\mathbb{R}^n$.

Para encontrar g plantemos las ecuaciones normales. $g = Ax \in G$ es la MA del vector b en G si se cumplen:

$$\begin{cases} (< \vec{A_1}, g > = \vec{A_1}^t Ax) = (< \vec{A_1}, b > = \vec{A_1}^t b) \\ (< \vec{A_2}, g > = \vec{A_2}^t Ax) = (< \vec{A_2}, b > = \vec{A_2}^t b) \\ \ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots \\ (< \vec{A_n}, g > = \vec{A_n}^t Ax) = (< \vec{A_n}, b > = \vec{A_n}^t b) \end{cases}$$

Donde $\vec{A_2}^t$ son los vectores fila de la matriz traspuesta A^t.

El sistema de ecuaciones normales puede escribirse como:

$$A^t Ax = A^t b$$

Si el rango de A es n, la matriz $A^t A$ es no singular, y la ecuación normal tiene una única solución.

Resumimos nuestras conclusiones en el siguiente teorema:

Teorema 2.6.12 *Sean $n \leq m$, $A \in \mathcal{M}_{m\times n}(\mathbb{R})$ y $b \in \mathbb{R}^m$. Supongamos que A tiene rango n. Entonces existe $u \in \mathbb{R}^n$ tal que $\|Au - b\| < \|Ax - b\|$ para cada $x \in \mathbb{R}^n$, $x \neq u$. El vector u es la única solución de la ecuación normal $A^T Ax = A^T b$.*

(Si $n = m$ u es la única solución del sistema lineal $Ax = b$)

2.6.4. Métodos Numéricos

En la práctica usaremos alguno de los métodos de resolución de sistemas lineales ya conocidos para resolver la ecuación normal. Notemos que cuando $A \in \mathcal{M}_{mn}$ con $m \geq n$ y $rango(A) = n$ ($Ax = 0$ sólo si $x = 0$) la matriz $A^T A \in \mathcal{M}_{nn}$ es siempre simétrica y definida positiva (ya que si $x \neq 0$ entonces $\langle A^T Ax, x\rangle = \langle Ax, Ax\rangle = \|Ax\|^2 > 0$) lo que hace que estemos en un contexto óptimo para usar el método de Cholesky para resolver el sistema de ecuaciones normales:

$$A^t Ax = A^t b.$$

El número de operaciones necesarios para encontrar la aproximación viene dado por

(I) El coste de hacer $A^t A$, que para $A \in \mathcal{M}_m n$ y como la matriz $A^t A$ es simétrica, el de multiplicar cada una de las filas A_i^t de m elementos de A^t por cada una de las i primeras columnas de A, en total $n(n+1)/2$ productos de filas por columnas con $2m-1$ operaciones en cada uno.

$$\text{Coste de } A^t A \longleftarrow \frac{n(n+1)(2m-1)}{2} \approx n(n+1)m \geq n^3.$$

(II) el coste de la factorización de Choleski que es aproximadamente $\frac{n^3}{3}$

(III) el coste de hacer $A^t b$ que es aproximadamente $n(2m-1) \gtrapprox 2n^2$

(IV) el coste de resolver dos sistemas triangulares por los métodos descendente y ascendente que es $2n^2$.

Cuando m es mucho mayor que n, el coste del producto $A^t A$ es el mayor de todos que son marginales frente a este.

Si pensamos en utilizar la factorización QR, $A = QR = Q_1 R_1$ (Teorema 2.5.4). Como $A^t A = (Q_1 R_1)^t (Q_1 R_1) = R_1^t R_1$ Tendremos que el sistema de ecuaciones normales se puede reescribir en la forma

$$R_1^t R_1 x = A^t b.$$

Ahora el coste de resolución es

(I) El coste de encotrar **sólo** R_1 con las transformaciones de Householder (se puede reescribir el algoritmo 2.7 evitando calcular Q_1) que es del orden de $mn(n+1)$ (ver el ejercicio 2.16).

(II) el coste de hacer $A^t b$ que es aproximadamente $n(2m-1) \gtrapprox 2n^2$

(III) el coste de resolver dos sistemas triangulares por los métodos descendente y ascendente que es $2n^2$.

Cuando m es mucho mayor que n, el coste de trabajar con Choleski es de un orden similar al de hacerlo con QR tal y como acabamos de describir con la diferencia de que este último consiste esencialmente en multiplicar por matrices ortogonales que no modifican el condicionamiento de las matrices y puede ser más estable. En las prácticas veremos un ejemplo para comparar las dos opciones.

2.6.5. Aplicaciones y Ejemplos

Estamos ahora en disposición de responder a la cuestión que abría el epígrafe. Recordemos que buscamos el polinomio $p(x) = a_0 + a_1 x + \cdots a_n x^n$ que mejor aproxima en norma euclídea a los puntos $\{(x_i, y_i)\}_{i=0}^m$, es decir, que minimiza la cantidad $\sum_{i=0}^m (y_i - p(x_i))^2$. Si consideramos la matriz

$$A = \begin{pmatrix} 1 & x_0 & x_0^2 & \cdots & x_0^n \\ 1 & x_1 & x_1^2 & \cdots & x_1^n \\ & & \vdots & & \\ 1 & x_m & x_m^2 & \cdots & x_m^n \end{pmatrix}$$

y los vectores $x = (a_0, a_1, \ldots, a_n)$ y $b = (y_0, y_1, \ldots, y_m)$, el problema se reduce a buscar $u = (c_0, c_1, \ldots, c_n)$ tal que $\|Au - b\| < \|Ax - b\|$ para cada $x \neq u$ (nótese que estamos utilizamos la el término x con un doble sentido, como variable del polinomio y como vector de $\mathbb{R}^n$, pero esto no debería inducir a confusión). En otras palabras, el polinomio $p(x) = c_0 + c_1 x + \cdots c_n x^n$ buscado es aquel cuyo vector de coeficientes u es la única solución de la ecuación normal $A^T Ax = A^T b$. Naturalmente la discusión anterior carece de sentido a menos que la matriz A tenga rango $n+1$, pero esto es fácil de probar. Por ejemplo

$$A' = \begin{pmatrix} 1 & x_0 & x_0^2 & \cdots & x_0^n \\ 1 & x_1 & x_1^2 & \cdots & x_1^n \\ & & \vdots & & \\ 1 & x_n & x_n^2 & \cdots & x_n^n \end{pmatrix}$$

es regular porque el sistema

$$\begin{aligned} a_0 + x_0 a_1 + \cdots + x_0^n a_n &= y_0, \\ a_0 + x_1 a_1 + \cdots + x_1^n a_n &= y_1, \\ &\cdots \\ a_0 + x_n a_1 + \cdots + x_n^n a_n &= y_n \end{aligned}$$

tiene solución única (los números $a_0, a_1, \ldots a_n$ son los coeficientes del polinomio interpolador en $\{(x_i, y_i)\}_{i=0}^n$, que como sabemos existe y es único).

Veamos a continuación un ejemplo ilustrativo.

Ejemplo 2.6.13 Consideremos la tabla de puntos

x	1	2	3	4	5	6	7	8	9
y	2.1	3.3	3.9	4.4	4.6	4.8	4.6	4.2	3.4

y busquemos el polinomio $p(x) = a + bx + cx^2$ que mejor la aproxime. El sistema sobredimensionado a considerar es $Ax \approx d$, donde

$$A = \begin{pmatrix} 1 & 1 & 1 \\ 1 & 2 & 4 \\ 1 & 3 & 9 \\ 1 & 4 & 16 \\ 1 & 5 & 25 \\ 1 & 6 & 36 \\ 1 & 7 & 49 \\ 1 & 8 & 64 \\ 1 & 9 & 81 \end{pmatrix},$$

$x = (a, b, c)$ y $d = (2.1, 3.3, 3.9, 4.4, 4.6, 4.8, 4.6, 4.2, 3.4)$. Los coeficientes (a, b, c) buscados serán la solución de la ecuación normal $A^T Ax = A^T d$, donde

$$A^T A = \begin{pmatrix} 9 & 45 & 285 \\ 45 & 285 & 2025 \\ 285 & 2025 & 15333 \end{pmatrix},$$

y $A^T d = (35.3, 186.2, 1178.2)$. Resolviendo la ecuación normal resulta $a = 0.9333$, $b = 1.3511$, $c = -0.1189$.

Con adecuadas modificaciones del método de aproximación por mínimos cuadrados descrito podemos aproximar familias de puntos por funciones no necesariamente polinómicas.

Ejemplo 2.6.14 Consideremos la tabla de puntos

x	1	2	3	4
y	7	11	17	27

y buscamos la función $y = ae^{bx}$ que mejor la aproxime. Tomando logaritmos y escribiendo $c = \log a$ tendríamos

$$\log y = \log a + bx = c + bx,$$

con lo que podemos encontrar c y b aplicando nuestro método a la tabla de datos

x	1	2	3	4
$\log y$	$\log 7$	$\log 11$	$\log 17$	$\log 27$

y una vez obtenidos c y b, calculamos $a = e^c$. Concretamente tendremos

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

$x = (c, b)$ y $d = (\log 7, \log 11, \log 17, \log 27)$. Entonces

$$A^T A = \begin{pmatrix} 4 & 10 \\ 10 & 30 \end{pmatrix}$$

y

$$A^T d = (\log 35343, \log 2211491279151) = (10.4729\ldots, 28.4227\ldots),$$

obteniéndose $c = 1.497$ y $b = 0.485$ como solución de la ecuación normal $A^T Ax = A^T d$. Finalmente $a = e^c = 4.468$.

Es importante resaltar que en el anterior ejemplo *no* estamos afirmando que la función $f(x) = 4.468e^{0.485x}$ sea aquella entre las de la forma $f(x) = ae^{bx}$ que minimiza $(f(1) - 7)^2 + (f(2) - 11)^2 + (f(3) - 17)^2 + (f(4) - 27)^2$. El último ejemplo de la sección ilustra con más énfasis esta cuestión.

Ejemplo 2.6.15 Se supone que el cometa Tentax, descubierto el año 1968, es un objeto del Sistema Solar. En cierto sistema de coordenadas polares (r, φ), centrado en el Sol, se han medido experimentalmente las siguientes posiciones del cometa:

r	2.20	2.00	1.61	1.20	1.02
φ	48º	67º	83º	108º	126º

Si se desprecian las perturbaciones de los planetas, las leyes de Kepler garantizan que el cometa se moverá en una órbita elíptica, parabólica o hiperbólica, que en dichas coordenadas polares tendrá en cualquier caso la ecuación

$$r = \frac{p}{1 + e\cos\varphi},$$

donde p es un parámetro y e la excentricidad. Ajustemos por mínimos cuadrados los valores p y e a partir de las medidas hechas.

A partir de los datos dados hay varias maneras de formular un problema de mínimos cuadrados, todas válidas pero no todas equivalentes entre sí. Mostramos a continuación dos posibles formas de formular el problema, que dan dos aproximaciones diferentes ya que son soluciones de dos problemas distintos:

Posibilidad 1. Despejando en la ecuación

$$r = \frac{p}{1 + e\cos\varphi}$$

llegamos a

$$r + er\cos\varphi = p$$

y de aquí

$$r = p + e(-r\cos\varphi).$$

Por tanto se trata de minimizar $\|Ax - b\|$, donde

$$A = \begin{pmatrix} 1 & -2.20\ \cos 48 \\ 1 & -2.00\ \cos 67 \\ 1 & -1.61\ \cos 83 \\ 1 & -1.20\ \cos 108 \\ 1 & -1.02\ \cos 126 \end{pmatrix} = \begin{pmatrix} 1 & -1.47209 \\ 1 & -0.78146 \\ 1 & -0.19621 \\ 1 & 0.37082 \\ 1 & 0.59954 \end{pmatrix},$$

$$b = \begin{pmatrix} 2.20 \\ 2.00 \\ 1.61 \\ 1.20 \\ 1.02 \end{pmatrix}$$

y

$$x = \begin{pmatrix} p \\ e \end{pmatrix}.$$

Dado que A tiene rango 2 tiene sentido plantear la ecuación normal $A^T Ax = A^T b$, es decir,

$$\begin{pmatrix} 5 & -1.47940 \\ -1.47940 & 3.31318 \end{pmatrix} \begin{pmatrix} p \\ e \end{pmatrix} = \begin{pmatrix} 8.03 \\ -4.06090 \end{pmatrix}$$

cuya solución $p = 1.43262$ y $e = -0.58599$ nos da los valores de p y e buscados (obteniéndose $\|Ax - b\| = 0.22686$ como medida de la aproximación).

Posibilidad 2. Dividiendo en

$$r + er\cos\varphi = p$$

por er y despejando se obtiene

$$\cos\varphi = (-1/e) + (p/e)(1/r).$$

Se trata ahora de minimizar $\|Ax - b\|$, donde

$$A = \begin{pmatrix} 1 & 1/2.20 \\ 1 & 1/2 \\ 1 & 1/1.61 \\ 1 & 1/1.20 \\ 1 & 1/1.02 \end{pmatrix} = \begin{pmatrix} 1 & 0.45454 \\ 1 & 0.5 \\ 1 & 0.62112 \\ 1 & 0.83333 \\ 1 & 0.98039 \end{pmatrix},$$

$$b = \begin{pmatrix} \cos 48 \\ \cos 67 \\ \cos 83 \\ \cos 108 \\ \cos 126 \end{pmatrix} = \begin{pmatrix} 0.66913 \\ 0.39073 \\ 0.12187 \\ -0.30902 \\ -0.58779 \end{pmatrix}$$

y

$$x = \begin{pmatrix} c \\ d \end{pmatrix}$$

con $c = -1/e$ y $d = p/e$. A partir de la ecuación normal $A^T Ax = A^T b$, es decir,

$$\begin{pmatrix} 5 & 3.38939 \\ 3.38939 & 2.49801 \end{pmatrix} \begin{pmatrix} c \\ d \end{pmatrix} = \begin{pmatrix} 0.28493 \\ -0.25856 \end{pmatrix},$$

obtenemos $c = 1.58479$ y $d = -2.25381$, de donde $p = 1.42215$ y $e = -0.63100$ (obteniéndose ahora $\|Ax - b\| = 0.29359$ como medida de la aproximación).

2.7. Actividades complementarias del capítulo

Los documentos se descargan de la Zona de Recursos en el Aula Virtual.

- Hoja de problemas nº2
- Prácticas 2, 3 y 4.

Bibliografía

[1] A. Delshams A. Aubanell, A. Benseny, *Útiles básicos de cálculo numérico*, Ed. Labor - Univ. Aut. de Barcelona, Barcelona, 1993.

[2] G. Allaire and S.M. Kaber, *Numerical linear algebra*, TAM 55, Springer,, New York, etc., 2008.

[3] C. Brézinski, *Introducton á la pratique du calcul numérique*, Dunod Université, Paris, 1988.

[4] R.L. Burden and J.D. Faires, *Análisis numérico, 7ª edición*, Thomson-Learning, Mexico, 2002.

[5] P.G. Ciarlet, *Introduction à l'analyse numérique matricielle et à l'optimisation*, Masson, Paris, 1990.

[6] P.M. Cohn, *Algebra, vol. 1 y 2*, Ed. Hohn Wiley and Sons, London, 1974.

[7] G. Hammerlin and K.H. Hoffmann, *Numerical mathematics*, Springer-Verlag, New York, 1991.

[8] J.Stoer and R. Burlisch, *Introduction to numerical analysis*, Springer Verlag, New York, 1980.

[9] D. Kincaid and W. Cheney, *Análisis numérico*, Ed. Addison-Wesley Iberoamericana, Reading, USA, 1994.

[10] J. R. Shewchuk, *An introduction to the conjugate gradient method without the agonizing pain*, Tech. report, Carnegie Mellon University, USA,http://www.cs.cmu.edu/~quake-papers/painless-conjugate-gradient-figs.pdf,doi=10.1.1.110.418, 1994.