

PRACTICA 3c

Cálculo numérico de autovalores y autovectores.

Aplicación del teorema de Pitágoras: Puesto que, por el método QR, los autovalores de una matriz de orden n se encuentran sobre la diagonal, el número de autovalores es n veces la raíz cuadrada de 2.

Muchos problemas de interés conducen al cálculo, o por lo menos a la estimación, de los autovalores de una matriz asociada con un sistema lineal de ecuaciones. Algebraicamente el problema consiste en, dada una matriz A de $n \times n$ elementos reales, encontrar los escalares (generalmente complejos), λ , los *autovalores* de A , y los vectores no nulos $\mathbf{x}$, los *autovectores* de A asociados a λ , tales que

$$A\mathbf{x} = \lambda\mathbf{x}.$$

Sabemos que tal matriz A tiene precisamente n autovalores, no necesariamente distintos, que son las raíces del *polinomio característico* de grado n :

$$p(\lambda) = \det(A - \lambda I).$$

Además, si A es una matriz real *simétrica*, entonces todos sus autovalores son reales. En el caso general, si A tiene elementos reales, sus posibles autovalores complejos aparecen en pares conjugados. Formalmente, los autovalores de A se pueden obtener encontrando las n raíces de $p(\lambda)$. Sin embargo, en la práctica, esto puede llevarse a cabo sólo con matrices de pequeño tamaño o de formas particulares. En general, sabemos que la determinación de las raíces de un polinomio es un problema difícil, puesto que, excepto para valores pequeños de n , no es un problema *cerrado* (esto es, no hay fórmulas explícitas para las raíces) y es además un problema *mal condicionado* en presencia de raíces múltiples. En consecuencia es necesario considerar algoritmos numéricos que permitan la determinación de los autovalores de una matriz en forma eficiente y sean numéricamente estables.

Teorema de Gerschgorin. De acuerdo al teorema de Gerschgorin, los autovalores de una matriz A de $n \times n$ están contenidos en la unión de los círculos del plano complejo $\mathbb{C}$ dados por

$$|z - a_{ii}| \leq r_i, \quad \text{donde } r_i = \sum_{\substack{j=1 \\ j \neq i}}^n |a_{ij}|,$$

para $i = 1, 2, \dots, n$. Además, la unión de cualesquiera k de estos círculos que no intersecten a los $(n - k)$ restantes, debe contener precisamente k (contando multiplicidades) autovalores.

Ejercicio 1. Utilizar el teorema de Gerschgorin para localizar los autovalores de las siguientes matrices y determinar una cota del *radio espectral* $\rho(A) = \max_{1 \leq i \leq n} \{|\lambda_i|\}$,

$$\begin{pmatrix} 4 & 1 & 1 \\ 0 & 2 & 1 \\ -2 & 0 & 9 \end{pmatrix}, \begin{pmatrix} 5 & 1 & 1 \\ 0 & 6 & 1 \\ 1 & 0 & -5 \end{pmatrix}, \begin{pmatrix} 3 & 2 & 1 \\ 2 & 3 & 0 \\ 1 & 0 & 3 \end{pmatrix}.$$

Método de potencia. Si A es una matriz $n \times n$ diagonalizable y tiene un autovalor dominante λ_1 , esto es,

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

sabemos que la sucesión de vectores $\{\mathbf{x}^{(k)}\}_{k=0}^{\infty}$, generada a partir de un vector inicial no nulo cualquiera $\mathbf{x}^{(0)}$ (en tanto su representación en términos de los autovectores de la matriz contenga una contribución no nula del autovector asociado al autovalor dominante), según

$$\mathbf{x}^{(k)} = A\mathbf{x}^{(k-1)},$$

tiende a la *dirección* del autovector correspondiente a λ_1 . Escalando la sucesión de vectores en forma apropiada se puede lograr que el límite sea un vector finito y no nulo y obtener así el autovalor λ_1 en el proceso. La descripción algorítmica del procedimiento es como sigue.

Método de potencia (norma l_{∞})

Dada A de $n \times n$ y un vector no nulo $\mathbf{x}^{(0)}$. Encontrar p ($1 \leq p \leq n$) tal que $|x_p^{(0)}| = \|\mathbf{x}^{(0)}\|_{\infty}$.

Tomar $\mathbf{x}^{(0)} = \mathbf{x}^{(0)} / x_p^{(0)}$.

Para $k = 0, 1, 2, \dots$

Tomar $\mathbf{y}^{(k+1)} = A\mathbf{x}^{(k)}$.

Tomar $\sigma_{k+1} = y_p^{(k+1)}$.

Encontrar p ($1 \leq p \leq n$) tal que $|y_p^{(k+1)}| = \|\mathbf{y}^{(k+1)}\|_{\infty}$.

Tomar $\mathbf{x}^{(k+1)} = \mathbf{y}^{(k+1)} / y_p^{(k+1)}$.

Entonces la sucesión de valores $\{\sigma_k\}$ tiende al autovalor dominante λ_1 y la sucesión de vectores $\{\mathbf{x}^{(k)}\}$ se aproxima al subespacio propio correspondiente a dicho autovalor dominante. Esto último quiere decir que dado cualquier número real $\epsilon > 0$ podemos encontrar un entero positivo k , suficientemente grande, y un vector propio $\mathbf{v}_k$ asociado al autovalor dominante, tales que $\|\mathbf{x}^{(k)} - \mathbf{v}_k\| < \epsilon$. Como en todo método iterativo, debe establecerse un criterio de paro para las iteraciones, el cual, en este contexto, puede ser:

$$|\sigma_k - \sigma_{k-1}| / |\sigma_k| < \epsilon$$

para cierta tolerancia ϵ prefijada.

El método de potencias es implementado en el módulo `power_method`, presentado en el código 1, por la subrutina `power_li`. (Nótese el uso de la función intrínseca `maxloc` para determinar el índice p).

Ejercicio 2. Utilizar la implementación del método de potencia para estimar el autovalor de mayor magnitud de las matrices del ejercicio 1.

Método de factorización QR. Los métodos que proveen aproximaciones para todos los autovalores de una matriz basan su procedimiento en la generación de una sucesión de matrices *semejantes* unas a otras de manera tal que se obtengan matrices que permitan determinar los autovalores de la forma más simple posible. Recordemos que B es *semejante* a A si existe una matriz invertible P tal que $B = P^{-1}AP$. Entonces, si $B\mathbf{y} = \lambda\mathbf{y}$, se sigue que $P^{-1}AP\mathbf{y} = \lambda\mathbf{y}$, y por lo tanto, $A(P\mathbf{y}) = \lambda(P\mathbf{y})$. Esto es, A y B tienen los mismos autovalores y, si $\mathbf{y}$ es un autovector de B , $\mathbf{x} = P\mathbf{y}$ es un autovector de A . Así, una transformación de semejanza preserva los autovalores y, aunque no preserva los autovectores, éstos pueden ser fácilmente determinados. Dentro de las transformaciones de semejanza está incluida un tipo de transformación más restrictivo: la transformación *ortogonal*, donde la inversa de P es su transpuesta, $P^{-1} = P^T$. Una transformación ortogonal no sólo preserva los autovalores sino también la simetría, si ésta existe, y la norma 2 de los autovectores. En efecto, si A es una matriz simétrica y B es ortogonalmente semejante a A , entonces $B^T = (P^TAP)^T = P^T A^T P = P^T A T = B$, es decir, B es simétrica. Finalmente, si $\mathbf{y}$ es un autovector de B , $\mathbf{x} = P\mathbf{y}$ es un autovector de A y $\|\mathbf{x}\|_2^2 = \mathbf{x}^T \mathbf{x} = (P\mathbf{y})^T (P\mathbf{y}) = \mathbf{y}^T (P^T P) \mathbf{y} = \mathbf{y}^T \mathbf{y} = \|\mathbf{y}\|_2^2$.

Los *métodos de factorización* para el cálculo de todos los autovalores de A parten de la suposición de que la matriz A puede ser factorizada en una matriz “a izquierda” F_L y una matriz “a derecha” F_R , esto es, $A = F_L F_R$. Si ahora invertimos los factores obtenemos la matriz $A' = F_R F_L = F_L^{-1} A F_L$, la cual es semejante a A a través de la transformación F_L . Así, cualquier factorización de A al ser multiplicada en orden inverso, constituye una transformación de semejanza. Los métodos de factorización explotan este idea, pero para que el algoritmo sea práctico, la factorización debe ser tal que el método converja, sea numéricamente estable y computacionalmente eficiente. Uno de los mejores métodos de factorización conocidos es el *método QR*.

El método QR genera una sucesión de matrices $A_0, A_1, A_2, \dots$ comenzando con $A_0 = A$ y calculando en el paso k -ésimo la *factorización QR* de A_k , es decir, genera matrices Q_k y R_k tales que $A_k = Q_k R_k$ donde Q_k es una matriz ortogonal ($Q_k^{-1} = Q_k^T$) y R_k es una matriz triangular superior. Una factorización de este tipo siempre existe y es posible de ser calculada en forma numéricamente estable (los tres métodos usuales son a través de *transformaciones de Householder*, o *transformaciones de Givens* u *ortogonalización de Gram-Schmidt*, siendo el primer método el más efectivo tanto en costo computacional como en almacenamiento). A continuación el método toma como siguiente matriz de la sucesión $A_{k+1} = R_k Q_k$. La transformación que lleva A_k a A_{k+1} no sólo es de semejanza, sino que también es ortogonal. En efecto, ya que Q_k es ortogonal, $R_k = Q_k^T A_k$ y por lo tanto $A_{k+1} = Q_k^T A_k Q_k$. Puede mostrarse que, bajo

ciertas circunstancias generales (en particular, si los autovalores de A tienen todos sus módulos distintos), entonces la sucesión de matrices A_k converge a una matriz triangular superior para una matriz inicial general, y a una matriz diagonal si la matriz inicial es simétrica. En ambos casos, las aproximaciones a los autovalores se encuentran en los elementos de la diagonal principal de las matrices generadas.

En los siguientes ejercicios consideraremos una implementación naïve del método QR, pero que, sin embargo, permite ilustrar las ideas descritas. Para ello necesitamos implementar en primer lugar la factorización QR de una matriz. Por simplicidad, consideraremos aquí el método (modificado) de ortogonalización de Gram-Schmidt. Aunque en nuestro problema A es una matriz cuadrada, consideraremos la situación general de una matriz rectangular puesto que la factorización obtenida resulta también de gran utilidad para el problema de mínimos cuadrados que discutiremos en una práctica posterior. Sea, entonces, A una matriz de $m \times n$ con $m \geq n$. Si las columnas de A son linealmente independientes, esto es, si el *rango* de A es n , entonces existe una matriz Q ortogonal de $m \times n$ y una matriz triangular superior R de $n \times n$, con elementos positivos sobre su diagonal, tal que $A = QR$. El método (modificado) de ortogonalización de Gram-Schmidt permite obtener una representación explícita de Q y R ortonormalizando las columnas de A como sigue. Sea $\mathbf{a}_j$ para $j = 1, 2, \dots, n$ las columnas de A , esto es, $A = [\mathbf{a}_1 | \mathbf{a}_2 | \dots | \mathbf{a}_n]$. Sea $\langle \mathbf{v} | \mathbf{u} \rangle = \mathbf{v}^T \mathbf{u}$ el producto interno canónico en $\mathbb{R}^m$ y $\|\mathbf{v}\| = \sqrt{\langle \mathbf{v} | \mathbf{v} \rangle}$ la norma asociada. Entonces la implementación algorítmica del método es como sigue:

Dadas $\mathbf{a}_1, \mathbf{a}_2, \dots, \mathbf{a}_n$

Para $j = 1, 2, \dots, n$

Tomar $\mathbf{q}_j = \mathbf{a}_j$

Para $i = 1, 2, \dots, j-1$

Calcular $r_{ij} = \langle \mathbf{q}_i | \mathbf{q}_j \rangle$

Tomar $\mathbf{q}_j = \mathbf{q}_j - r_{ij} \mathbf{q}_i$

Calcular $r_{jj} = \|\mathbf{q}_j\|$

Tomar $\mathbf{q}_j = \mathbf{q}_j / r_{jj}$

De este modo, $Q = [\mathbf{q}_1 | \mathbf{q}_2 | \dots | \mathbf{q}_n]$ y $R = (r_{ij})$.

Ejercicio 3. Implementar la factorización QR de una matriz A de $m \times n$ como una subrutina Fortran (Nótese que Q puede ser devuelta en A).

Ejercicio 4. Utilizando la subrutina anterior implementar un programa Fortran para calcular los autovalores de una matriz por el método QR. En particular, obtener los autovalores y visualizar la sucesión de matrices obtenidas por el método QR para la matriz

$$\begin{pmatrix} 7 & 2 \\ 2 & 4 \end{pmatrix}$$

cuyos autovalores exactos son 8 y 3.

Para una matriz A densa, el costo computacional por iteración del método QR es $\mathcal{O}(n^3)$. Este costo puede ser reducido si la matriz es inicialmente transformada, vía semejanza, a una forma más simple. En general, toda matriz puede ser reducida por transformaciones de semejanza ortogonales en un número finito de pasos a una matriz de Hessenberg superior. Una matriz de Hessenberg superior tiene todos sus elementos nulos debajo de la diagonal inferior, esto es, es próxima a una matriz triangular superior excepto por la diagonal adicional de elementos no nulos inmediatamente debajo de la diagonal principal. En el caso particular de una matriz simétrica ésta es reducida a una matriz tridiagonal¹. El costo computacional en la obtención de la forma reducida es $\mathcal{O}(n^3)$, pero ahora, puesto que puede mostrarse que el método QR preserva la forma de la matriz de Hessenberg superior o la tridiagonal en sus iteraciones, el costo computacional por iteración del método QR es $\mathcal{O}(n^2)$ en el caso general y $\mathcal{O}(n)$, en el caso simétrico. Así pues, el método QR es implementado como un proceso de dos etapas:

Matriz general $\longrightarrow$ Hessenberg $\longrightarrow$ triangular.
Matriz simétrica $\longrightarrow$ tridiagonal $\longrightarrow$ diagonal.

Implementaciones eficientes del método QR están disponibles en LAPACK95 en las subrutinas `la_syev` y `la_geev` para matrices simétricas y generales, respectivamente.

Ejercicio 5. Utilizando la subrutina `la_syev` encontrar aproximaciones para los autovalores de las siguientes matrices simétricas.

$$\begin{pmatrix} 12 & 10 & 4 \\ 10 & 8 & -5 \\ 4 & -5 & 3 \end{pmatrix}, \quad \begin{pmatrix} 2 & 0 & 1 \\ 0 & 3 & -2 \\ 1 & -2 & -1 \end{pmatrix},$$

$$\begin{pmatrix} 4 & -1 & -1 & 0 \\ -1 & 4 & 0 & -1 \\ -1 & 0 & 4 & -1 \\ 0 & -1 & -1 & 4 \end{pmatrix}, \quad \begin{pmatrix} 4 & 2 & 0 & 0 & 0 \\ 2 & 4 & 2 & 0 & 0 \\ 0 & 2 & 4 & 2 & 0 \\ 0 & 0 & 2 & 4 & 2 \\ 0 & 0 & 0 & 2 & 4 \end{pmatrix}.$$

Ejercicio 6. Utilizando la subrutina `la_geev` encontrar aproximaciones a los autovalores de las siguientes matrices.

$$\begin{pmatrix} 5 & 2 & 0 \\ 1 & 4 & -1 \\ 0 & 3 & 7 \end{pmatrix}, \quad \begin{pmatrix} 6 & -2 & 0 & 0 \\ 3 & 4 & -1 & 0 \\ 0 & 0 & 5 & -1 \\ 0 & 0 & 2 & 3 \end{pmatrix}.$$

Aplicación a las ecuaciones algebraicas. Los métodos numéricos para calcular autovalores no involucran al polinomio característico, pues, como hemos indicado, la determinación de sus ceros es en general un problema mal condicionado. De hecho, podemos, a la inversa, utilizar los métodos numéricos del cálculo de autovalores para obtener las raíces de una ecuación algebraica de la forma $x^n + a_{n-1}x^{n-1} + \dots + a_1x + a_0 = 0$ notando que ellos son los ceros del polinomio característico de la siguiente matriz acompañante de $n \times n$:

$$\begin{pmatrix} 0 & 0 & \dots & 0 & -a_0 \\ 1 & 0 & \dots & 0 & -a_1 \\ 0 & 1 & \dots & 0 & -a_2 \\ \vdots & \vdots & \ddots & \vdots & \vdots \\ 0 & 0 & \dots & 1 & -a_{n-1} \end{pmatrix}.$$

Ejercicio 7. Determinar las raíces de la ecuación $2x^5 - 12x^4 + 24x^3 - 24x^2 + 22x - 12 = 0$ utilizando el procedimiento descrito.

¹Nótese que una matriz de Hessenberg simétrica es tridiagonal.

Código 1. Implementación del método de potencia

```

module power_method

  use iso_fortran_env, only: wp => real64
  implicit none

contains

  subroutine power_li(a,x,tol,iter,lambda,clave)

    ! Calculo del autovalor dominante y su respectivo autovector
    ! por el método de la potencia (versión norma infinito)
    !
    ! Argumentos de la subrutina:
    !
    ! Matriz del problema
    real(wp), intent(in) :: a(:, :)
    ! Vector inicial (entrada) / Estimación del autovector (salida)
    real(wp), intent(inout) :: x(:)
    ! Tolerancia prefijada para el criterio de paro
    real(wp), intent(in) :: tol
    ! Máximo de iteraciones (entrada) / iteraciones realizadas (salida)
    integer, intent(inout) :: iter
    ! Estimación del autovalor dominante
    real(wp), intent(out) :: lambda
    ! Clave de error: 0 ok, /= 0 iteraciones máximas alcanzado
    integer, intent(out) :: clave

    ! Variables locales
    integer :: i
    integer :: p
    real(wp) :: lambda0
    real(wp) :: y(size(x))

    ! Procedimiento

    clave = 1
    lambda0 = 0.0_wp
    p = maxloc(abs(x),1)
    x = x/x(p)

    do i = 1, iter
      y = matmul(a,x)
      lambda = y(p)
      p = maxloc(abs(y),1)
      x = y/y(p)
      if ( abs(lambda-lambda0) <= tol*abs(lambda) ) then
        clave = 0
        iter = i
        exit
      end if
      lambda0 = lambda
    end do

  end subroutine power_li
end module power_method

```