Cómputo Científico **Omar García Ramos**
Tarea IV

Ejercicio 1

Sea $A = (a_{ij}) \in \mathbb{C}^{n\times n}$. Definimos para cada fila i el disco de Gershgorin

$$D_i = \left\{ z \in \mathbb{C} : |z - a_{ii}| \leq \sum_{j\neq i} |a_{ij}| \right\}.$$

El **Teorema de Gershgorin** establece que *todos los eigenvalores de A pertenecen a la unión de los discos $D_1, \ldots, D_n$*. Además, si un conjunto de k discos es disjunto de los demás, entonces dicho conjunto contiene exactamente k eigenvalores de A.

Consideremos la matriz

$$A(\varepsilon) = \begin{pmatrix} 8 & 1 & 0 \\ 1 & 4 & \varepsilon \\ 0 & \varepsilon & 1 \end{pmatrix}, \qquad |\varepsilon| \leq 1.$$

Los discos de Gershgorin correspondientes son:

- D_1: centro en 8, radio 1,
- D_2: centro en 4, radio $1 + |\varepsilon|$,
- D_3: centro en 1, radio $|\varepsilon|$.

Por el teorema, todos los eigenvalores de $A(\varepsilon)$ deben encontrarse dentro de la unión de estos tres discos.

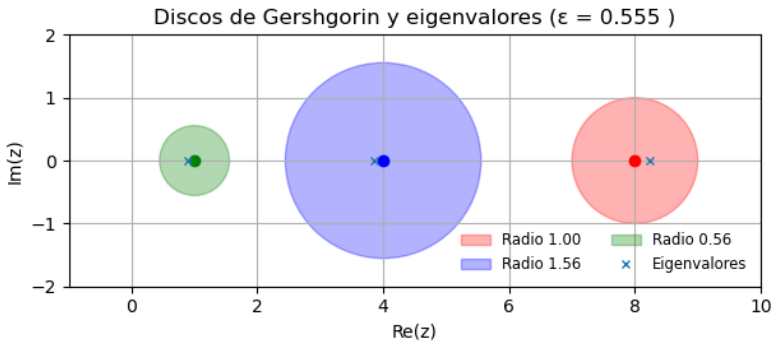

Figura 1: Discos de Gershgorin para $\varepsilon = 0{,}555$, junto con los eigenvalores exactos de la matriz.

En la Figura 1 observamos que, conforme ε aumenta, los discos D_2 y D_3 pueden llegar a solaparse. Cuando $|\varepsilon| = 1$, los radios de estos discos alcanzan su máximo valor, y D_2 se superpone parcialmente con D_3. Aun así, el disco D_1 permanece separado, garantizando la presencia de un eigenvalor cercano a 8. Los discos D_2 y D_3, al formar un dominio conexo, contienen en conjunto exactamente dos eigenvalores. En total, se recupera así la localización de los tres eigenvalores de la matriz.

Ejercicio 2

El método QR consiste en calcular iterativamente una secuencia de matrices A_k similares a la matriz inicial $A_0 = A$, cuyas diagonales convergen a los eigenvalores de A.

En cada paso k de la iteración se realiza la factorización QR:

$$A_{k-1} = Q_k R_k,$$

donde Q_k es ortogonal y R_k es triangular superior.

A continuación, se construye la siguiente matriz de la secuencia como

$$A_k = R_k Q_k.$$

Obsérvese que

$$A_k = R_k Q_k = Q_k^\top A_{k-1} Q_k,$$

de modo que A_k es similar a A_{k-1} y, por lo tanto, todas las matrices A_k tienen los mismos eigenvalores que A.

En nuestra implementación, la factorización QR se realiza de dos maneras:

1. Usando *Gram–Schmidt modificado* (nuestro propio algoritmo).
2. Usando la función `scipy.linalg.qr` de `SciPy`, como referencia.

En ambos casos, los eigenvalores aproximados se obtienen a partir de las entradas diagonales de A_k cuando el término fuera de la diagonal cumple

$$\sum_{i \neq j} |(A_k)_{ij}| < \text{tol},$$

donde `tol` es la tolerancia establecida.

Se aplicó el método QR a la matriz

$$A = \begin{bmatrix} 8 & 1 & 0 \\ 1 & 4 & \varepsilon \\ 0 & \varepsilon & 1 \end{bmatrix}, \qquad \varepsilon \in \{10^0, 10^{-1}, 10^{-2}, 10^{-4}, 10^{-5}\}.$$

Para ilustrar cómo la iteración QR transforma la matriz en cada paso, mostramos el resultado de las primeras 5 iteraciones para el caso $\varepsilon = 10^{-1}$.

Iteración con Gram-Schmidt (propio)

$$A^{(0)} = \begin{bmatrix} 8{,}1846 & 0{,}4771 & 5{,}03 \cdot 10^{-17} \\ 0{,}4771 & 3{,}8186 & 2{,}59 \cdot 10^{-2} \\ 0 & 2{,}59 \cdot 10^{-2} & 0{,}9967 \end{bmatrix},$$

$$A^{(1)} = \begin{bmatrix} 8{,}2253 & 0{,}2202 & 4{,}84 \cdot 10^{-17} \\ 0{,}2202 & 3{,}7782 & 6{,}83 \cdot 10^{-3} \\ 0 & 6{,}83 \cdot 10^{-3} & 0{,}9965 \end{bmatrix},$$

$$A^{(2)} = \begin{bmatrix} 8{,}2339 & 0{,}1009 & 5{,}02 \cdot 10^{-17} \\ 0{,}1009 & 3{,}7696 & 1{,}80 \cdot 10^{-3} \\ 0 & 1{,}80 \cdot 10^{-3} & 0{,}9965 \end{bmatrix}.$$

Iteración con SciPy QR

$$A^{(0)} = \begin{bmatrix} 8{,}1846 & 0{,}4771 & -5{,}20 \cdot 10^{-18} \\ 0{,}4771 & 3{,}8186 & -2{,}59 \cdot 10^{-2} \\ 0 & -2{,}59 \cdot 10^{-2} & 0{,}9967 \end{bmatrix},$$

$$A^{(1)} = \begin{bmatrix} 8{,}2253 & 0{,}2202 & 5{,}64 \cdot 10^{-18} \\ 0{,}2202 & 3{,}7782 & 6{,}83 \cdot 10^{-3} \\ 0 & 6{,}83 \cdot 10^{-3} & 0{,}9965 \end{bmatrix},$$

$$A^{(2)} = \begin{bmatrix} 8{,}2339 & 0{,}1009 & -5{,}58 \cdot 10^{-18} \\ 0{,}1009 & 3{,}7696 & -1{,}80 \cdot 10^{-3} \\ 0 & -1{,}80 \cdot 10^{-3} & 0{,}9965 \end{bmatrix}.$$

Se observa que en ambas variantes, la matriz $A^{(k)}$ tiende rápidamente a una forma casi triangular, con los eigenvalores apareciendo en la diagonal, no obstante podemos notar que la iteración con Scipy ofrece ya convergencias mayores a la del algoritmo propio.

Los resultados obtenidos se muestran en el Cuadro 1.

ε	Eigenvalores teóricos	Gram-Schmidt	Scipy QR	Error Rel GS	Error Rel Sp	Iter GS	Iter Sp
$1e{+}0$	[8.2436, 4.0711, 0.6853]	[8.2436, 4.0711, 0.6853]	[8.2436, 4.0711, 0.6853]	$4{,}31 \cdot 10^{-16}$	$7{,}52 \cdot 10^{-16}$	1500	1500
$1e{-}1$	[8.2361, 3.7674, 0.9965]	[8.2361, 3.7674, 0.9965]	[8.2361, 3.7674, 0.9965]	$6{,}34 \cdot 10^{-16}$	$9{,}20 \cdot 10^{-16}$	1500	1500
$1e{-}2$	[8.2361, 3.7640, 1.0000]	[8.2361, 3.7640, 1.0000]	[8.2361, 3.7640, 1.0000]	$2{,}92 \cdot 10^{-16}$	$6{,}17 \cdot 10^{-16}$	1500	1500
$1e{-}4$	[8.2361, 3.7639, 1.0000]	[8.2361, 3.7639, 1.0000]	[8.2361, 3.7639, 1.0000]	$5{,}02 \cdot 10^{-17}$	$3{,}93 \cdot 10^{-16}$	1500	48
$1e{-}5$	[8.2361, 3.7639, 1.0000]	[8.2361, 3.7639, 1.0000]	[8.2361, 3.7639, 1.0000]	$4{,}17 \cdot 10^{-16}$	$6{,}17 \cdot 10^{-16}$	1500	1500

Cuadro 1: Comparación entre la iteración QR con los algoritmos Gram-Schmidt y con SciPy, con una tolerancia de 1×10^{-16} y un máximo de 1,500 iteracciones.

A partir de los resultados podemos concluir que ambos métodos (Gram–Schmidt y SciPy) obtienen los eigenvalores correctos con errores relativos muy pequeños del orden de 10^{-16}, la diferencia radica en la velocidad de convergencia, pues nuestra implementación con Gram–Schmidt requiere prácticamente siempre el máximo de iteraciones permitidas (1500), mientras que la factorización QR de SciPy puede converger mucho más rápido, como ocurre en el caso $\varepsilon = 10^{-4}$ (48 iteraciones),esto muestra que, aunque la implementación propia es teóricamente correcta, la versión optimizada y numéricamente estable de SciPy es mucho más eficiente en la práctica.