

UNIVERSIDAD NACIONAL DEL ALTIPLANO

ESCUELA DE POSGRADO

DOCTORADO EN CIENCIAS DE LA COMPUTACIÓN



TESIS

**ALGORITMO BASADO EN EL MÉTODO QR PARA ANALIZAR LA
SENSIBILIDAD DE AUTOVALORES Y AUTOVECTORES REALES DE
MATRICES REALES BIEN CONDICIONADAS**

PRESENTADA POR:

EDGAR RAMOS FLORES

PARA OPTAR EL GRADO ACADÉMICO DE:

DOCTOR EN CIENCIAS DE LA COMPUTACION

PUNO, PERÚ

2019

UNIVERSIDAD NACIONAL DEL ALTIPLANO
ESCUELA DE POSGRADO
DOCTORADO EN CIENCIAS DE LA COMPUTACIÓN



TESIS

**ALGORITMO BASADO EN EL MÉTODO QR PARA ANALIZAR LA
SENSIBILIDAD DE LOS AUTOVALORES Y AUTOVECTORES REALES
DE MATRICES REALES BIEN CONDICIONADAS**

PRESENTADA POR:

EDGAR RAMOS FLORES

**PARA OPTAR EL GRADO ACADÉMICO DE:
DOCTOR EN CIENCIAS DE LA COMPUTACIÓN**

APROBADA POR EL JURADO SIGUIENTE:

PRESIDENTE



.....
Dra. GUINA GUADALUPE SOTOMAYOR ALZAMORA

PRIMER MIEMBRO



.....
Dr. ELVIS AUGUSTO ALIAGA PAYEHUANCA

SEGUNDO MIEMBRO



.....
Dr. JUAN CARLOS JUAREZ VARGAS

ASESOR DE TESIS



.....
Dr. VLADIMIRO IBÁÑEZ QUISPE

Puno, 29 de noviembre del 2019

ÁREA: Ciencias de la computación

TEMA: Algoritmos basados de sensibilidad de autovalores y autovectores

LÍNEA: Ciencia e informática en la ingeniería

DEDICATORIA

A mis queridos padres Felipe y Clementina, a mi
hermana María y a mis sobrinas Milesa y Sandra.

AGRADECIMIENTOS

A la Universidad Nacional del Altiplano por la formación tanto a nivel de pregrado y posgrado que he recibido.

Agradezco a los docentes del doctorado en Ciencias de la Computación por impartir los conocimientos en el área computacional. Además, agradezco a mi Asesor de Tesis Dr. Vladimiro Ibáñez Quispe por haber conducido la presente tesis.

A los miembros del Jurado Dra. Guina Guadalupe Sotomayor Alzamora, Dr. Elvis Augusto Aliaga Payehuanca, Dr. Juan Carlos Juarez Vargas; por todas sus sugerencias y recomendaciones para mejorar la presente investigación.

ÍNDICE GENERAL

| | Pág. |
|-------------------------|-------------|
| DEDICATORIA | i |
| AGRADECIMIENTOS | ii |
| ÍNDICE GENERAL | iii |
| ÍNDICE DE TABLAS | vii |
| ÍNDICE DE FIGURAS | viii |
| ÍNDICE DE ANEXOS | ix |
| RESUMEN | x |
| ABSTRACT | xi |
| INTRODUCCIÓN | 1 |

CAPÍTULO I

REVISIÓN DE LITERATURA

| | |
|---|----|
| 1.1. Marco teórico | 3 |
| 1.1.1. Autovalores y Autovectores..... | 3 |
| 1.1.2. Normas y número de condición | 5 |
| 1.1.3. Sensibilidad de Autovalores y Autovectores | 8 |
| 1.1.3.1. Sensibilidad de autovalores y autovectores caso simétrico..... | 8 |
| 1.1.3.2. Sensibilidad de autovalores y autovectores caso no simétrico.... | 14 |
| 1.1.4. El Método QR para una matriz simétrica | 15 |
| 1.1.4.1. El método de Householder | 15 |
| 1.1.4.2. Descripción del Método QR..... | 21 |
| 1.1.4.3. Factorización QR..... | 22 |
| 1.1.4.4. Aceleración de convergencia..... | 26 |
| 1.1.5. El método QR para una matriz no simétrica..... | 28 |

| | |
|---|----|
| 1.1.5.1. Factorización QR..... | 28 |
| 1.1.5.2. Calculo de autovalores mediante el método QR | 31 |
| 1.1.6. El método de la potencia inversa con traslación..... | 32 |
| 1.1.6.1. Método de la Potencia | 32 |
| 1.1.6.2. Método de las potencias inversas con traslación..... | 34 |
| 1.2. Antecedentes | 35 |

CAPÍTULO II

PLANTEAMIENTO DEL PROBLEMA

| | |
|---------------------------------------|----|
| 2.1. Identificación del problema..... | 39 |
| 2.2. Justificación..... | 40 |
| 2.3. Hipótesis..... | 41 |
| 2.3.1. Hipótesis General..... | 41 |
| 2.3.2. Hipótesis Especifica..... | 41 |
| 2.4. Objetivos | 41 |
| 2.4.1. Objetivo General..... | 41 |
| 2.4.2. Objetivos Especifico | 41 |

CAPÍTULO III

MATERIALES Y MÉTODOS

| | |
|--|----|
| 3.1. Lugar de estudio | 42 |
| 3.2. Población | 42 |
| 3.3. Muestra..... | 42 |
| 3.4. Método de investigación | 42 |
| 3.5. Metodología para el desarrollo de algoritmos..... | 43 |

CAPÍTULO IV

RESULTADOS Y DISCUSIÓN

| | |
|---|----|
| 4.1. Análisis del problema de sensibilidad de autovalores y autovectores reales de matrices reales bien condicionadas | 46 |
| 4.1.1. Método analítico para el problema de sensibilidad de autovalores y autovectores de una matriz simétrica | 46 |
| 4.1.1.1. Cálculo de autovalores y autovectores | 46 |
| 4.1.1.2. Cálculo de cotas de sensibilidad..... | 47 |
| 4.1.1.3. Análisis de sensibilidad..... | 50 |
| 4.1.2. Método analítico para el problema de sensibilidad de autovalores y autovectores de una matriz no simétrica | 53 |
| 4.1.2.1. Cálculo de autovalores y autovectores | 53 |
| 4.1.2.2. Cálculos de cotas de sensibilidad | 54 |
| 4.1.2.3. Análisis de sensibilidad..... | 59 |
| 4.1.3. Resultado | 61 |
| 4.1.4. Discusión | 61 |
| 4.2. Diseño del algoritmo basado en el método QR para analizar la sensibilidad de autovalores y autovectores..... | 62 |
| 4.2.1. Método QR como método base | 62 |
| 4.2.2. Algoritmos básicos para el algoritmo propuesto | 64 |
| 4.2.2.1. Algoritmo de Householder | 64 |
| 4.2.2.2. Algoritmo QR para matrices simétricas | 65 |
| 4.2.2.3. Algoritmo QR Schur para matrices no simétricas..... | 67 |
| 4.2.2.4. Algoritmo de potencia inversa con traslación | 67 |
| 4.2.2.5. Algoritmo para la verificación de simetría..... | 68 |
| 4.2.3. Diseño del algoritmo propuesto..... | 69 |

| | |
|--|----|
| 4.2.4. Algoritmo basado en el método QR | 70 |
| 4.2.5. Resultado | 72 |
| 4.2.6. Discusión | 73 |
| 4.3. Codificación, ejecución y verificación del algoritmo QR extendido para analizar la sensibilidad de autovalores y autovectores | 74 |
| 4.3.1. Primera prueba para una matriz simétrica | 74 |
| 4.3.2. Segunda prueba para una matriz simétrica | 76 |
| 4.3.3. Tercera prueba para una matriz no simétrica..... | 78 |
| 4.3.4. Cuarta prueba para una matriz no simétrica | 80 |
| 4.3.5. Resultado | 81 |
| 4.3.6. Discusión | 82 |
| CONCLUSIONES | 83 |
| RECOMENDACIONES..... | 85 |
| BIBLIOGRAFÍA | 86 |
| ANEXO | 89 |

ÍNDICE DE TABLAS

| | Pág. |
|---|-------------|
| 1. Cota 1 para los autovalores de la matriz simétrica A | 50 |
| 2. Cotas 2 correspondientes a cada autovector de la matriz simétrica A | 50 |
| 3. Resultados para el análisis de sensibilidad de autovalores del caso simétrico. | 51 |
| 4. Resultados para el análisis de sensibilidad de los autovectores del caso simétrico.... | 52 |
| 5. Cota 1 para los autovalores de la matriz no simétrica A | 58 |
| 6. Cotas 2 correspondientes a cada autovector de la matriz no simétrica A | 59 |
| 7. Resultados para el análisis de sensibilidad de autovalores caso no simétrico | 59 |
| 8. Resultados para el análisis de sensibilidad de los autovectores caso no simétrico | 60 |
| 9. De autovalores de los métodos analíticos, QR, Jacobi y Leverrier-Laguerre..... | 63 |
| 10. Distancias los autovalores obtenidos por los otros métodos con relación al valor exacto | 63 |
| 11. Resultados para el análisis de sensibilidad para autovalores de la primera prueba. . | 74 |
| 12. Resultados para analizar la sensibilidad de autovectores de la primera prueba. | 75 |
| 13. Resultados para el análisis de sensibilidad para autovalores de la segunda prueba. 76 | 76 |
| 14. Resultados para analizar la sensibilidad de autovectores de la segunda prueba..... | 77 |
| 15. Resultados para el análisis de sensibilidad para autovalores de la tercera prueba.... | 78 |
| 16. Resultados para analizar la sensibilidad de autovectores de la tercera prueba. | 79 |
| 17. Resultados para el análisis de sensibilidad para autovalores de la cuarta prueba..... | 80 |
| 18. Resultados para analizar la sensibilidad de autovectores de la cuarta prueba. | 81 |

ÍNDICE DE FIGURAS

| | Pág. |
|---|-------------|
| 1. Interpretación geométrica de autovector y autovalor | 5 |
| 2. Subespacios invariantes de A y $A+E$ | 10 |
| 3: Algoritmo como una maquina | 43 |
| 4. Algoritmo basado en el método QR compuesto por otros algoritmos..... | 69 |

ÍNDICE DE ANEXOS

| | Pág. |
|---|-------------|
| 1. Código en Scilab del algoritmo propuesto..... | 89 |
| 2. Código en Scilab del algoritmo Householder..... | 90 |
| 3. Código en Scilab del algoritmo potencia inversa con traslación..... | 91 |
| 4. Código en Scilab del algoritmo QR simétrico..... | 92 |
| 5. Código en Scilab del algoritmo QR con descomposición de Schur..... | 94 |
| 6. Código en Scilab del algoritmo para verificar si es simétrica..... | 94 |
| 7. Código en Scilab del algoritmo de factorización QR..... | 94 |
| 8. Método de Jacobi..... | 95 |
| 9. Método de Leverrier-Laguerre..... | 104 |

RESUMEN

En las investigaciones de las propiedades de los autovalores y autovectores se tiene el problema de analizar la sensibilidad de autovalores y autovectores de una matriz cuadrada real considerando una matriz perturbadora de norma pequeña. Una manera de analizar sus sensibilidades es mediante la determinación de cotas de las distancias de los autovalores y subespacios invariantes generados por autovectores de la matriz original y la matriz perturbada. Estas cotas nos dan información acerca de la sensibilidad de todos los autovalores y cuál de los autovectores son más sensibles. El problema radica en el método analítico (exacto) utilizado para obtención de las cotas, es un trabajo laborioso y restringido para matrices de ordenes menores o iguales a cuatro. El objetivo de esta investigación es desarrollar un algoritmo basado en el método QR para analizar la sensibilidad de los autovalores y autovectores reales de matrices reales bien condicionadas. Para cumplir este objetivo en primer lugar se analizó el problema identificando la entrada, la salida y los procesos matemáticos que permiten resolverlo. Se diseñó un algoritmo basado en el método QR siguiendo el método denominado divide y vencerás. Luego, se codificó y ejecutó el algoritmo propuesto usando el programa Scilab; finalmente, se verificó el algoritmo haciendo pruebas a matrices especificadas cuyos resultados obtenidos se mostraron en tablas, donde se compararon las cotas obtenidas mediante el algoritmo propuesto y los obtenidos por el método analítico, donde se observaron valores muy aproximados hasta más de 10 cifras decimales exactas.

Palabras Clave: Algoritmo QR, autovalores, autovectores, cotas, método QR, sensibilidad de autovalores y autovectores, programa Scilab.

ABSTRACT

In properties investigations of eigenvalues and eigenvectors there is problem of analyzing the sensitivity of eigenvalues and eigenvectors of a real square matrix considering a perturbing matrix of small norm. One way to analyze their sensitivities is by determining levels of the distances of the eigenvalues and invariant subspaces generated by autovectors of the original matrix and the disturbed matrix. These dimensions give us information about the sensitivity of all eigenvalues and which of the eigenvectors are most sensitive. The problem lies in the analytical method (exact) used to obtain the dimensions, it is a laborious and restricted work for matrices of orders less than or equal to four. The objective of this research is to develop an algorithm based on the QR method to analyze the sensitivity of the eigenvalues and real autovectors of well conditioned real matrices. To meet this objective, the problem was first analyzed by identifying the input, the output and the mathematical processes that allow it to be solved. An algorithm based on the QR method was designed following the method called divide and conquer. Then, the proposed algorithm was encoded and executed using the Scilab program; finally, the algorithm was verified by testing specified matrices whose results were shown in tables, where the dimensions obtained by the proposed algorithm and those obtained by the analytical method were compared, where very approximate values were observed up to more than 10 exact decimal figures.

Keywords: Dimensions, eigenvalues, eigenvectors, QR algorithm, QR method, sensitivity of eigenvalues and eigenvectors, Scilab program.

INTRODUCCIÓN

En el área del álgebra lineal, uno de los grandes problemas consiste en la determinación y estudio de las propiedades de los autovalores y autovectores de una matriz cuadrada arbitraria (González Marrero et al., 2018). Su importancia radica en el hecho de que se presenta en numerosas ramas de la ciencia, como en la teoría de las ecuaciones diferenciales parciales, en la determinación de ejes de cónicas y cuádricas, en el estudio de oscilaciones de ciertas estructuras (Cervantes-Sánchez y Pérez-Lona, 2018), en sistemas dinámicos (de Lima y de Souza, 2013), entre otros.

En las investigaciones relativas al estudio de las propiedades de los autovalores y autovectores se tiene el problema de analizar la sensibilidad de autovalores y autovectores de una matriz $A \in \mathbb{R}^{n \times n}$ dada una matriz perturbadora $E \in \mathbb{R}^{n \times n}$ con $\|E\|_2$ pequeña. Una manera de analizar sus sensibilidades es mediante la determinación de cotas de las distancias de los autovalores y subespacios invariantes generados por los autovectores entre la matriz A y la matriz perturbada $A+E$. Estas cotas nos dan información acerca de la sensibilidad de todos los autovalores, así como cuál de los autovectores son más sensibles. El problema radica en que el procedimiento para obtener las cotas resulta ser laboriosas y restringidas para matrices de ordenes menores o iguales a 4, debido a que solo se pueden resolver de forma exacta ecuaciones características de grados menores o iguales a 4 por métodos analíticos (exactos), por lo que se requiere desarrollar un algoritmo basado en métodos numéricos confiables que permita hallar de manera aproximada las cotas para los autovalores y autovectores de tal manera se pueda analizar su sensibilidad.

La investigación tiene importancia altamente teórica, ya que este estudio permitirá contribuir en el área del álgebra lineal numérica, en el tema de sensibilidad de autovalores y autovectores de matrices reales bien condicionadas, más específicamente se contribuye en la teoría de la perturbación mediante métodos numéricos. Por otro lado, la investigación puede servir de base para estudios de perturbaciones de sistemas dinámicos oscilantes, en el estudio de vibraciones en edificios, en el estudio de ruidos en imágenes digitales, entre otros.

El propósito de esta investigación es desarrollar un algoritmo basado en el método QR (Francis, 1961) para analizar la sensibilidad de los autovalores y autovectores reales de matrices bien condicionadas. La investigación se utiliza una metodología deductiva y lógica de soporte dada las características de la investigación.

La investigación se ha dividido en cuatro capítulos, los cuales están divididos de la siguiente manera:

En el **Capítulo I** se desarrollan los antecedentes de la investigación y el marco teórico en el que se detallan las definiciones, propiedades y teoremas necesarias para desarrollar un algoritmo basado en el método QR para analizar la sensibilidad de autovalores y autovectores.

En el **Capítulo II** se presenta el planteamiento del problema, es decir, la identificación del problema, la justificación, la hipótesis y el objetivo de la investigación.

En el **Capítulo III** se presentan materiales y métodos de investigación, es en donde se consigna el lugar de estudio, la población, la muestra, la metodología general de la investigación, la metodología para la elaboración de algoritmos.

En el **Capítulo IV** se muestran los resultados y discusiones de la investigación; es donde se presentan los resultados correspondientes a cada objetivo específico. Y finalmente tenemos las conclusiones, recomendaciones, la bibliografía y el anexo de los algoritmos en Scilab de la investigación, entre otros.

CAPÍTULO I

REVISIÓN DE LITERATURA

1.1. Marco teórico

1.1.1. Autovalores y Autovectores

Definición (Autovalor) Si A es una matriz real de orden $n \times n$, entonces sus n autovalores $\lambda_1, \lambda_2, \dots, \lambda_n$ son las raíces, reales o complejas, de su polinomio característico

$$p(\lambda) = \det(A - \lambda I) \quad (1)$$

donde I representa la matriz identidad de orden n .

Ejemplo Determinar los autovalores de la matriz $A = \begin{bmatrix} 0 & 1 & -1 \\ -3 & 4 & -1 \\ -1 & 1 & 2 \end{bmatrix}$.

Solución

Primeramente, determinamos el polinomio característico de A

$$\begin{aligned} p(\lambda) &= \det(A - \lambda I) \\ &= \begin{vmatrix} -\lambda & 1 & -1 \\ -3 & 4 - \lambda & -1 \\ -1 & 1 & 2 - \lambda \end{vmatrix} \\ &= \lambda^3 - 6\lambda^2 + 11\lambda - 6 \\ &= (\lambda - 1)(2 - \lambda)(\lambda - 3) \end{aligned}$$

Luego, haciendo $p(\lambda) = 0$, se tiene que los autovalores de A que son 1, 2 y 3.

Definición (Autovector) Si λ es un autovalor de A y el vector no nulo V verifica

$$AV = \lambda V \quad (2)$$

entonces se dice que V es un autovector de A correspondiente al autovalor λ , lo que también se expresara diciendo que (λ, V) es una pareja autovalor-autovector de A .

Nota (Calculo de autovector) Para hallar el autovector V asociada a un autovalor λ se escribe la ecuación (2) en la forma:

$$(A - \lambda I)V = 0 \quad (3)$$

donde V es la incógnita (Grossman, 1996).

Ejemplo Hallar los autovectores correspondiente al autovalor $\lambda = 1$ del ejemplo anterior.

Solución

Considerando $\lambda = 1$ en la ecuación (3), se tiene que

$$A - (1)I = \begin{bmatrix} -1 & 1 & -1 \\ -3 & 3 & -1 \\ -1 & 1 & 1 \end{bmatrix}$$

Luego, sea $V = [x_1 \quad x_2 \quad x_3]^t$ el autovector asociado a λ . Entonces

$$\begin{aligned} [A - (1)I]V = 0 &\Rightarrow \begin{bmatrix} -1 & 1 & -1 \\ -3 & 3 & -1 \\ -1 & 1 & 1 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ 0 \end{bmatrix} \\ &\Rightarrow \begin{cases} -x_1 + x_2 - x_3 = 0 \\ -3x_1 + 3x_2 - x_3 = 0 \\ -x_1 + x_2 + x_3 = 0 \end{cases} \\ &\Rightarrow \begin{cases} x_1 = x_2 \\ x_3 = 0 \end{cases} \end{aligned}$$

Luego, los autovectores son $V = [x_1 \ x_1 \ 0]^t$, $x_1 \in \mathbb{R} - \{0\}$. Por lo tanto, los autovectores asociados a $\lambda = 1$ están dados por $V = [\beta \ \beta \ 0]^t$, $\beta \in \mathbb{R} - \{0\}$

Nota (Interpretación geométrica de los autovalores y autovectores)

Sea $A = \begin{bmatrix} 3 & -2 \\ 1 & 0 \end{bmatrix}$ una matriz que tiene como autovalor a $\lambda = 2$ con un autovector asociado $V = [2 \ 1]^t$. Consideremos además un vector cualquiera $U = [-1 \ 1]^t$. La multiplicación de la matriz A por V y U se muestra en la Figura 1. Se observa que, AV es justamente $2V$, esto es, A “estira” o dilata V en un vector $2V$, mientras que AU no tiene esa propiedad (Lay, 2007).

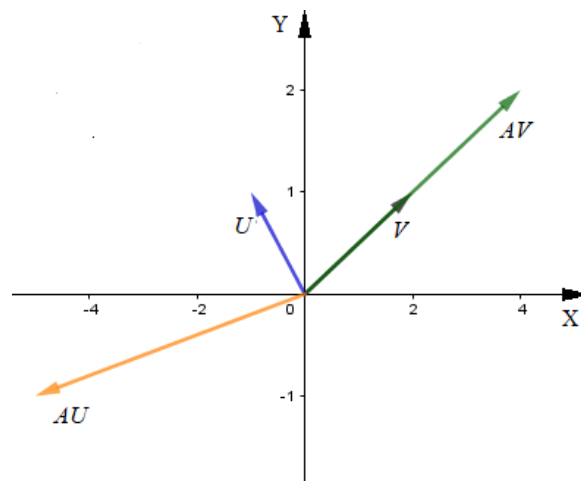


Figura 1. Interpretación geométrica de autovector y autovalor

1.1.2. Normas y número de condición

Definición (Normas de un vector) La norma de un vector en \mathbb{R}^n es una función $f : \mathbb{R}^n \rightarrow \mathbb{R}$ que satisface las siguientes propiedades:

$$\begin{array}{ll}
 f(x) \geq 0 & x \in \mathbb{R}^n \\
 f(x + y) \leq f(x) + f(y) & x, y \in \mathbb{R}^n \\
 f(\alpha x) = |\alpha| f(x) & \alpha \in \mathbb{R}, x \in \mathbb{R}^n
 \end{array}$$

Se denotaran esta función con una doble barra $f(x) = \|x\|$. Subíndices en la barra doble son usados para distinguir entre varias normas.

Nota Una clase usual de norma de vectores son las p -normas definidas por:

$$\|x\|_p = \left(|x_1|^p + \dots + |x_n|^p\right)^{\frac{1}{p}} \quad p \geq 1$$

De estas las 1,2 y ∞ normas son las más importantes, dado su mayor uso:

$$\|x\|_1 = |x_1| + \dots + |x_n|$$

$$\|x\|_2 = \left(|x_1|^2 + \dots + |x_n|^2\right)^{\frac{1}{2}} = (x^t x)^{\frac{1}{2}}$$

$$\|x\|_\infty = \max_{1 \leq i \leq n} |x_i|$$

Nota (normas de matrices) Desde que $\mathbb{R}^{m \times n}$ es isomorfo a \mathbb{R}^{mn} , la definición de norma de una matriz debería ser equivalente a la definición de norma de un vector. En particular, $f : \mathbb{R}^{m \times n} \rightarrow \mathbb{R}$ es una norma de una matriz si las siguientes propiedades se cumplen:

$$\begin{aligned} f(A) &\geq 0 & A \in \mathbb{R}^{m \times n} \\ f(A+B) &\leq f(A) + f(B) & A, B \in \mathbb{R}^{m \times n} \\ f(\alpha A) &= |\alpha| f(A) & \alpha \in \mathbb{R}, A \in \mathbb{R}^{m \times n} \end{aligned}$$

Como con las normas de un vector, se usa la notación de doble barra con subíndices para designar norma de una matriz, es decir, $f(A) = \|A\|$.

Nota Las más frecuentemente normas de matrices usadas en el álgebra lineal numérica son la norma Frobenius,

$$\|A\|_F = \sqrt{\sum_{i=1}^m \sum_{j=1}^n |a_{ij}|^2}$$

y las p -normas

$$\|A\|_p = \sup_{x \neq 0} \frac{\|Ax\|_p}{\|x\|_p}$$

Note que la p -normas de una matriz son definidos en términos de la p -normas de un vector que se vio en la sección anterior. Es claro que $\|A\|_p$ es la p -norma de un vector grande obtenido por aplicación de A para una vector unitario con p -norma:

$$\|A\|_p = \sup_{x \neq 0} \left\| A \left(\frac{x}{\|x\|_p} \right) \right\|_p = \max_{\|x\|_p=1} \|Ax\|_p$$

La Frobenius y p -normas (especialmente 1,2, ∞) son las más usadas, se presenta a continuación las normas 1 y ∞ :

$$\|A\|_1 = \max_{1 \leq j \leq n} \sum_{i=1}^m |a_{ij}|$$

$$\|A\|_\infty = \max_{1 \leq i \leq m} \sum_{j=1}^n |a_{ij}|$$

Una buena característica de la 1-norma matriz y la ∞ -norma matriz es que ellos son fácilmente computables. La característica de 2-norma es considerablemente complicada, pero considerando algunas propiedades se puede se puede definir la 2-norma de A como la raíz cuadrada de la raíz más grande de los autovalores de $A^t A$.

Nota (Número de condición de una matriz) Para estudiar la sensibilidad de autovalores de una matriz A se precisará de un número real positivo $\Gamma(A)$ llamado número de condición de la matriz A para el problema de autovalores, que nos servirá para determinar si los autovalores son poco o muy sensibles a una cierta perturbación. El número de condición de una matriz A para el problema de autovalores bajo la p -norma, está dado por:

$$\Gamma_p(A) = \inf_{Q^{-1}AQ=D} \|Q\|_p \|Q^{-1}\|_p$$

donde D corresponde a la Forma Canónica de Jordán y se verifica que $\Gamma_p(A) \geq 1$. Matrices con pequeño número de condición son llamados a ser bien condicionados. En la 2-norma, matrices simétricas son perfectamente condicionados en que $\Gamma_p(A) = 1$ ya que Q es ortogonal.

Cabe señalar que en esta investigación solo se considerará las matrices bien condicionadas, es decir, aquellas matrices donde su número de condición para el problema de autovalores es muy cercano a 1. Esta restricción es debida a que los resultados considerados para el análisis de sensibilidad requieren que se cumpla esta condición.

1.1.3. Sensibilidad de Autovalores y Autovectores

1.1.3.1. Sensibilidad de autovalores y autovectores caso simétrico

Los teoremas y corolarios de esta subsección fueron obtenidas de los textos de Golub (1996) y Stewart (1998). El siguiente resultado muestra que si A es perturbado por una matriz simétrica E entonces sus autovalores no se mueven más de $\|E\|_2$.

Teorema 1 Si A y $A+E$ son $n \times n$ matrices simétricas, entonces

$$|\lambda_k(A+E) - \lambda_k(A)| \leq \|E\|_2, k = 1, 2, \dots, n$$

Ejemplo Sea las matrices

$$A = \begin{bmatrix} 6 & 2 & 3 \\ 2 & 8 & 4 \\ 3 & 4 & 1 \end{bmatrix} \text{ y } E = \begin{bmatrix} 0.002 & 0.003 & 0.004 \\ 0.003 & 0.001 & 0.005 \\ 0.004 & 0.005 & 0.002 \end{bmatrix}$$

donde $\|E\|_2 = 0.00974$. Además, se tiene que

$$\begin{aligned} \lambda(A) &= \{-1.3708, 4.7855, 11.5853\} \\ \lambda(A+E) &= \{-1.3733, 4.7844, 11.5939\} \end{aligned}$$

luego se cumple que:

$$\begin{aligned} |-1.3733 - (-1.3708)| &= |-0.0025| \leq 0.00974 \\ |4.7844 - 4.7855| &= |-0.0011| \leq 0.00974 \\ |11.5939 - 11.5853| &= |0.0086| \leq 0.00974 \end{aligned}$$

En la presente investigación utilizara este resultado para determinar la sensibilidad de los autovalores de una matriz simétrica y además se podrá determinar la cuota que no puede sobre pasar, esto hace que no exista grandes sensibilidades exageradas en los autovalores frente a perturbaciones.

Teorema 2 Sea A y $A+E$ son matrices $n \times n$ matrices simétricas y que

$$Q = \begin{bmatrix} q_1 & Q_2 \\ 1 & n-1 \end{bmatrix} \quad n$$

es una matriz ortogonal tal que q_1 es un autovector de A . Particionar las matrices $Q^t A Q$ y $Q^t E Q$ como sigue

$$Q^t A Q = \begin{bmatrix} \lambda & 0 \\ 0 & D_2 \end{bmatrix} \quad \begin{matrix} 1 \\ n-1 \end{matrix} \quad Q^t E Q = \begin{bmatrix} \varepsilon & e^t \\ e & E_{22} \end{bmatrix} \quad \begin{matrix} 1 \\ n-1 \end{matrix}$$

Si $d = \min_{\mu \in \lambda(D_2)} |\lambda - \mu| > 0$ y $\|e\|_2 \leq \frac{d}{4}$, entonces existe $p \in \mathbb{R}^{n-1}$ satisfaciendo

$\|p\|_2 \leq \frac{4}{d} \|e\|_2$ tal que $\hat{q}_1 = (q_1 + Q_2 p) / \sqrt{1 + p^t p}$ es un autovector para $A+E$.

Además

$$\text{dist}(\text{span}(q_1), \text{span}(\hat{q}_1)) = \sqrt{1 - (q_1^t \hat{q}_1)^2} \leq \frac{4}{d} \|e\|_2.$$

Nota Se cumple:

$$\cos \theta = \frac{q_1^t \hat{q}_1}{\|q_1\| \|\hat{q}_1\|}$$

por ser vectores ortonormales

$$\|q_1\| = \|\hat{q}_1\| = 1.$$

Por lo tanto,

$$\text{dist}(\text{span}(q_1), \text{span}(\hat{q}_1)) = \sqrt{1 - (q_1^t \hat{q}_1)^2} = \sqrt{1 - \cos^2 \theta_1} = \sin \theta_1$$

Ejemplo Si $A = \begin{bmatrix} 0.999 & 0 & 0 \\ 0 & 1.001 & 0 \\ 0 & 0 & 2 \end{bmatrix}$ y $E = \begin{bmatrix} 0 & 0.01 & 0.01 \\ 0.01 & 0 & 0.01 \\ 0.01 & 0.01 & 0 \end{bmatrix}$, entonces

$$\hat{Q}^t (A + E) \hat{Q} = \begin{bmatrix} 0.9899 & 0 & 0 \\ 0 & 1.0098 & 0 \\ 0 & 0 & 2.0002 \end{bmatrix}$$

donde

$$\hat{Q} = \begin{bmatrix} -0.7418 & 0.6706 & 0.0101 \\ 0.6708 & 0.7417 & 0.0101 \\ 0.0007 & -0.143 & 0.9999 \end{bmatrix}$$

es ortogonal, que representa la matriz formado por los autovectores de $A + E$.

Sea $\hat{q}_i = \hat{Q}e_i, i = 1, 2, 3$. Así \hat{q}_i es una perturbación de autovectores de A , $q_i = e_i$

Tenemos que:

$$\text{dist}(\text{span}(q_1), \text{span}(\hat{q}_1)) = \sin \theta_1 = 0.6706$$

$$\text{dist}(\text{span}(q_2), \text{span}(\hat{q}_2)) = \sin \theta_2 = 0.6706$$

$$\text{dist}(\text{span}(q_3), \text{span}(\hat{q}_3)) = \sin \theta_3 = 0.0191$$

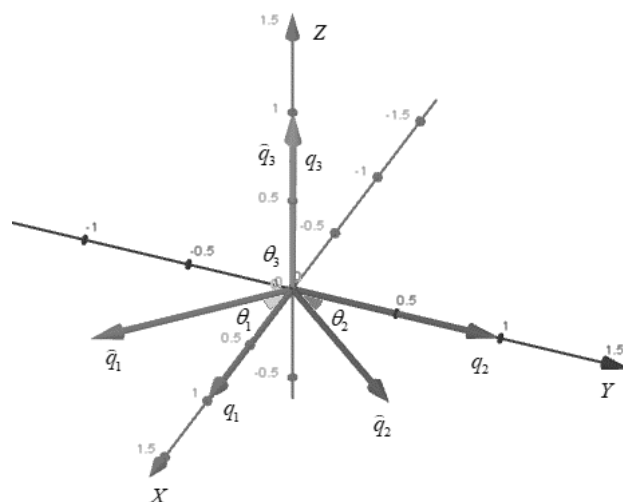


Figura 2. Subespacios invariantes de A y $A + E$

La distancia entre subespacios invariantes determinados por los autovectores que está dado por los senos de los ángulos que forman los autovectores originales con los perturbados nos servirán para determinar la sensibilidad de los autovectores.

Nota Del ejemplo anterior tenemos que los autovalores de la matriz A son $\lambda_1 = 0.999$, $\lambda_2 = 1.001$, $\lambda_3 = 2$ y su matriz de paso formado por sus autovectores respectivos

$$Q = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

Además, podemos calcular las cotas de las distancias de los subespacios invariantes generados por los autovectores, en el cual se puede apreciar la influencia que ejerce el hecho de que los autovalores se hallen más cercanos. Se tendrá tres casos:

Para el autovalor $\lambda_1 = 0.999$, tenemos que

$$Q^T A Q = \begin{bmatrix} 0.999 & 0 & 0 \\ 0 & 1.001 & 0 \\ 0 & 0 & 2 \end{bmatrix} \text{ y } Q^T E Q = \begin{bmatrix} 0 & 0.01 & 0.01 \\ 0.01 & 0 & 0.01 \\ 0.01 & 0.01 & 0 \end{bmatrix},$$

luego, considerando las sub matrices, tenemos que

$$e_1 = \begin{bmatrix} 0.01 \\ 0.01 \end{bmatrix} \quad D_2 = \begin{bmatrix} 1.001 & 0 \\ 0 & 2 \end{bmatrix}$$

esto implica que

$$\|e_1\|_2 = [\text{mayor autovalor de } e^T e]^{1/2} = 0.0141$$

$$\lambda(D_2) = \{1.001, 2\} \rightarrow d_1 = \min_{\mu \in \lambda(D_2)} |\lambda_1 - \mu| = 0.002.$$

Por lo tanto, la cota de la distancia del subespacio invariante relativo al autovalor λ_1 está dado por:

$$\frac{4}{d_1} \|e_1\|_2 = \frac{4(0.0141)}{0.002} = 28.2.$$

Para el autovalor $\lambda_2 = 1.001$, haciendo un cambio en el orden de los autovectores en Q , se tiene

$$Q = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix},$$

luego, tenemos que

$$Q' A Q = \begin{bmatrix} 1.001 & 0 & 0 \\ 0 & 0.999 & 0 \\ 0 & 0 & 2 \end{bmatrix} \text{ y } Q' E Q = \begin{bmatrix} 0 & 0.01 & 0.01 \\ 0.01 & 0 & 0.01 \\ 0.01 & 0.01 & 0 \end{bmatrix},$$

luego, considerando las sub matrices, tenemos que

$$e_2 = \begin{bmatrix} 0.01 \\ 0.01 \end{bmatrix} \quad D_2 = \begin{bmatrix} 0.999 & 0 \\ 0 & 2 \end{bmatrix},$$

esto implica que

$$\|e_2\|_2 = [\text{mayor autonalor de } e' e]^{1/2} = 0.0141$$

$$\lambda(D_2) = \{1.001, 2\} \rightarrow d_2 = \min_{\mu \in \lambda(D_2)} |\lambda_2 - \mu| = 0.002$$

Por lo tanto, la cota de la distancia del subespacio invariante relativo al autovalor λ_2 está dado por:

$$\frac{4}{d_2} \|e_2\|_2 = \frac{4(0.0141)}{0.002} = 28.2$$

Para el autovalor $\lambda_3 = 2$, haciendo un cambio en el orden de los autovectores en Q , se tiene

$$Q = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix},$$

luego, tenemos que

$$Q^T A Q = \begin{bmatrix} 2 & 0 & 0 \\ 0 & 0.999 & 0 \\ 0 & 0 & 1.001 \end{bmatrix} \text{ y } Q^T E Q = \begin{bmatrix} 0 & 0.01 & 0.01 \\ 0.01 & 0 & 0.01 \\ 0.01 & 0.01 & 0 \end{bmatrix}$$

considerando las sub matrices, tenemos que

$$e_3 = \begin{bmatrix} 0.01 \\ 0.01 \end{bmatrix} \quad D_2 = \begin{bmatrix} 0.999 & 0 \\ 0 & 1.001 \end{bmatrix},$$

esto implica que

$$\|e_3\|_2 = [\text{mayor autonalor de } e^t e]^{1/2} = 0.0141$$

$$\lambda(D_2) = \{1.001, 2\} \rightarrow d_3 = \min_{\mu \rightarrow \lambda(D_2)} |\lambda_3 - \mu| = 0.999$$

Por lo tanto, la cota de la distancia del subespacio invariante relativo al autovalor λ_3 está dado por:

$$\frac{4}{d_3} \|e_3\|_2 = \frac{4(0.0141)}{0.999} = 0.056$$

De las tres cuotas de las distancias de los subespacios, se observa que se cumple el Teorema 2 ya que hay mayor sensibilidad para autovalores cercanos.

A continuación, se describirá resultados que permitirán analizar la sensibilidad tanto de autovalores como de autovectores en particular de matrices reales no simétricas. Estos resultados difieren de los considerados para el caso simétrico, ya que ahora se considera el número de condición de la matriz en el estudio de la sensibilidad de los autovalores y autovectores ya que se requieren que sean valores cercanos a 1 para que sean bien condicionadas.

1.1.3.2. Sensibilidad de autovalores y autovectores caso no simétrico

Teorema 3 (Bauer-Fike) Si μ es un autovalor de $A + E \in \mathbb{C}^{n \times n}$ y $X^{-1}AX = D = \text{diag}(\lambda_1, \dots, \lambda_n)$, entonces

$$\min_{\lambda \in \lambda(A)} |\lambda - \mu| \leq \|X\|_p \|X^{-1}\|_p \|E\|_p$$

donde $\|\cdot\|_p$ denota la p -norma.

Nota En particular se puede escribir la desigualdad anterior considerando el número de condición de una matriz:

$$\min_{\lambda \in \lambda(A)} |\lambda - \mu| \leq \Gamma_p(A) \|E\|_p$$

Teorema 4 Sea $A, E \in \mathbb{C}^{n \times n}$ y que $Q = [q_1 \quad Q_2] \in \mathbb{C}^{n \times n}$ es unitario con $q_1 \in \mathbb{C}^n$.

Se asume

$$Q^H A Q = \begin{bmatrix} \lambda & v^H \\ 0 & T_{22} \end{bmatrix} \begin{matrix} 1 \\ n-1 \end{matrix} \quad Q^H E Q = \begin{bmatrix} \varepsilon & \gamma^H \\ \delta & E_{22} \end{bmatrix} \begin{matrix} 1 \\ n-1 \end{matrix}$$

$$\begin{matrix} 1 & n-1 \end{matrix}$$

Si $\sigma = \sigma_{\min}(T_{22} - \lambda I) > 0$ y

$$\|E\|_2 \left(1 + \frac{5\|v\|_2}{\sigma} \right) \leq \frac{\sigma}{5},$$

entonces existe $p \in \mathbb{R}^{n-1}$ satisfaciendo $\|p\|_2 \leq \frac{4}{d}\|\delta\|_2$ tal que

$\hat{q}_1 = (q_1 + Q_2 p) / \sqrt{1 + p^H p}$ es un autovector para $A + E$. Además

$$\text{dist}(\text{span}(q_1), \text{span}(\hat{q}_1)) \leq \frac{4}{\sigma}\|\delta\|_2.$$

1.1.4. El Método QR para una matriz simétrica

1.1.4.1. El método de Householder

El método de Householder se usa para encontrar una matriz simétrica tridiagonal a partir de una matriz simétrica. Luego, una vez implementado este método se recurrirá al algoritmo QR, para aproximar los autovalores de la matriz tridiagonal simétrica resultante (Burden, 1998).

Definición (Transformación de Householder) Sea $w \in \mathbb{R}^n$ con $w^T w = 1$. La matriz de $n \times n$

$$P = I - 2ww^T$$

recibe el nombre de transformación de Householder.

Las transformaciones de Householder sirven para suprimir de manera selectiva bloques de elementos de vectores o columnas de matrices en una forma extremadamente estable respecto al error de redondeo. Las propiedades de estas transformaciones se dan en el siguiente teorema.

Teorema 5 Si $P = I - 2ww^T$ es una transformación de Householder, entonces P es simétrica y ortogonal, es decir, $P^{-1} = P$.

A continuación, se describe el proceso de transformación de una matriz simétrica con entradas reales a una matriz simétrica tridiagonal mediante el método de Householder.

Sea la matriz simétrica original

$$A = \begin{bmatrix} a_{11}^{(1)} & a_{12}^{(1)} & a_{13}^{(1)} & \cdots & a_{1n}^{(1)} \\ a_{21}^{(1)} & a_{22}^{(1)} & a_{23}^{(1)} & \cdots & a_{2n}^{(1)} \\ a_{31}^{(1)} & a_{32}^{(1)} & a_{33}^{(1)} & \cdots & a_{3n}^{(1)} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ a_{n1}^{(1)} & a_{n2}^{(1)} & a_{n3}^{(1)} & \cdots & a_{nn}^{(1)} \end{bmatrix}$$

El método de Householder empieza con la determinación de la matriz de transformación $P^{(1)}$. Esta matriz servirá para obtener $A^{(2)} = P^{(1)}AP^{(1)}$ tal que

$$\begin{cases} a_{j1}^{(2)} = 0, j = 3, 4, 5, \dots, n \\ a_{11}^{(2)} = a_{11}^{(1)} \end{cases} \quad (4)$$

Se elige el vector $w = (w_1, w_2, \dots, w_n)^t$ tal que se cumpla $ww^t = 1$ y las igualdades (4). Al usar $w_1 = 0$ se garantiza que $a_{11}^{(2)} = a_{11}^{(1)}$. Queremos que $P^{(1)} = I - 2ww^t$ satisfaga

$$P^{(1)}(a_{11}, a_{21}, a_{31}, \dots, a_{n1})^t = (a_{11}, \alpha, 0, \dots, 0)^t \quad (5)$$

donde más tarde se seleccionará α . Para simplificar la notación, se utiliza

$$\begin{aligned} \hat{w} &= (w_2, w_3, \dots, w_n)^t \in \mathbb{R}^{n-1} \\ \hat{y} &= (a_{21}, a_{31}, \dots, a_{n1})^t \in \mathbb{R}^{n-1} \end{aligned}$$

y \hat{P} como la $(n-1) \times (n-1)$ transformación de Householder dado por

$$\hat{P} = I_{n-1} - 2\hat{w}\hat{w}^t$$

La ecuación (5) se convierte entonces en

$$P^{(1)} \begin{bmatrix} a_{11} \\ a_{21} \\ a_{31} \\ \vdots \\ a_{n1} \end{bmatrix} = \begin{bmatrix} 1 & 0 & \dots & 0 \\ 0 & & & \\ \vdots & & \hat{P} & \\ 0 & & & \end{bmatrix} \begin{bmatrix} a_{11} \\ \hat{y} \end{bmatrix} = \begin{bmatrix} a_{11} \\ \hat{P}\hat{y} \end{bmatrix} = \begin{bmatrix} a_{11} \\ \alpha \\ 0 \\ \vdots \\ 0 \end{bmatrix}$$

con

$$\hat{P}\hat{y} = (I_{n-1} - 2\hat{w}\hat{w}^t)\hat{y} = \hat{y} - 2(\hat{w}^t\hat{y})\hat{w} = (\alpha, 0, \dots, 0)^t \quad (6)$$

Sea $r = \hat{w}^t\hat{y}$. Entonces

$$(\alpha, 0, \dots, 0)^t = (a_{21} - 2rw_2, a_{31} - 2rw_3, \dots, a_{n1} - 2rw_n)^t$$

y podemos determinar todas las w_1 si logramos determinar α y r . Al igualar los componentes, obtenemos

$$\begin{cases} \alpha = a_{21} - 2rw_2 \\ 0 = a_{j1} - 2rw_j, j = 3, 4, \dots, n \end{cases}$$

Esto es,

$$2rw_2 = a_{21} - \alpha \quad (7)$$

y

$$2rw_j = a_{j1}, j = 3, 4, \dots, n \quad (8)$$

Al elevar al cuadrado ambos lados de las ecuaciones anteriores y al sumar, obtenemos

$$4r^2 \sum_{j=2}^n w_j^2 = (a_{21} - \alpha)^2 + \sum_{j=2}^n a_{j1}^2$$

Puesto que $w^t w = 1$ y $w_1 = 0$, tenemos $\sum_{j=2}^n w_j^2 = 1$ y

$$4r^2 = \sum_{j=2}^n a_{j1}^2 - 2\alpha a_{21} + \alpha^2, \tag{9}$$

con base en la ecuación (6) y en el hecho de que P es ortogonal, tenemos

$$\alpha^2 = (\alpha, 0, \dots, 0)(\alpha, 0, \dots, 0)^t = (\hat{P}\hat{y})^t \hat{P}\hat{y} = \hat{y}^t \hat{P}^t \hat{P}\hat{y} = \hat{y}^t \hat{y} = \sum_{j=2}^n a_{j1}^2$$

así que

$$\alpha = \pm \left(\sum_{j=2}^n a_{j1}^2 \right)^{1/2}$$

que al ser sustituida en (6) nos da

$$2r^2 = \sum_{j=2}^n a_{j1}^2 \pm a_{21} \left(\sum_{j=2}^n a_{j1}^2 \right)^{1/2}$$

Para asegurarnos de que $2r^2 = 0$ solo si $a_{21} = a_{31} = \dots = a_{n1} = 0$, elegimos el signo de modo que

$$2r^2 = \sum_{j=2}^n a_{j1}^2 + |a_{21}| \left(\sum_{j=2}^n a_{j1}^2 \right)^{1/2}$$

esto es, seleccionamos

$$\alpha = -(\text{signo } a_{21}) \left(\sum_{j=2}^n a_{j1}^2 \right)^{1/2}$$

Con estas opciones de α y $2r^2$ resolvemos las ecuaciones (7) y (8) para obtener

$$w_2 = \frac{a_{21} - \alpha}{2r}$$

y

$$w_j = \frac{a_{j1}}{2r}, j = 3, 4, \dots, n$$

Para resumir la selección de $P^{(1)}$ tenemos

$$\alpha = -(\text{signo } a_{21}) \left(\sum_{j=2}^n a_{j1}^2 \right)^{1/2}$$

$$r = \left(\frac{1}{2} \alpha^2 - \frac{1}{2} a_{21} \alpha \right)^{1/2}$$

$$w_1 = 0$$

$$w_2 = \frac{a_{21} - \alpha}{2r}$$

y

$$w_j = \frac{a_{j1}}{2r}, j = 3, 4, \dots, n$$

Con esta opción elegida, se obtiene:

$$A^{(2)} = P^{(1)} A P^{(1)} = \begin{bmatrix} a_{11}^{(2)} & a_{12}^{(2)} & 0 & \cdots & 0 \\ a_{21}^{(2)} & a_{22}^{(2)} & a_{23}^{(2)} & \cdots & a_{2n}^{(2)} \\ 0 & a_{32}^{(2)} & a_{33}^{(2)} & \cdots & a_{3n}^{(2)} \\ \vdots & \vdots & \vdots & & \vdots \\ 0 & a_{n2}^{(2)} & a_{n3}^{(2)} & \cdots & a_{nn}^{(2)} \end{bmatrix}$$

Una vez que hemos encontrado $P^{(1)}$ y $A^{(2)}$ el proceso se repite hasta obtener

$$P^{(2)} = I - 2w^{(2)} [w^{(2)}]^t \text{ donde } [w^{(2)}]^t w^{(2)} = 1 \text{ y } w_1^{(2)} = w_2^{(2)} = 0$$

con la propiedad de que

$$A^{(3)} = P^{(2)} A^{(2)} P^{(2)} = \begin{bmatrix} a_{11}^{(3)} & a_{12}^{(3)} & 0 & \cdots & 0 \\ a_{21}^{(3)} & a_{22}^{(3)} & a_{23}^{(3)} & \cdots & a_{2n}^{(3)} \\ 0 & a_{32}^{(3)} & a_{33}^{(3)} & \cdots & a_{3n}^{(3)} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 0 & a_{n2}^{(3)} & a_{n3}^{(3)} & \cdots & a_{nn}^{(3)} \end{bmatrix}$$

Continuando de este modo se forma la matriz tridiagonal y simétrica $A^{(n-1)}$ donde

$$A^{(n-1)} = P^{(n-2)} P^{(n-3)} \dots P^{(1)} A P^{(1)} \dots P^{(n-3)} P^{(n-2)}$$

En esta subsección se presenta el método QR, que es una técnica de reducción matricial que nos permite determinar simultáneamente todos los autovalores de una matriz simétrica. Para que el algoritmo QR sea eficaz, la matriz simétrica debe tener la forma tridiagonal. Si no la tiene, el primer paso consiste en aplicar el método de Householder, ya que con el se genera una matriz tridiagonal simétrica, semejante a la matriz simétrica dada (Burden, 1998).

En el resto de esta sección supondremos que la matriz simétrica cuyos autovalores vamos a calcular es tridiagonal. Si con A denotamos una matriz de este tipo, podremos simplificar la notación marcando los elementos de A de la siguiente manera:

$$A = \begin{bmatrix} a_1 & b_2 & 0 & \cdots & 0 \\ b_2 & a_2 & b_3 & \cdots & 0 \\ 0 & b_3 & a_3 & \ddots & \vdots \\ \vdots & \vdots & \ddots & \ddots & b_n \\ 0 & 0 & \cdots & b_n & a_n \end{bmatrix} \quad (10)$$

1.1.4.2. Descripción del Método QR

A continuación, se describe el proceso para la determinación de todos los autovalores de la matriz tridiagonal. Primero se tiene que ver si $b_2=0$ o si $b_n=0$, entonces la matriz 1×1 $[a_1]$ o bien $[a_n]$ produce inmediatamente un autovalor a_1 o a_n de A .

Cuando $b_j=0$ para algún j , donde $a < j < n$, el problema se puede minimizar considerando, y

$$\begin{bmatrix} a_1 & b_2 & 0 & \cdots & 0 \\ b_2 & a_2 & b_3 & \cdots & 0 \\ 0 & b_3 & a_3 & \ddots & \vdots \\ \vdots & \vdots & \ddots & \ddots & b_{j-1} \\ 0 & 0 & \cdots & b_{j-1} & a_{j-1} \end{bmatrix} \text{ y } \begin{bmatrix} a_j & b_{j+1} & 0 & \cdots & 0 \\ b_{j+1} & a_{j+1} & b_{j+2} & \cdots & 0 \\ 0 & b_{j+2} & a_{j+2} & \ddots & \vdots \\ \vdots & \vdots & \ddots & \ddots & b_n \\ 0 & 0 & \cdots & b_n & a_n \end{bmatrix} \quad (11)$$

Si ninguna de b_j es cero en la matriz A , el método QR forma una sucesión de las matrices $A = A^{(1)}, A^{(2)}, A^{(3)}, \dots$, así:

1. $A^{(1)} = A$ se factoriza como un producto $A^{(1)} = Q^{(1)}R^{(1)}$, donde $Q^{(1)}$ es ortogonal y $R^{(1)}$ es triangular superior.
2. $A^{(2)}$ se define como $A^{(2)} = R^{(1)}Q^{(1)}$

en general, para $i \geq 2$, $A^{(i)}$ se factoriza como un producto $A^{(i)} = Q^{(i)} R^{(i)}$ de una matriz ortogonal $Q^{(i)}$ y una matriz triangular superior $R^{(i)}$. Después definimos $A^{(i+1)}$ como el producto de $R^{(i)}$ y $Q^{(i)}$ en la dirección inversa $A^{(i+1)} = R^{(i)} Q^{(i)}$. Dado que $Q^{(i)}$ es ortogonal,

$$A^{(i+1)} = R^{(i)} Q^{(i)} = \left[(Q^{(i)})^t A^{(i)} \right] Q^{(i)} = (Q^{(i)})^t A Q^{(i)} \quad (12)$$

y $A^{(i+1)}$ es simétrica con los mismos autovalores que $A^{(i)}$. Por la forma en que definimos $R^{(i)}$ y $Q^{(i)}$ también podemos garantizar que $A^{(i+1)}$ es tridiagonal.

Continuando por inducción, $A^{(i+1)}$ tiene los mismos autovalores que la matriz original A . El éxito del procedimiento se debe al hecho de que $A^{(i+1)}$ tiende a una matriz diagonal con los autovalores de A a lo largo de la diagonal.

1.1.4.3. Factorización QR

Sea la matriz tridiagonal simétrica

$$A_1^{(1)} = \begin{bmatrix} a_1 & b_2 & 0 & \cdots & 0 \\ b_2 & a_2 & b_3 & \cdots & 0 \\ 0 & b_3 & a_3 & \ddots & \vdots \\ \vdots & \vdots & \ddots & \ddots & b_n \\ 0 & 0 & \cdots & b_n & a_n \end{bmatrix}$$

Definición (Matriz de rotación) Una matriz de rotación P difiere de la matriz identidad en cuatro elementos como máximo. estos tienen la forma

$$p_{ii} = p_{jj} = \cos \theta \quad \text{y} \quad p_{ij} = -p_{ji} = \sin \theta$$

para algún θ y $i \neq j$.

La factorización $A_1^{(1)} = Q^{(1)}R^{(1)}$ utiliza un producto de $n-1$ matrices de rotación de este tipo para construir

$$R^{(1)} = P_n P_{n-1} \dots P_2 A_1^{(1)}$$

Primero establecemos que la matriz de rotación P_2 tenga

$$p_{11} = p_{22} = \cos \theta_2 \text{ y } p_{12} = -p_{21} = \sin \theta_2$$

donde

$$\sin \theta_2 = \frac{b_2}{\sqrt{b_2^2 + a_1^2}} \text{ y } \cos \theta_2 = \frac{a_1}{\sqrt{b_2^2 + a_1^2}}$$

Entonces la matriz

$$A_2^{(1)} = P_2 A^{(1)}$$

tiene un cero en la posición (2,1), esto es, en el segundo renglón y en la primera columna, ya que el elemento (2,1) de $A_2^{(1)}$ es

$$(-\sin \theta_2)a_1 + (\cos \theta_2)b_2 = \frac{b_2 a_1}{\sqrt{b_2^2 + a_1^2}} + \frac{a_1 b_2}{\sqrt{b_2^2 + a_1^2}} = 0$$

Como la multiplicación $P_2 A^{(1)}$ afecta a los renglones 1 y 2 de $A^{(1)}$ la nueva matriz no necesariamente conserva los elementos cero en las posiciones (1,3), (1,4), ... y (1, n). Sin embargo, $A^{(1)}$ es tridiagonal y, por tanto, los elementos (1,4), ..., (1, n) de $A_2^{(1)}$ son cero. solo el elemento (1,3), el del primer renglón y tercera columna, puede hacerse distinto de cero.

En términos generales, seleccionamos la matriz P_k de modo que el $(k, k+1)$ -ésimo elemento de $A_k^{(1)} = P_k A_{k-1}^{(1)}$ sea cero lo cual hace que el $(k-1, k+1)$ elemento se convierta distinto de cero. La matriz $A_k^{(1)}$ tiene la forma:

$$A_k^{(1)} = \begin{bmatrix} z_1 & q_1 & r_1 & 0 & \cdots & \cdots & \cdots & \cdots & \cdots & 0 \\ 0 & \ddots & \ddots & \ddots & \ddots & & & & & \vdots \\ 0 & \ddots & \ddots & \ddots & \ddots & \ddots & & & & \vdots \\ \vdots & \ddots & 0 & z_{k-1} & q_{k-1} & r_{k-1} & \ddots & & & \vdots \\ \vdots & & \ddots & 0 & x_k & y_k & 0 & \ddots & & \vdots \\ \vdots & & & \ddots & b_{k+1} & a_{k+1} & b_{k+2} & \ddots & \ddots & \vdots \\ \vdots & & & & \ddots & \ddots & \ddots & \ddots & \ddots & 0 \\ \vdots & & & & & \ddots & \ddots & \ddots & \ddots & 0 \\ \vdots & & & & & & \ddots & \ddots & \ddots & b_n \\ 0 & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots & 0 & b_n & a_n \end{bmatrix}$$

y P_{k+1} tiene la forma

$$P_{k+1} = \begin{bmatrix} I_{k-1} & 0 & 0 & 0 \\ 0 & c_{k+1} & s_{k+1} & 0 \\ 0 & -s_{k+1} & c_{k+1} & 0 \\ 0 & 0 & 0 & I_{n-k-1} \end{bmatrix}$$

Elegimos las constantes $c_{k+1} = \cos \theta_{k+1}$ y $s_{k+1} = \sin \theta_{k+1}$ en P_{k+1} de modo que el $(k+1, k)$ elemento $A_{k+1}^{(1)}$ sea cero, es decir:

$$s_{k+1} x_k - c_{k+1} b_{k+1} = 0$$

Como $c_{k+1}^2 + s_{k+1}^2 = 1$ la solución de esta ecuación es

$$s_{k+1} = \frac{b_{k+1}}{\sqrt{b_{k+1}^2 + x_k^2}} \quad \text{y} \quad c_{k+1} = \frac{x_k}{\sqrt{b_{k+1}^2 + x_k^2}}$$

y $A_{k+1}^{(1)}$ tiene la forma

$$A_k^{(1)} = \begin{bmatrix} z_1 & q_1 & r_1 & 0 & \cdots & \cdots & \cdots & \cdots & \cdots & 0 \\ 0 & \ddots & \ddots & \ddots & \ddots & & & & & \vdots \\ 0 & \ddots & \ddots & \ddots & \ddots & \ddots & & & & \vdots \\ \vdots & \ddots & 0 & z_k & q_k & r_k & \ddots & & & \vdots \\ \vdots & & \ddots & 0 & x_{k+1} & y_{k+1} & 0 & \ddots & & \vdots \\ \vdots & & & \ddots & b_{k+1} & a_{k+1} & b_{k+2} & \ddots & \ddots & \vdots \\ \vdots & & & & \ddots & \ddots & \ddots & \ddots & \ddots & 0 \\ \vdots & & & & & \ddots & \ddots & \ddots & \ddots & 0 \\ \vdots & & & & & & \ddots & \ddots & \ddots & b_n \\ 0 & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots & 0 & b_n & a_n \end{bmatrix}$$

Al proseguir con esta construcción en la sucesión P_2, \dots, P_n obtenemos la matriz triangular superior

$$R^{(1)} = A_n^{(1)} = \begin{bmatrix} z_1 & q_1 & r_1 & 0 & \cdots & \cdots & \cdots & 0 \\ 0 & \ddots & \ddots & \ddots & \ddots & & & \vdots \\ \vdots & \ddots & \ddots & \ddots & \ddots & \ddots & & \vdots \\ \vdots & & \ddots & \ddots & \ddots & \ddots & & \vdots \\ \vdots & & & \ddots & \ddots & \ddots & & 0 \\ \vdots & & & & \ddots & \ddots & & r_{n-2} \\ \vdots & & & & & \ddots & & q_{n-1} \\ \vdots & & & & & & z_{n-1} & q_{n-1} \\ 0 & \cdots & \cdots & \cdots & \cdots & \cdots & 0 & x_n \end{bmatrix}$$

La otra mitad de la factorización QR es la matriz

$$Q^{(1)} = P_2^t P_3^t \dots P_n^t$$

porque la ortogonalidad de las matrices de rotación implica que

$$Q^{(1)} R^{(1)} = (P_2^t P_3^t \dots P_n^t)(P_n \dots P_3 P_2) A^{(1)} = A^{(1)}.$$

La matriz $Q^{(1)}$ es ortogonal por que

$$(Q^{(1)})^t Q^{(1)} = (P_2' P_3' \dots P_n')^t (P_2' P_3' \dots P_n') = P_n \dots P_3 P_2 P_2' P_3' \dots P_n' = I$$

Además, $Q^{(1)}$ es una matriz de Hessemberg superior.

En consecuencia, $A^{(2)} = R^{(1)} Q^{(1)}$ es también una matriz de Hessemberg superior ya que la multiplicación de $Q^{(1)}$ de la izquierda por la matriz triangular superior $Q^{(1)}$ no influye en los elementos del triángulo inferior. Lo anterior implica que $A^{(2)}$ efectivamente es tridiagonal, pues ya sabemos que es simétrica.

Los elementos situados fuera de la diagonal de $A^{(2)}$ generalmente serán más pequeños que los correspondientes de $A^{(1)}$, así que $A^{(2)}$ se acerca más a ser una matriz diagonal que $A^{(1)}$. El proceso se repite para construir $A^{(3)}, A^{(4)}, \dots$

Si los autovalores de A tienen módulos distintos y si están ordenados por $|\lambda_1| > |\lambda_2| > \dots > |\lambda_n|$ el método QR converge, es decir, $A^{(i+1)}$ tiende a ser una matriz diagonal. la rapidez de convergencia de un elemento fuera de la diagonal $b_{j+1}^{(i+1)}$ cero, depende de la razón $|\lambda_{j+1}/\lambda_j|$. Por tanto, la convergencia de $a_j^{(i+1)}$ en el autovalor λ_j puede ser lenta si $|\lambda_{j+1}/\lambda_j|$ se acerca la unidad.

1.1.4.4. Aceleración de convergencia

Si se quiere acelerar esta convergencia se selecciona una constante s cercana a un autovalor de A , con los cual se modifica la factorización de (12), para escoger $Q^{(i)}$ y $R^{(i)}$ de modo que

$$A^{(i)} - sI = Q^{(i)} R^{(i)} \tag{13}$$

y equivalentemente, podemos definir que la matriz $A^{(i+1)}$ sea

$$A^{(i+1)} = R^{(i)}Q^{(i)} = sI \quad (14)$$

Con esta modificación, rapidez de convergencia de $b_{j+1}^{(i+1)}$ a cero depende de la razón $\left| \frac{\lambda_{j+1} - s}{\lambda_j - s} \right|$, la cual puede ocasionar una mejora significativa en la rapidez original de convergencia de $A_j^{(i+1)}$ en λ_j si s está cerca de λ_{j+1} pero no de λ_j .

En el listado del algoritmo QR, cambiamos s en cada paso para que, cuando A tenga los autovalores del módulo definido $b_n^{(i+1)}$ converja a cero más rápido que $b_j^{(i+1)}$ para cualquier j menor que n . Cuando $b_n^{(i+1)}$ es suficientemente pequeña, suponemos que $\lambda_n \approx a_n^{(i+1)}$, eliminamos el n -ésimo renglón y columna de la matriz y procedemos en la misma forma para obtener una aproximación a λ_{n-1} . El proceso continúa hasta determinar una aproximación para cada autovalor.

El algoritmo incorpora el método de desplazamiento al seleccionar en el i -ésimo paso la constante de desplazamiento s_i , donde s_i es el autovalor más cercano a $a_n^{(i)}$ de la matriz

$$E^{(i)} = \begin{bmatrix} a_{n-1}^{(i)} & b_n^{(i)} \\ b_n^{(i)} & a_n^{(i)} \end{bmatrix}$$

Este desplazamiento traduce los autovalores de A por un factor s_i con esta técnica de desplazamiento, la convergencia suele ser cubica. El algoritmo acula esto desplazamientos hasta que $b_n^{(i+1)} \approx 0$ luego agrega los desplazamientos a $a_n^{(i+1)}$ para aproximar el autovalor λ_n .

Si A tiene los autovalores del mismo modulo $b_j^{(i+1)}$ puede tender a cero para algún $j \neq n$ con mayor rapidez que $b_n^{(i+1)}$. En este caso, el método de desplazamiento de matrices descrito en (11) puede servir para convertir el problema en uno que contenga un par de matrices de orden reducido. el algoritmo QR ejecuta de este modo.

1.1.5. El método QR para una matriz no simétrica

Este método construye una sucesión de matrices semejantes a una dada que tiende hacia una matriz triangular superior. Los autovalores se leen entonces en la diagonal. El método supone que la matriz que se estudia es diagonalizable. Se basa en la factorización o descomposición QR, que consiste en escribir cualquier matriz como producto de un matriz ortogonal Q por otra triangular superior R . Esta factorización se consigue mediante la llamada transformación de Householder que se detalla en la siguiente subsección (Vazquez y Jimenz, 2009).

1.1.5.1. Factorización QR

Ahora utilizara las transformaciones de Householder para hallar la factorización de cualquier matriz A como producto $A=QR$, siendo Q una matriz ortogonal ($Q^t = Q^{-1}$) y R una matriz triangular superior.

El proceso es un algoritmo con un número finito de pasos. Partimos de una matriz A de dimensiones $m \times m$. Consideramos su primera columna como un vector columna c_1 de \mathbb{R}^m :

$$A = \begin{bmatrix} a_{11} & a_{12} & \dots & a_{1m} \\ a_{21} & a_{22} & \dots & a_{2m} \\ \vdots & \vdots & \ddots & \vdots \\ a_{m1} & a_{m2} & \dots & a_{mm} \end{bmatrix} \text{ y } c_1 = \begin{bmatrix} a_{11} \\ a_{21} \\ \vdots \\ a_{m1} \end{bmatrix}$$

definimos entonces el vector de Householder para c_1 :

$$v_1 = c_1 - \alpha_1 e_1$$

con $\alpha_1 = -\text{signo}(a_{11})\|c_1\|_2$ y $e_1 = (1,0,0, \dots, 0) \in \mathbb{R}^m$ y construimos la matriz de Householder

$$H_1(v_1) = I - \frac{2}{\|v_1\|_2^2} v_1 v_1^t$$

Definimos una nueva matriz $P_1 = H_1$; multiplicándola por A obtenemos una matriz que llamamos A_1 con la primera columna igual a $H_1 c_1$, es decir, toda nula salvo su primer elemento

$$A_1 = P_1 A = \begin{bmatrix} \alpha_1 & a_{12}^1 & a_{13}^1 & \cdots & a_{1m}^1 \\ 0 & a_{22}^1 & a_{23}^1 & \cdots & a_{2m}^1 \\ 0 & a_{32}^1 & a_{33}^1 & \cdots & a_{3m}^1 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 0 & a_{m2}^1 & a_{m3}^1 & \cdots & a_{mm}^1 \end{bmatrix}$$

La idea es ahora repetir lo anterior para el bloque formado por las filas 2 en adelante y las columnas 2 en adelante.

Tomamos el vector columna $c_2 = [a_{22}^1 \ a_{32}^1 \ \dots \ a_{m2}^1]^t$, ahora en \mathbb{R}^{m-1} y construimos $v_2 = c_2 - \alpha_2 e_2$ con $\alpha_2 = -\text{signo}(a_{22}^1)\|c_2\|_2$ y $e_2 = (1,0,0, \dots, 0) \in \mathbb{R}^{m-1}$. La matriz de Householder $H_2(v_2)$ es una matriz de dimensiones $m-1 \times m-1$ y definimos la matriz P_2 de dimensiones $m \times m$ como:

$$P_2 = \begin{bmatrix} 1 & O \\ O & H_2 \end{bmatrix}$$

Haciendo ahora el correspondiente producto, tenemos:

$$A_2 = P_2 A_1 = \begin{bmatrix} \alpha_1 & a_{12}^1 & a_{13}^1 & \cdots & a_{1m}^1 \\ 0 & \alpha_2 & a_{23}^2 & \cdots & a_{2m}^2 \\ 0 & 0 & a_{33}^2 & \cdots & a_{3m}^2 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & a_{m3}^2 & \cdots & a_{mm}^2 \end{bmatrix}$$

Y aplicamos el proceso al nuevo bloque de la fila 3 en adelante y para la columna 3 en adelante. Cuando estamos tratando la columna k , c_k es un vector de \mathbb{R}^{m-k+1} , H_k es una matriz de dimensiones $(m-k+1) \times (m-k+1)$ y

$$P_k = \begin{bmatrix} I_{k-1 \times k-1} & O \\ O & H_k \end{bmatrix}$$

Si la matriz de partida A no es invertible, puede ocurrir que en este proceso se obtenga en algún momento un vector c_k nulo. En ese caso hay que tomar $H_k = I_{m-k+1}$, es decir, $P_k = I_{m \times m}$ y seguir (es decir, la columna a tratar ya tiene ceros debajo de la diagonal y podemos sin alterar nada, pasar a la siguiente columna).

Al final del proceso tenemos

$$A_{m-1} = P_{m-1} A_{m-1} = \cdots = P_{m-1} P_{m-2} \cdots P_2 P_1 A$$

Siendo la matriz A_{m-1} triangular superior. Definimos $R = A_{m-1}$ y $Q^t = P_{m-1} P_{m-2} \cdots P_2 P_1$. Vamos a comprobar que Q es ortogonal y que por tanto de la igualdad $R = Q^t A$ se deduce $A = QR$, en efecto:

Para $P_1 = H_1$ es una matriz de Householder, que es ortogonal y además simétrica.

En cuanto a una matriz genérica P_k con $k = 2, 3, \dots, m-1$.

$$P_k^t = \begin{bmatrix} I_{k-1 \times k-1} & \mathbf{O} \\ \mathbf{O} & H_k^t \end{bmatrix} = \begin{bmatrix} I_{k-1 \times k-1} & \mathbf{O} \\ \mathbf{O} & H_k^{-1} \end{bmatrix} = P_k^{-1},$$

la que por ser H_k matriz de Householder es ortogonal. Además, las matrices de Householder son simétricas y la identidad también lo es, P_k es simétrica. Por lo tanto, cada una de las matrices P_i con $i=1,2,\dots,m-1$ es ortogonal y simétrica. Veamos que su producto es ortogonal (aunque no simétrico):

$$\begin{aligned} Q^t &= P_{m-1} P_{m-2} \dots P_2 P_1 \\ &= (P_1^{-1} P_2^{-1} \dots P_{m-2}^{-1} P_{m-1}^{-1})^{-1-1} \\ &= (P_1^t P_2^t \dots P_{m-2}^t P_{m-1}^t)^{-1} \\ &= ((P_{m-1} P_{m-2} \dots P_2 P_1)^t)^{-1} \\ &= Q^{-1} \end{aligned}$$

con todo esto tenemos

$$\begin{aligned} R &= Q^t A \\ R &= Q^{-1} A \end{aligned}$$

Finalmente,

$$A = QR$$

1.1.5.2. Calculo de autovalores mediante el método QR

Este método crea a partir de una matriz A una sucesión de matrices A_n semejantes a A que converge a una matriz triangular superior. Esta convergencia esta asegura si A es diagonalizable con todos los autovalores simples (no se repite ninguno) y teniendo además todos ellos, además, distinto valor absoluto:

$$|\lambda_1| > |\lambda_2| > \dots > |\lambda_m|.$$

Sea $A = A_0$ una matriz $m \times m$. Construimos $A_1 = R_0 Q_0$, siendo Q_0 y R_0 las matrices de la factorización QR de $A_0 : A_0 = Q_0 R_0$. En general, dada A_n y su factorización $A_n = Q_n R_n$ definimos $A_{n+1} = R_n Q_n$. Se puede demostrar que bajo las hipótesis mencionadas para A la sucesión converge hacia una matriz triangular superior semejante a A . Si vamos a demostrar sin embargo, que las matrices de la sucesión son semejantes: puesto que $A = A_0$, estas dos matrices son obviamente similares. También se cumple para todo n :

$$A_{n+1} = R_n Q_n = (Q_n^{-1} Q_n) R_n Q_n = Q_n^{-1} (Q_n R_n) Q_n = Q_n^{-1} A_n Q_n$$

Por lo tanto, A_n y A_{n+1} son semejantes y queda demostrado que toda matriz A_n es semejante a A .

1.1.6. El método de la potencia inversa con traslación

En esta parte se desarrollará el método de la potencia inversa con traslación basado en el método de la potencia inversa. Este método permite hallar la pareja autovalor-autovector dada una buena aproximación del autovalor, luego este método es fundamental para hallar todos los autovectores asociados a los autovalores. Este método generado nos permitirá determinar los algoritmos basados en el método QR para analizar la sensibilidad de autovalores y autovectores.

1.1.6.1. Método de la Potencia

Antes de enunciar el teorema que sustenta el método de potencias se necesita algunos conceptos importantes como los que se da a continuación.

Definición (Autovalor-autovector dominante) Si λ_1 es un autovalor de A y que en valor absoluto es mayor que cualquier otro autovalor, entonces se dice que es un autovalor dominante y sus autovectores correspondientes se llaman autovectores dominantes.

Definición (Vector normalizado) Se dice que un autovector V está normalizado cuando su coordenada de mayor tamaño es 1.

Nota Para normalizar un autovector $[v_1 \ v_2 \ \dots \ v_n]^t$, basta construir el nuevo autovector $V = \frac{1}{c} [v_1 \ v_2 \ \dots \ v_n]^t$, donde $c = v_j$ y $|v_j| = \max \{|v_i| : 1 \leq i \leq n\}$.

Para poder hallar la pareja autovalor-autovector dominante de una matriz A que nuestro caso será simétrica, se comienza suponiendo que la matriz A tiene un autovalor dominante λ y que hay un único (salvo el signo) autovector real normalizado V correspondiente a λ . Este par autovalor-autovector (λ, V) puede encontrarse mediante el siguiente procedimiento iterativo conocido como el método de las potencias. empecemos con el vector

$$X_0 = [1 \ 1 \ \dots \ 1]^t \tag{15}$$

y generamos recursivamente una sucesión $\{X_k\}$ de acuerdo con el siguiente esquema

$$\begin{aligned} Y_k &= AX_k \\ X_{k+1} &= \frac{1}{c_{k+1}} Y_k \end{aligned} \tag{16}$$

donde c_{k+1} es la coordenada de Y_k de mayor tamaño (en caso de empate, se toma la que aparezca en primer lugar). Las sucesiones $\{X_k\}$ y $\{c_k\}$ convergerán, respectivamente a V y λ . Esto es,

$$\lim_{k \rightarrow \infty} X_k = V \text{ y } \lim_{k \rightarrow \infty} c_k = \lambda \tag{17}$$

Teorema 6 (Método de las potencias) Supongamos que una matriz A de orden $n \times n$ tiene n autovalores distintos $\lambda_1, \lambda_2, \dots, \lambda_n$ ordenados en tamaño decreciente

$$|\lambda_1| > |\lambda_2| > \dots > |\lambda_n| \quad (18)$$

Si X_0 se escoge adecuadamente (veremos luego que significa esto), entonces las sucesiones $\{X_k = [x_1^{(k)} \quad x_2^{(k)} \quad \dots \quad x_n^{(k)}]^t\}$ y $\{c_k\}$ generadas recursivamente con el esquema iterativo

$$Y_k = AX_k \quad (19)$$

y

$$X_{k+1} = \frac{1}{c_{k+1}} Y_k \quad (20)$$

donde

$$c_{k+1} = x_j^{(k)} \quad \text{y} \quad x_j^{(k)} = \max \{|x_i^{(k)}| : 1 \leq i \leq n\} \quad (21)$$

convergen, respectivamente al autovector dominante V_1 y autovalor dominante λ_1 . Es decir,

$$\lim_{k \rightarrow \infty} X_k = V_1 \quad \text{y} \quad \lim_{k \rightarrow \infty} c_k = \lambda_1 \quad (22)$$

1.1.6.2. Método de las potencias inversas con traslación

Se describirá ahora el método de las potencias inversas con traslación. Para aplicar este método es necesario disponer de una buena aproximación inicial de un autovalor y lo que se consigue mediante las iteraciones del método de la potencia inversa con traslación es obtener una solución más precisa. La aproximación inicial puede obtenerse mediante otros procedimientos. Hay tener en cuenta, sin embargo, que cuando el autovalor es complejo, cuando es

múltiple o cuando hay una pareja de autovalores del mismo tamaño, aparecen dificultades computacionales que deben abordarse con métodos más avanzados, los ejemplos que presentaremos ilustran el caso en los que los autovalores son todos distintos (Mathews, 2000). El método de las potencias inversas con traslación se basa en los siguientes tres resultados.

El siguiente teorema es fundamental para poder obtener el algoritmo que nos permita hallar el autovector asociado a un determinado autovalor.

Teorema 7 (Método de las potencias inversas con traslación) Supongamos que una matriz A de orden $n \times n$ posee n autovalores distintos $\lambda_1, \lambda_2, \dots, \lambda_n$ y fijemos uno de estos autovalores λ_j . Entonces existe una constante α tal que $\mu_1 = 1/(\lambda_j - \alpha)$ es el autovalor dominante de la matriz $(A - \alpha I)^{-1}$. Es más, si X_0 se escoge adecuadamente, entonces las sucesiones $\{X_k = [x_1^{(k)} \ x_2^{(k)} \ \dots \ x_n^{(k)}]^t\}$ y $\{c_k\}$ generada recursivamente por las formulas

$$Y_k = (A - \alpha I)^{-1} X_k \text{ y } X_{k+1} = \frac{1}{c_{k+1}} Y_k$$

donde $c_{k+1} = x_j^{(k)}$ y $x_j^{(k)} = \max \{|x_i^{(k)}| : 1 \leq i \leq n\}$ converge a la pareja autovalor-autovector dominante (μ_1, V_j) de la matriz $(A - \alpha I)^{-1}$. Finalmente, el autovalor de A que hemos fijado viene dado por $\lambda_j = \frac{1}{\mu_1} + \alpha$.

1.2. Antecedentes

Los antecedentes de la investigación corresponden a todas aquellas investigaciones relacionadas al método QR:

Yao (2015) hace una comparación del método QR, método de Jacobi y el método divide y conquistaras. Para hacer la comparación de estos métodos se usa una matriz simétrica triangular. Además, se considera las ventajas y desventajas de cada método.

Aurentz *et al.* (2017) presentan un nuevo algoritmo para resolver el problema de hallar el autovalor de una matriz simétrica tridiagonal, este algoritmo funciona primero usando una transformación de Cayley para convertir la matriz simétrica en una unitaria y luego utiliza el algoritmo QR unitario de Gragg. Demuestra que bajo suposiciones razonables que este algoritmo es más estable. Es comparable a otros algoritmos en términos de exactitud. Aunque no es el algoritmo más rápido, tampoco es visiblemente lento. Lo es aproximadamente tan rápido como el algoritmo QR para matrices tridiagonales simétricas.

Aurentz *et al.* (2015) presentan una implementación rápida en Fortran de una variante del algoritmo QR de Hessemberg unitario de Gragg. Además, se demuestra que todos los algoritmos de tipo QR y QZ para los problemas de autovalores unitarios son equivalentes. El algoritmo es estable hacia atrás. Se presentan experimentos numéricos que confirman la estabilidad hacia atrás y comparan la velocidad y precisión de este algoritmo con otros métodos.

Garrett *et al.* (2016) en este artículo se presenta una variación del método QR no lineal de Kublanovskaya para resolver problemas de autovalores no lineales. El nuevo método es iterativo y está diseñado específicamente para problemas demasiado grandes para usar técnicas de álgebra lineal densa. Para el problema del autovalor no lineal no estructurada, se utiliza una nueva estructura de datos para almacenar las matrices para mantener bajos los costos de cálculo y de memoria. Además, se presenta un algoritmo para calcular autovalores propios no lineales cercanos a los ya calculados. Finalmente, se dan ejemplos numéricos para mostrar la eficacia de los nuevos métodos, y el código fuente se ha puesto a disposición del público.

Cho e Ipsen (1998) para matrices con un único autovalor, se analiza la sensibilidad del autovalor a las perturbaciones en la matriz. Se derivan límites simples que separan la influencia del índice en la sensibilidad del autovalor de la influencia de la no normalidad. Los límites sugieren que el error del autovalor crece proporcionalmente a la desviación

de la normalidad cuando el índice es grande, mientras que un índice pequeño debilita la influencia de la no normalidad en el error del autovalor. Los experimentos numéricos con matrices que son diagonalmente similares a una matriz de Jordán ilustran que los límites predicen con precisión el comportamiento del error. También se muestra cómo determinar una descomposición de Schur refinada de una matriz de Jordán.

Panju (2011) examina algunos métodos iterativos numéricos para calcular los autovalores y los autovectores de matrices reales. Los cinco métodos examinados aquí van desde el método de iteración de potencia simple hasta el método de iteración QR más complicado. Las derivaciones, el procedimiento y las ventajas de cada método se explican brevemente.

Ballard *et al.* (2018) mencionan que la comunicación entre procesadores a menudo domina el tiempo de ejecución de los cálculos matriciales grandes. Presentan un algoritmo paralelo para calcular las descomposiciones QR cuyo costo de ancho de banda (volumen de comunicación) se puede disminuir al costo de aumentar su costo de latencia (número de mensajes). Al variar un parámetro para navegar por el ancho de banda / comercio de latencia, podemos ajustar este algoritmo para máquinas con diferentes costos de comunicación.

Yeralan *et al.* (2017) mencionan que la factorización de matriz dispersa implica una combinación de cálculos regulares e irregulares, lo que constituye un desafío particular al intentar obtener un alto rendimiento en los núcleos de computación de uso general altamente paralelos disponibles en unidades de procesamiento de gráficos (GPUs). Se presenta un método de factorización QR que se enfrenta a este desafío y es hasta once veces más rápido que un método altamente optimizado en una CPU multinúcleo. Nuestro método factoriza muchas matrices en paralelo y mantiene todos los datos transmitidos entre matrices en la GPU. Este nuevo algoritmo amplía la factorización QR que evita la comunicación para matrices densas, explotando más paralelismo y explotando la forma de escalera presente en las matrices.

Laskar y Behera (2017) el objetivo de este trabajo es comparar el método de Jacobi y el método de Givens para encontrar los valores propios y los vectores propios correspondientes de matrices simétricas reales. Finalmente, hemos visto que con ejemplos, el método de Givens no es iterativo y es más eficiente que el método de Jacobi,

aunque requiere la matriz simétrica dada en una matriz tridiagonal que tenga los mismos valores propios.

Badia et al. (2000) proponen dos nuevos algoritmos que utilizan la iteración de Laguerre para calcular los valores propios de una matriz tridiagonal simétrica. El primer algoritmo aplica esta iteración en el marco del método de bisección, logrando unas prestaciones mejores que las de cualquier otro método de cálculo de raíces usado hasta ahora. El segundo algoritmo utiliza el paradigma divide y vencerás y combina la utilización de modificaciones de rango uno con la iteración de Laguerre. Estos dos nuevos algoritmos se comparan con otros cinco que utilizan diferentes métodos para resolver el mismo problema de valores propios. Desde el punto de vista experimental se analizan como influyen las características de las matrices en las prestaciones de los algoritmos. Los clústeres y los valores propios ocultos son determinantes en los comportamientos de los algoritmos de bisección. Mientras que las prestaciones de los algoritmos del tipo divide y vencerás dependen fundamentalmente del número de deflaciones detectadas. También probamos que este algoritmo de bisección es más rápido que los basados en el método de divide y vencerás en casi todos los casos, y proporciona más precisión en el cálculo de los valores propios que la iteración QR.

Bona (1999) desarrolla un programa que genera matrices con autovalores prefijados, eligiendo cuáles de ellos está bien y cuales mal condicionadas, es decir prefijando la sensibilidad, la dimensión de la matriz, sus autovalores –con la sensibilidad requerida- y la matriz perturbadora son datos de entrada. El trabajo se basa en el Teorema de Bauer-Fike, la fórmula de Wilkinson para el número de condición de los autovalores, controlando factores que gobiernan la sensibilidad en el proceso de construcción de la matriz. El trabajo se completa con un análisis de sensibilidad de autovalores y autovectores.

CAPÍTULO II

PLANTEAMIENTO DEL PROBLEMA

2.1. Identificación del problema

En el área del álgebra lineal, uno de los grandes problemas consiste en la determinación y estudio de las propiedades de los autovalores y autovectores de una matriz cuadrada arbitraria (González *et al.*, 2018). Su importancia radica en el hecho de que se presenta en numerosas ramas de la ciencia, como en la teoría de ecuaciones diferenciales parciales, en la determinación de ejes de cónicas y cuádricas, en el estudio de oscilaciones de ciertas estructuras (Cervantes y Lona, 2018), en sistemas dinámicos (Lima y Souza, 2013), etc.

En las investigaciones relativas al estudio de las propiedades de los autovalores y autovectores se tiene el problema de analizar la sensibilidad de autovalores y autovectores de una matriz $A \in \mathbb{C}^{n \times n}$ dada una matriz perturbadora $E \in \mathbb{C}^{n \times n}$ con $\|E\|_2$ pequeña. Una manera de analizar sus sensibilidades es mediante la determinación de cotas de las distancias de los autovalores y subespacios invariantes generados por autovectores entre la matriz A y la matriz perturbada $A+E$. Estos datos nos dan información acerca de la sensibilidad de todos los autovalores, así como cuál de los autovectores son más sensibles

El problema radica en el método analítico (método exacto) utilizado para obtención de las cotas, que resulta ser un trabajo laborioso y restringido para matrices de ordenes menores o iguales a 4, esto debido a que la ecuación característica solo tiene solución exacta (mediante radicales) para ecuación de grado menores o iguales a 4, por lo que se

requiere un algoritmo basado en un método numérico confiable y eficiente que nos permita aproximarlos.

ENUNCIADO DEL PROBLEMA

El problema de investigación queda determinado bajo la siguiente interrogante:

¿Un algoritmo basado en el método QR permitirá analizar la sensibilidad de los autovalores y autovectores reales de matrices reales bien condicionadas?

2.2. Justificación

La investigación tiene una justificación teórica, ya que este estudio permitirá contribuir en el área del álgebra lineal numérica, en el tema de sensibilidad de autovalores y autovectores de matrices reales simétricas y no simétricas bien condicionadas, más específicamente se contribuye en la teoría de la perturbación de matrices reales mediante métodos numéricos.

La investigación tiene una justificación práctica ya que puede servir de base para investigaciones en el campo de los sistemas dinámicos oscilantes, como es el caso de los sistemas vibratorios masa resorte con varios grados de libertad, movimientos oscilatorios de rotacionales, oscilaciones de estructuras de edificios producidos por sismos, así mismo, también en el estudio de la estabilidad estructural de sistemas de ecuaciones diferenciales ordinarias lineales hiperbólicos, en el estudio de perturbaciones de imágenes digitales debida a ruidos, entre otros.

También, la investigación tiene una justificación metodológica, ya que, como se estableció el método deductivo como metodología de la investigación y en este marco, se estableció la metodología para desarrollar el algoritmo, conformado por la etapa de análisis, diseño, codificación, ejecución y verificación, esta es análogo con las etapas de solución de problemas matemáticos.

Fundamentalmente, el algoritmo propuesto basado en el método QR para analizar la sensibilidad de autovalores y autovectores reales de matrices reales bien condicionadas, se justifica por su novedad en el área de álgebra lineal computacional, esto es, la

utilización de algoritmos en el estudio de las propiedades de sensibilidad de los autovalores y autovectores en el área del álgebra lineal.

2.3. Hipótesis

2.3.1. Hipótesis General

Un algoritmo basado en el método QR permite analizar la sensibilidad de autovalores y autovectores reales de matrices reales bien condicionadas.

2.3.2. Hipótesis Específica

- a. El análisis del problema permite determinar la sensibilidad de los autovalores y autovectores reales de matrices reales bien condicionadas.
- b. El diseño del algoritmo basado en el método QR permite analizar la sensibilidad de los autovalores y autovectores reales de matrices reales bien condicionadas.
- c. La codificación, ejecución y verificación del algoritmo basado en el método QR permite analizar la sensibilidad de los autovalores y autovectores reales de matrices reales bien condicionadas.

2.4. Objetivos

2.4.1. Objetivo General

Desarrollar un algoritmo basado en el método QR para analizar la sensibilidad de los autovalores y autovectores reales de matrices reales bien condicionadas.

2.4.2. Objetivos Específico

- a. Analizar el problema del análisis de sensibilidad de los autovalores y autovectores reales de matrices reales bien condicionadas.
- b. Diseñar un algoritmo basado en el método QR extendido para analizar la sensibilidad de los autovalores y autovectores reales de matrices reales bien condicionadas.
- c. Codificar, ejecutar y verificar un algoritmo basado en el método QR para analizar la sensibilidad de los autovalores y autovectores reales de matrices reales bien condicionadas.

CAPÍTULO III

MATERIALES Y MÉTODOS

3.1. Lugar de estudio

La presente investigación se realizó en la región Puno - Perú.

3.2. Población

La población de estudio son todas las matrices reales con autovalores y autovectores reales bien condicionadas.

3.3. Muestra

La muestra está conformada por matrices reales bien condicionadas de órdenes 3×3 y 4×4 , con autovalores y autovectores reales, las mismas que son propuestas por el autor.

3.4. Método de investigación

El método utilizado son el método deductivo y lógico de soporte. El método deductivo, consiste en extraer consecuencias lógicas de enunciados dados, esto es, en deducir conclusiones a partir de premisas, aplicando en cada paso una regla de inferencia. Utilizan sistemas axiomáticos: conjunto de enunciados tales que, algunos de ellos, los axiomas (supuestamente verdaderos y no se demuestran), se toman como punto de partida, de estos enunciados se deducen otros, llamados teoremas, mediante la aplicación de reglas de inferencia, los cuales garantizan que, si los axiomas son verdaderos, entonces los

teoremas también lo son. En síntesis, el método deductivo va de lo general a lo particular, es prospectivo y teórico.

Los métodos lógicos de soporte comprenden los métodos analógico, analítico, sintético, sistémico y modelado, que son los métodos por excelencia que más se utilizan en la informática para abordar los fenómenos y resolver situaciones problemáticas (Barchini, 2005).

Una vez establecida el método de investigación se procede a sentar la metodología de la elaboración y diseño de algoritmos siguiendo la metodología de solución de problemas mediante a una computadora.

3.5. Metodología para el desarrollo de algoritmos

Para comprender adecuadamente la metodología para el desarrollo de algoritmos, se requiere definir lo que es un algoritmo. Un algoritmo es una serie ordenada de pasos precisos que produce un resultado y termina en un tiempo finito (Forouzan, 2003). Un algoritmo es una secuencia de pasos computacionales que transforma la entrada en salida asemejándose a una maquina como se muestra en la Figura 3.



Figura 3: Algoritmo como una maquina

De las definiciones dadas se puede deducir que un algoritmo es una función, que por lo general es una función compuesta. Las funciones se construyen o componen de otras funciones más básicas, este proceso de construcción representa la fase de diseño del algoritmo. Siendo la construcción de funciones compuestas un problema matemático, es que se hace necesario adaptar las etapas de solución de problemas matemáticos a la solución de problemas mediante una computadora.

Etapas para la obtención de algoritmos

Los pasos para la elaboración de un algoritmo para la resolución de un problemas se puede dividir en tres etapas importantes, tal como indica (Joyanes, 1996):

- **Etapas de análisis del problema**

El propósito del análisis de un problema es ayudar al programador para llegar a una cierta comprensión de la naturaleza del problema. El problema debe estar bien definido si se desea llegar a una solución satisfactoria. Para poder definir con precisión el problema se requiere que las especificaciones de entrada y salida sean descritas con detalle. Una buena definición del problema, junto con una descripción detallada de las especificaciones de entrada y salida son los requisitos más importantes para llegar a una solución eficaz.

- **Etapas de diseño del algoritmo**

Una computadora no tiene la capacidad para solucionar problemas más que cuando se le proporcionan los sucesivos pasos a realizar. Estos pasos sucesivos que indican las instrucciones a ejecutar por la maquina constituyen, como ya sabemos el algoritmo. La información proporcionada al algoritmo constituye su entrada y la información producida por el algoritmo constituye su salida.

Los problemas complejos se pueden resolver más eficazmente con la computadora cuando se rompen en sub problemas que sean más fáciles de solucionar que el original. Este método o técnica se suele denominar “divide y vencerás” y consiste en dividir el problema complejo en otros más simples.

- **Etapas de codificación, ejecución y verificación del algoritmo**

Una vez que el algoritmo está diseñado y descrito mediante un pseudocódigo se debe pasar a la fase de resolución practica del problema con la computadora, que consiste primeramente en la codificación que consiste la conversión del algoritmo en un lenguaje de programación, ya que el algoritmo escrito en un lenguaje especifico de programación se llama código. Luego, tras codificación del

algoritmo, deberá ejecutarse en una computadora y al final se proceder a la verificación de los resultados.

CAPÍTULO IV

RESULTADOS Y DISCUSIÓN

4.1. Análisis del problema de sensibilidad de autovalores y autovectores reales de matrices reales bien condicionadas

Este primer resultado corresponde al análisis del problema de sensibilidad de autovalores y autovectores reales de matrices reales bien condicionadas mediante el método analítico (exacto). A continuación, se aplica el método analítico a casos particulares de una matriz simétrica y no simétrica para poder observar los problemas que puedan tener en este proceso de análisis de sensibilidad de autovalores y autovectores.

4.1.1. Método analítico para el problema de sensibilidad de autovalores y autovectores de una matriz simétrica

4.1.1.1. Cálculo de autovalores y autovectores

Sea la matriz original y una matriz perturbadora

$$A = \begin{bmatrix} 5.1057932 & 5.640524 & 1.0362838 \\ 5.640524 & 9.4466014 & -1.6094262 \\ 1.0362838 & -1.6094262 & 14.447605 \end{bmatrix} \text{ y } E = \begin{bmatrix} 0.001 & 0.001 & 0.002 \\ 0.001 & 0 & 0.001 \\ 0.002 & 0.001 & 0.002 \end{bmatrix}$$

cuyo polinomio característico de la matriz A está dado por $P(\lambda) = \lambda^3 - 29\lambda^2 + 223\lambda - 195$, utilizando formulas con radicales para hallar las solución de la ecuación $P(\lambda) = 0$, se tiene los autovalores de A dadas por $\lambda_1 = 15$, $\lambda_2 = 13$ y $\lambda_3 = 1$. Cuyos autovectores respectivos son:

$$V_1 = \begin{bmatrix} 0.1305777 \\ 0.396035 \\ -0.9089036 \end{bmatrix}, V_2 = \begin{bmatrix} 0.567677 \\ 0.7217334 \\ 0.396035 \end{bmatrix}, V_3 = \begin{bmatrix} 0.8128298 \\ -0.567677 \\ -0.1305776 \end{bmatrix}.$$

Ahora hallamos los autovalores y autovectores de

$$A + E = \begin{bmatrix} 5.1067932 & 5.641524 & 1.0382838 \\ 5.641524 & 9.4466014 & -1.6074262 \\ 1.0382838 & -1.6084262 & 14.449605 \end{bmatrix}$$

su polinomio característico está dado por: $\bar{P}(\lambda) = \lambda^3 - 29.003\lambda^2 + 223.042\lambda - 194.967$,

utilizando las formulas con radicales para hallar las solución de la ecuación

$\bar{P}(\lambda) = 0$, se tiene los autovalores de $A + E$, $\bar{\lambda}_1 = 15.00022$, $\bar{\lambda}_2 = 13.00321$ y

$\bar{\lambda}_3 = 0.9995697$. Sus autovectores respectivos son:

$$\bar{V}_1 = \begin{bmatrix} 0.1298959 \\ 0.3951612 \\ -0.9093815 \end{bmatrix}, \bar{V}_2 = \begin{bmatrix} 0.5678201 \\ 0.7220229 \\ 0.3953015 \end{bmatrix}, \bar{V}_3 = \begin{bmatrix} 0.8128012 \\ -0.567705 \\ -0.1306339 \end{bmatrix}$$

4.1.1.2. Cálculo de cotas de sensibilidad

Calculamos primeramente la cota 1 correspondiente a los autovalores, para los cual calculamos

$$E^t E = \begin{bmatrix} 0.000006 & 0.000003 & 0.000008 \\ 0.000003 & 0.000002 & 0.000004 \\ 0.000008 & 0.000004 & 0.000012 \end{bmatrix},$$

hallamos los autovalores mediante el polinomio característica $\det(E^t E - \lambda I) = 0$,

utilizando las fórmulas de Cardano-Tartaglia para hallar los autovalores

$\lambda_1 = 0.0000003$, $\lambda_2 = 0.0000007$ y $\lambda_3 = 0.000019$. La cota 1 está dado por la raíz

cuadrada del autovalor mayor

$$\|E\|_2 = \sqrt{0.000019} = 0.0043602$$

Ahora, hallamos las cotas 2, correspondiente a cada autovalor.

Para el autovalor $\lambda_1 = 15$ y autovector $V_1 = \begin{bmatrix} 0.1305777 \\ 0.396035 \\ -0.9089036 \end{bmatrix}$, tenemos que la matriz de

paso formado por sus autovectores:

$$Q = [V_1 \ V_2 \ V_3] = \begin{bmatrix} 0.1305777 & 0.567677 & 0.8128298 \\ 0.396035 & 0.7217334 & -0.567677 \\ -0.9089036 & 0.396035 & -0.1305776 \end{bmatrix}$$

$$Q'EQ = \begin{bmatrix} 0.0002181 & -0.0015975 & -0.0005078 \\ -0.0024104 & 0.0032121 & 0.0007046 \\ 0.0000598 & 0.0005741 & -0.0004302 \end{bmatrix}$$

luego obtenemos la siguiente sub matriz:

$$e_1 = \begin{bmatrix} -0.0024104 \\ 0.0000598 \end{bmatrix},$$

esto implica que $\|e_1\|_2 = 0.0024111$. Además, $d_1 = \min_{\mu \in \lambda(D_2)} |\lambda_1 - \mu| = 1.9999997$.

Por lo tanto, la cota de la distancia del subespacio invariante relativo al autovalor λ_1 está dado por:

$$\frac{4}{d_1} \|e_1\|_2 = 0.0048222$$

Para el autovalor $\lambda_2 = 13$ y autovector $V_2 = \begin{bmatrix} 0.567677 \\ 0.7217334 \\ 0.396035 \end{bmatrix}$, haciendo un cambio en

el orden de los autovectores en Q , se tiene

$$Q = [V_2 \ V_3 \ V_1] = \begin{bmatrix} 0.567677 & 0.8128298 & 0.1305777 \\ 0.7217334 & -0.567677 & 0.396035 \\ 0.396035 & -0.1305776 & -0.9089036 \end{bmatrix}$$

luego tenemos que

$$Q'EQ = \begin{bmatrix} 0.0032121 & 0.0007046 & -0.0024104 \\ 0.0005741 & -0.0004302 & 0.0000598 \\ -0.0015975 & -0.0005078 & 0.0002181 \end{bmatrix},$$

luego obtenemos la siguiente sub matriz:

$$e_2 = \begin{bmatrix} 0.0005741 \\ -0.0015975 \end{bmatrix}$$

esto implica que $\|e_2\|_2 = 0.0016975$. Además, $d_2 = \min_{\mu \in \lambda(D_2)} |\lambda_2 - \mu| = 1.9999997$.

Por lo tanto, la cota de la distancia del subespacio invariante relativo al autovalor λ_2 está dado por:

$$\frac{4}{d_2} \|e_2\|_2 = 0.0033951$$

Para el autovalor $\lambda_3 = 1$ y autovector $V_3 = \begin{bmatrix} 0.8128298 \\ -0.567677 \\ -0.1305776 \end{bmatrix}$, haciendo un cambio en

el orden de los autovectores en Q , se tiene

$$Q = [V_3 \quad V_1 \quad V_2] = \begin{bmatrix} 0.8128298 & 0.1305777 & 0.567677 \\ -0.567677 & 0.396035 & 0.7217334 \\ -0.1305776 & -0.9089036 & 0.396035 \end{bmatrix},$$

luego, tenemos que

$$Q'EQ = \begin{bmatrix} -0.0004302 & 0.0000598 & 0.0005741 \\ -0.0005078 & 0.0002181 & -0.0015975 \\ 0.0007046 & -0.0024104 & 0.0032121 \end{bmatrix}$$

luego, obtenemos la siguiente sub matriz:

$$e_3 = \begin{bmatrix} -0.0005078 \\ 0.0007046 \end{bmatrix}$$

esto implica que $\|e_3\|_2 = 0.0008686$. Además, $d_3 = \min_{\mu \in \lambda(D_2)} |\lambda_1 - \mu| = 12$.

Por lo tanto, la cota de la distancia del subespacio invariante relativo al autovalor λ_3 está dado por:

$$\frac{4}{d_3} \|e_3\|_2 = 0.0002895$$

A continuación, se resumen los resultados hallados por los métodos analíticos en las Tabla 1 y Tabla 2.

Tabla 1

Cota 1 para los autovalores de la matriz simétrica

| AUTOVALORES | COTA 1 |
|------------------|-----------|
| $\lambda_1 = 15$ | 0.0043602 |
| $\lambda_2 = 13$ | 0.0043602 |
| $\lambda_3 = 1$ | 0.0043602 |

A

Tabla 2

Cotas 2 correspondientes a cada autovector de la matriz simétrica A

| AUTOVECTORES | COTA 2 |
|--|-----------|
| $V_1 = \begin{bmatrix} 0.1305777 \\ 0.396035 \\ -0.9089036 \end{bmatrix}$ | 0.0048222 |
| $V_2 = \begin{bmatrix} 0.567677 \\ 0.7217334 \\ 0.396035 \end{bmatrix}$ | 0.0033951 |
| $V_3 = \begin{bmatrix} 0.8128298 \\ -0.567677 \\ -0.1305776 \end{bmatrix}$ | 0.0002895 |

4.1.1.3. Análisis de sensibilidad

La Cota 1 nos indica dos cosas, primero nos da una cota de todas las distancias entre los autovalores de la matriz original y los autovalores de la matriz perturbada y segundo nos indica el grado de sensibilidad de los autovalores. Como la Cota 1 depende de la norma de la matriz perturbadora y del número

de condición de A es 1 (bien condicionada), se tiene, que pequeñas perturbaciones producen pequeños cambios en los autovalores. A continuación, se muestra la Tabla 3 donde se verifica las distancias con respecto a la Cota 1.

Tabla 3

Resultados para el análisis de sensibilidad de autovalores del caso simétrico.

| AUTOVALORES DE LA MATRIZ SIMÉTRICO ORIGINAL | AUTOVALORES DE LA MATRIZ SIMÉTRICA PERTURBADA | DISTANCIA ENTRE LOS AUTOVALORES | COTA1 |
|---|---|---------------------------------|-----------|
| $\lambda_1 = 15$ | $\bar{\lambda}_1 = 15.00022$ | 0.00022 | 0.0043602 |
| $\lambda_2 = 13$ | $\bar{\lambda}_2 = 13.00321$ | 0.0032102 | 0.0043602 |
| $\lambda_3 = 1$ | $\bar{\lambda}_3 = 0.9995697$ | 0.0004303 | 0.0043602 |

La Tabla 3 nos indica que las distancias entre los autovalores $|\bar{\lambda}_i - \lambda_i|$ son menores que la cota 0.0043602. Además, dado que la cota 1 depende de la norma de E y del número de condicionamiento A que es 1, se tiene, que pequeños cambios en la matriz original generan pequeños cambios en los autovalores.

Las Cotas 2 nos dan dos tipos de información, la primera es que son cotas para las distancias de los subespacios invariantes generados por los autovectores de la matriz original y la matriz perturbada respectivamente, la segunda está referida al grado de sensibilidad de los autovectores que según el marco teórico está determinado por la cercanía o lejanía de sus autovalores respectivos. Siendo más sensibles lo aquellos autovectores que sus autovalores son más cercanos.

A continuación, se muestra la Tabla 4 en donde se muestran las distancias de los subespacios invariantes generados por los autovectores entre la matriz original y la matriz perturbada, conjuntamente con la Cotas 2.

Tabla 4

Resultados para el análisis de sensibilidad de los autovectores del caso simétrico.

| AUTOVECTORES DE LA MATRIZ SIMÉTRICA ORIGINAL | AUTOVECTORES DE LA MATRIZ SIMÉTRICA PERTURBADO | DISTANCIAS ENTRE LOS SUBESPACIOS INVARIANTES | COTA2 |
|--|--|--|-----------|
| $V_1 = \begin{bmatrix} 0.1305777 \\ 0.396035 \\ -0.9089036 \end{bmatrix}$ | $\bar{V}_1 = \begin{bmatrix} 0.1298959 \\ 0.3951612 \\ -0.9093815 \end{bmatrix}$ | 0.001207 | 0.0048222 |
| $V_2 = \begin{bmatrix} 0.567677 \\ 0.7217334 \\ 0.396035 \end{bmatrix}$ | $\bar{V}_2 = \begin{bmatrix} 0.5678201 \\ 0.7220229 \\ 0.3953015 \end{bmatrix}$ | 0.0008014 | 0.0033951 |
| $V_3 = \begin{bmatrix} 0.8128298 \\ -0.567677 \\ -0.1305776 \end{bmatrix}$ | $\bar{V}_3 = \begin{bmatrix} 0.8128012 \\ -0.567705 \\ -0.1306339 \end{bmatrix}$ | 0.000069 | 0.0002895 |

En la Tabla 4 se muestra las Cotas 2 de las distancias de los subespacios invariantes de la matriz original y de su matriz perturbada respectiva, la cota para el primer autovector es 0.0048222, para el segundo autovector 0.0033951 y finalmente para el tercer autovector 0.0002895. Además, se observa que la cota correspondiente al tercer autovector es más pequeña, esto significa que es menos sensible. Este hecho es coherente con el marco teórico, ya que su autovalor correspondiente es el más aislado.

4.1.2. Método analítico para el problema de sensibilidad de autovalores y autovectores de una matriz no simétrica

4.1.2.1. Cálculo de autovalores y autovectores

Sea la matriz original no simétrica

$$A = \begin{bmatrix} 2.2 & -8.8 & 15.2 \\ -12 & 5 & 16 \\ -7.2 & -3.2 & 21.8 \end{bmatrix} \text{ y } E = \begin{bmatrix} 0.001 & 0.001 & 0 \\ 0.003 & 0 & 0 \\ 0.002 & 0 & 0.002 \end{bmatrix}$$

Cuyo polinomio característico está dado por $P(\lambda) = \lambda^3 - 29\lambda^2 + 223\lambda - 195$. Utilizando las fórmulas con radicales para hallar las solución de la ecuación $P(\lambda) = 0$, se tiene los autovalores de A $\lambda_1 = 15$, $\lambda_2 = 13$, $\lambda_3 = 1$. Sus respectivos autovectores los calculados mediante el sistema de ecuaciones $(A - \lambda_i I)V_i = 0$, luego:

$$.V_1 = \begin{bmatrix} 0.3333333 \\ 0.6666667 \\ 0.6666667 \end{bmatrix}, V_2 = \begin{bmatrix} 0.6666667 \\ 0.3333333 \\ 0.6666667 \end{bmatrix}, V_3 = \begin{bmatrix} 0.6666667 \\ 0.6666667 \\ 0.3333333 \end{bmatrix}$$

Ahora hallamos los autovalores y autovectores de

$$A + E = \begin{bmatrix} 2.201 & -8.799 & 15.2 \\ -11.997 & 5 & 16 \\ -7.198 & -3.2 & 21.802 \end{bmatrix}$$

Su polinomio característico está dado por $\bar{P}(\lambda) = \lambda^3 - 29.003\lambda^2 + 223.049\lambda - 195.11$. Utilizando las formulas con radicales para hallar las solución de la ecuación $\bar{P}(\lambda) = 0$, se tiene los autovalores de $A + E$, $\bar{\lambda}_1 = 15.001803$, $\bar{\lambda}_2 = 13.000797$, $\bar{\lambda}_3 = 1.0004002$. Sus autovectores los calculados mediante el sistema de ecuaciones $(A - \bar{\lambda}_i I)\bar{V}_i = 0$, luego:

$$\vec{V}_1 = \begin{bmatrix} 0.3336345 \\ 0.6664272 \\ 0.6667554 \end{bmatrix}, \vec{V}_2 = \begin{bmatrix} 0.6671522 \\ 0.3325916 \\ 0.6665514 \end{bmatrix}, \vec{V}_3 = \begin{bmatrix} 0.6667086 \\ 0.6666626 \\ 0.3332575 \end{bmatrix}$$

4.1.2.2. Cálculos de cotas de sensibilidad

Calculamos primeramente la cota 1 correspondiente a los autovalores, para lo cual se tiene:

$$E^t E = \begin{bmatrix} 0.000014 & 0.000001 & 0.000004 \\ 0.000001 & 0.000001 & 0 \\ 0.000004 & 0 & 0.000004 \end{bmatrix}$$

Luego, hallamos los autovalores $E^t E$ mediante la solución de la ecuación característica $\det(E^t E - \lambda I) = 0$, utilizando las fórmulas de Cardano-Tartaglia tenemos $\lambda_1 = 0.0000155$, $\lambda_2 = 0.0000027$ y $\lambda_3 = 0.0000009$. Para determinar la cota 1 se calcula la raíz cuadrada del mayor autovalor

$$\|E\|_2 = \sqrt{0.0000155} = 0.0039325$$

Construimos una matriz Q formada por las columnas que son los autovectores de la matriz no simétrica A . Luego, calculamos número de condición de A para el problema de autovalores

$$Q = \begin{bmatrix} 0.3333333 & 0.6666667 & 0.6666667 \\ 0.6666667 & 0.3333333 & 0.6666667 \\ 0.6666667 & 0.6666667 & 0.3333333 \end{bmatrix}$$

$$Q^{-1} = \begin{bmatrix} -1.8 & 1.2 & 1.2 \\ 1.2 & -1.8 & 1.2 \\ 1.2 & 1.2 & -1.8 \end{bmatrix}$$

$$\Gamma_2(A) = \kappa_2(Q) = \|Q\|_2 \|Q^{-1}\|_2 = 5$$

Por lo tanto, se tiene

$$\kappa_2(Q)\|E\|_2 = 5(0.0039325) = 0.0196625$$

Ahora, hallamos las cotas 2 de manera separada para cada autovalor y su autovector asociado.

Para el autovalor $\lambda_1 = 15$ y autovector $V_1 = \begin{bmatrix} 0.3333333 \\ 0.6666667 \\ 0.6666667 \end{bmatrix}$, primeramente necesitamos

la matriz de autovectores

$$Q = [V_1 \ V_2 \ V_3] = \begin{bmatrix} 0.3333333 & 0.6666667 & 0.6666667 \\ 0.6666667 & 0.3333333 & 0.6666667 \\ 0.6666667 & 0.6666667 & 0.3333333 \end{bmatrix}$$

Ortogonalizamos los vectores que forman Q mediante Gram Schmidt y se obtiene

$$q = \begin{bmatrix} 0.3333333 & 0.8084521 & 0.4850713 \\ 0.6666667 & -0.5659165 & 0.4850713 \\ 0.6666667 & 0.1616904 & -0.7276069 \end{bmatrix}$$

$$q^t A q = \begin{bmatrix} 15 & -3.88057 & -28.716218 \\ 0 & 13 & -6.4 \\ 0 & 0 & 1 \end{bmatrix}$$

$$q^t E q = \begin{bmatrix} 0.0023333 & 0.0029913 & 0.0009701 \\ 0.0005659 & -0.0008627 & -0.0001176 \\ -0.0004851 & -0.0001176 & 0.0015294 \end{bmatrix}$$

De estas matrices obtenemos las siguientes sub matrices

$$T_{22} = \begin{bmatrix} 13 & -6.4 \\ 0 & 1 \end{bmatrix} \quad \delta_1 = \begin{bmatrix} 0.0005659 \\ -0.0004851 \end{bmatrix}$$

Hallamos el mínimo valor singular de $T_{22} - \lambda I$, para ello hacemos

$$\sigma_1 = \sigma_{\min}(T_{22} - \lambda I) = 1.8162637$$

$$\|\delta_1\|_2 = 0.0007454$$

Por lo tanto, la cota de la distancia del subespacio invariante relativo al autovalor λ_1 está dado por:

$$\frac{4}{\sigma_1} \|\delta_1\|_2 = 0.0016415$$

Para el autovalor $\lambda_2 = 13$ y autovector $V_2 = \begin{bmatrix} 0.6666667 \\ 0.3333333 \\ 0.6666667 \end{bmatrix}$, primeramente necesitamos la matriz de autovectores

$$Q = [V_2 \quad V_3 \quad V_1] = \begin{bmatrix} 0.6666667 & 0.6666667 & 0.3333333 \\ 0.3333333 & 0.6666667 & 0.6666667 \\ 0.6666667 & 0.3333333 & 0.6666667 \end{bmatrix}$$

Ortogonalizamos los vectores que forman Q mediante Gram Schmidt y se obtiene

$$q = \begin{bmatrix} 0.6666667 & 0.1616904 & -0.7276069 \\ 0.3333333 & 0.8084521 & 0.4850713 \\ 0.6666667 & -0.5659165 & 0.4850713 \end{bmatrix}$$

$$q^t A q = \begin{bmatrix} 13 & -23.28342 & 16.815803 \\ 0 & 1 & 7.4666667 \\ 0 & 0 & 15 \end{bmatrix}$$

$$q^t E q = \begin{bmatrix} 0.0031111 & 0.0002695 & -0.0012127 \\ 0.0002695 & 0.0010065 & -0.0015294 \\ 0.0015361 & -0.0008627 & -0.0011176 \end{bmatrix}$$

De estas matrices obtenemos las siguientes sub matrices

$$T_{22} = \begin{bmatrix} 1 & 7.46666667 \\ 0 & 15 \end{bmatrix} \quad \delta_2 = \begin{bmatrix} 0.0002695 \\ 0.0015361 \end{bmatrix}$$

Hallamos el mínimo valor singular de $T_{22} - \lambda I$, para ello hacemos

$$\sigma_2 = \sigma_{\min}(T_{22} - \lambda I) = 0.769789$$

$$\|\delta_2\|_2 = 0.0029936$$

Por lo tanto, la cota de la distancia del subespacio invariante relativo al autovalor λ_2 está dado por:

$$\frac{4}{\sigma_2} \|\delta_2\|_2 = 0.0155554$$

Para el autovalor $\lambda_3 = 1$ y autovector $V_3 = \begin{bmatrix} 0.6666667 \\ 0.6666667 \\ 0.3333333 \end{bmatrix}$, primeramente necesitamos

la matriz de autovectores

$$Q = [V_3 \quad V_1 \quad V_2] = \begin{bmatrix} 0.6666667 & 0.3333333 & 0.6666667 \\ 0.6666667 & 0.6666667 & 0.3333333 \\ 0.3333333 & 0.6666667 & 0.6666667 \end{bmatrix}$$

Ortogonalizamos los vectores que forman Q mediante Gram Schmidt y se obtiene

$$q = \begin{bmatrix} 0.6666667 & -0.5659165 & 0.4850713 \\ 0.6666667 & 0.1616904 & -0.7276069 \\ 0.3333333 & 0.8084521 & 0.4850713 \end{bmatrix}$$

$$q^t A q = \begin{bmatrix} 1 & 27.16399 & 11.900415 \\ 0 & 15 & -1.0666667 \\ 0 & 0 & 13 \end{bmatrix}$$

$$q^t E q = \begin{bmatrix} 0.0028889 & -0.0012396 & 0.0014552 \\ 0.0011857 & 0.0003464 & 0.0019412 \\ 0.0001617 & 0.0012745 & -0.0002353 \end{bmatrix}$$

De estas matrices obtenemos las siguientes sub matrices

$$T_{22} = \begin{bmatrix} 15 & -1.0666667 \\ 0 & 13 \end{bmatrix} \quad \delta_3 = \begin{bmatrix} 0.0011857 \\ 0.0001617 \end{bmatrix}$$

Hallamos el mínimo valor singular de $T_{22} - \lambda I$, para ello hacemos

$$\sigma_3 = \sigma_{\min}(T_{22} - \lambda I) = 10.961178$$

$$\|\delta_3\|_2 = 0.0009772$$

Por lo tanto, la cota de la distancia del subespacio invariante relativo al autovalor λ_3 está dado por:

$$\frac{4}{\sigma_3} \|\delta_3\|_2 = 0.0003566$$

A continuación, se resumen los resultados hallados por los métodos analíticos en las siguientes tablas:

Tabla 5
Cota 1 para los autovalores de la matriz no simétrica A

| AUTOVALORES | COTA 1 |
|------------------|-----------|
| $\lambda_1 = 15$ | 0.0196625 |
| $\lambda_2 = 13$ | 0.0196625 |
| $\lambda_3 = 1$ | 0.0196625 |

Tabla 6
Cotas 2 correspondientes a cada autovector de la matriz no simétrica A

| AUTOVECTORES | COTA 2 |
|---|-----------|
| $V_1 = \begin{bmatrix} 0.3333333 \\ 0.6666667 \\ 0.6666667 \end{bmatrix}$ | 0.0016415 |
| $V_2 = \begin{bmatrix} 0.6666667 \\ 0.3333333 \\ 0.6666667 \end{bmatrix}$ | 0.0155554 |
| $V_3 = \begin{bmatrix} 0.6666667 \\ 0.6666667 \\ 0.3333333 \end{bmatrix}$ | 0.0003566 |

4.1.2.3. Análisis de sensibilidad

La Cota 1 nos indica dos cosas, primero nos da una cota de todas las distancias entre los autovalores de la matriz original y sus correspondientes autovalores de la matriz perturbada y segundo nos indica el grado de sensibilidad de los autovalores, ya que como depende de la norma de la matriz perturbadora y del número de condición $\Gamma_2(A)$, el grado de sensibilidad depende del valor de $\Gamma_2(A)$, es decir, si $\Gamma_2(A)$ resulta ser pequeño cercano a 1 significa que pequeñas perturbaciones producen pequeños cambios en el valor de los autovalores y si $\Gamma_2(A)$ es un valor muy grande esto implica que está mal condicionado, esto es, que una pequeña perturbación de A producirá grandes cambios en los autovalores. A continuación, se muestra un cuadro donde se verifica las distancias de los autovalores con respecto a la Cota 1.

Tabla 7
Resultados para el análisis de sensibilidad de autovalores de la matriz no simétrica A

| AUTOVALORES DE LA MATRIZ SIMÉTRICO ORIGINAL | AUTOVALORES DE LA MATRIZ SIMÉTRICA PERTURBADA | DISTANCIA ENTRE LOS AUTOVALORES | COTA1 |
|---|---|---------------------------------|-----------|
| $\lambda_1 = 15$ | $\bar{\lambda}_1 = 15.001803$ | 0.0018032 | 0.0196625 |
| $\lambda_2 = 13$ | $\bar{\lambda}_2 = 13.000797$ | 0.0007965 | 0.0196625 |
| $\lambda_3 = 1$ | $\bar{\lambda}_3 = 1.0004002$ | 0.0004002 | 0.0196625 |

La Tabla 7 nos indica que las distancias entre los autovalores $|\bar{\lambda}_i - \lambda_i|$ son menores que la Cota 1 dada por 0.0196625 . Además, dado que la Cota 1 depende de la norma de E y del número de condición de A que en este caso es $\Gamma_2(A) = \|Q\|_2 \|Q^{-1}\|_2 = 5$ que significa que está bien condicionado, se tiene que, pequeños cambios en la matriz original generan pequeños cambios en los autovalores.

Las Cotas 2 nos dan información de dos tipos, la primera es que son cotas para las distancias de los subespacios invariantes generados por los autovectores de la matriz original y su matriz perturbada correspondiente, la segunda está referida al grado de sensibilidad de los autovectores que según el marco teórico está determinado por la cercanía o lejanía de sus autovalores respectivos. Siendo más sensibles lo aquellos autovectores que sus autovalores son más cercanos.

A continuación, se muestra la Tabla 8 en donde se muestran las distancias de los subespacios invariantes generados por los autovectores entre la matriz original y la matriz perturbada, conjuntamente con la Cota 2.

Tabla 8

Resultados para el análisis de sensibilidad de los autovectores de la matriz no simétrica A

| AUTOVECTORES DE LA MATRIZ ORIGINAL | AUTOVECTORES DE LA MATRIZ PERTURBADO | DISTANCIAS ENTRE LOS SUBESPACIOS INVARIANTES | COTA 2 |
|---|---|--|-----------|
| $V_1 = \begin{bmatrix} 0.3333333 \\ 0.6666667 \\ 0.6666667 \end{bmatrix}$ | $\bar{V}_1 = \begin{bmatrix} 0.3336345 \\ 0.6664272 \\ 0.6667554 \end{bmatrix}$ | 0.0003948 | 0.0016415 |
| $V_2 = \begin{bmatrix} 0.6666667 \\ 0.3333333 \\ 0.6666667 \end{bmatrix}$ | $\bar{V}_2 = \begin{bmatrix} 0.6671522 \\ 0.3325916 \\ 0.6665514 \end{bmatrix}$ | 0.000894 | 0.0155554 |
| $V_3 = \begin{bmatrix} 0.6666667 \\ 0.6666667 \\ 0.3333333 \end{bmatrix}$ | $\bar{V}_3 = \begin{bmatrix} 0.6667086 \\ 0.6666626 \\ 0.3332575 \end{bmatrix}$ | 0.0000868 | 0.0003566 |

En la Tabla 8 se muestra las cotas de las distancias de los subespacios invariantes de la matriz original y de su matriz perturbada respectiva, la cota para el primer autovector es 0.0016415 , para el segundo autovector 0.0155554 y finalmente para el tercer autovector 0.0003566 . Además, se observa que la cota correspondiente al tercer autovector es más pequeña, esto significa que es menos sensible. Este hecho es coherente con el marco teórico, ya que su autovalor correspondiente es el más aislados.

4.1.3. Resultado

Se analizó el problema de sensibilidad de autovalores y autovectores reales de matrices reales bien condicionadas. En este análisis se identificó como entrada del problema a las matrices cuadradas de orden arbitrario con entradas reales que tenga números condición pequeña (cercano a uno) y otra matriz de igual orden con entradas reales que tiene norma dos pequeña denominada matriz perturbadora. En cuanto a la salida se identificó a un número real denominado Cota 1 y una matriz columna con entras reales denominado Cotas 2. Para obtener estas salidas a partir de las matrices de entrada, se tiene que seguir un proceso determinado por los resultados matemáticos dados por los Teoremas 1 y 2 para el caso de matrices simétricas y los y Teoremas 3 y 4 para el caso de matrices no simétricas. La Cota 1 es solo depende de la norma dos de la matriz perturbadora para el caso de la matriz simétrica y depende además de la norma dos de la matriz perturbadora del número de condición de la matriz analizada. La Cota 2 depende de una submatriz semejante a la matriz perturbadora tanto para matrices simétricas como para no simétricas.

4.1.4. Discusión

El hecho de considerar solo 2 casos particulares como consecuencia de resultados matemáticos derivados del Teorema 1 y del Teorema 2 para el caso simétrico y los Teorema 3 y Teorema 4 para el caso no simétrico, es suficiente para llegar a la conclusión indicada. Cabe mencionar que en los textos donde se encuentran estos resultados como Golub (1996) y Stewart (1998) se considera pocos ejemplos detallados de los resultados mencionados, sin embargo estos son suficientes para comprender sus alcances . Estas conclusiones alcanzadas pueden servir de referencia a otros investigadores para proponer soluciones innovadoras al problema. También,

este resultado se puede utilizar en casos aplicativos, bastaría en darle una interpretación a la matriz de entra y a la matriz perturbadora.

4.2. Diseño del algoritmo basado en el método QR para analizar la sensibilidad de autovalores y autovectores

4.2.1. Método QR como método base

El Algoritmo QR de 1961 basado en el método QR descubierta de forma independiente por el informático inglés John GF Francis y por la matemática rusa Vera N. Kublanovscaya, tuvo como antecesor al método LR del matemático Heinz Rutishauser que utiliza descomposiciones triangulares, crea una secuencia de matrices semejantes que bajo ciertas condiciones, convergen a una matriz triangular superior o diagonal cuyos elementos de la diagonal principal forman los autovalores (Oliveira, 2006).

El algoritmo LR tiene una aplicación restringida, pues depende del hecho de como sea la matriz dada, esta tiene que ser descomponible por el método LU, donde L es una matriz triangular inferior con todos los elementos de la diagonal principal igual a 1 y U una matriz triangular superior.

El método QR actualmente es considerado como uno de los métodos más poderosos para hallar autovalores, su método modifica el algoritmo LR en cuanto a la descomposición de cada matriz de la secuencia. En vez de la descomposición triangular (LU), utiliza la descomposición ortogonal-triangular (QR), donde Q es una matriz ortogonal y R es una matriz triangular superior.

En la actualidad existen diferentes métodos para poder hallar los autovalores de una matriz, como el método Leverrier-Laguerre que consiste en encontrar todas las raíces de la ecuación característica que resultan ser poco estables, ya que unas perturbaciones de sus coeficientes generan en algunos casos raíces muy distintas (Badia Contelles & Vidal Maciá, 2000). Otro método muy conocido es el método de Jacobi se muestra como un algoritmo eficiente, pero con la restricción de que solo funcionan para matrices simétricas.

A continuación, se hace una comparación del método QR con los métodos de Jacobi y el método de Leverrier-Laguerre, para ello consideramos la matriz

$$A = \begin{bmatrix} 8 & -1 & 3 & -1 \\ -1 & 6 & 2 & 0 \\ 3 & 2 & 9 & 1 \\ -1 & 0 & 1 & 7 \end{bmatrix}$$

Se obtuvieron los siguientes resultados:

Tabla 9

De autovalores de los métodos analíticos, QR, Jacobi y Leverrier-Laguerre

| MÉTODO ANALÍTICO (λ) | MÉTODO QR (λ_{QR}) | MÉTODO JACOBI (λ_{Jacobi}) | MÉTODO LEVERRIER-LAGUERRE (λ_{LL}) |
|-----------------------------------|---------------------------------|---|---|
| 3.29569865813870422 | 3.29569865813874507 | 3.295698658144097 | 3.295697519717757 |
| 6.5923380437500958 | 6.59233804374996435 | 6.592338043758817 | 6.592343316943876 |
| 8.40766195624992996 | 8.40766195625003654 | 8.407661956243984 | 8.407656683057276 |
| 11.7043013418612674 | 11.704301341861255 | 11.704301341853059 | 11.704302480281093 |

Los resultados de la Tabla 9 se obtuvieron utilizando los algoritmos mostrados en el marco teórico. A continuación, se muestra las distancias de los autovalores obtenidos con relación a los autovalores exactos dados por el método analítico:

Tabla 10

Distancias los autovalores obtenidos por los otros métodos

| DISTANCIA 1 $ \lambda - \lambda_{QR} $ | DISTANCIA 2 $ \lambda - \lambda_{Jacobi} $ | DISTANCIA 3 $ \lambda - \lambda_{LL} $ |
|---|---|---|
| 3.9968E-14 | 5.38991E-12 | 1.13842E-06 |
| 1.29674E-13 | 8.72014E-12 | 5.27319E-06 |
| 1.10134E-13 | 5.94014E-12 | 5.27319E-06 |
| 0 | 8.19966E-12 | 1.13842E-06 |

De la Tabla 10 se deduce que el método QR ofrece autovalores más cercanos a los valores exactos dado por el método analítico.

De los párrafos anteriores se deduce que el método QR es un método adecuado para determinar la sensibilidad de autovalores y autovectores reales de matrices simétricas y no simétricas bien condicionadas.

4.2.2. Algoritmos básicos para el algoritmo propuesto

Una vez establecido el método QR como base para el algoritmo propuesto, ahora se fijará el diseño “divide y vencerás” para elaborarlo, por el hecho de que el algoritmo propuesto está compuesto por otros algoritmos o funciones.

El algoritmo propuesto basado en el método QR está compuesta por los algoritmos internos correspondientes al lenguaje de programación de Scilab y por los algoritmos externos dados en las siguientes subsecciones.

4.2.2.1. Algoritmo de Householder

Este algoritmo transforma una matriz simétrica A en una matriz tridiagonal simétrica B , este algoritmo es fundamental ya que el algoritmo QR simétrico, solo halla autovalores de matrices tridiagonales simétricas.

```

1: Function B=HOUSE(A)
2: n: Orden de la matriz A.
3: For k = 1, 2, ..., n - 2 do
4:    $q = \sum_{j=k+1}^n (a_{jk}^{(k)})^2$ 
5:   If  $a_{k+1,k}^{(k)} = 0$  then
6:      $\alpha = -q^{1/2}$ 
7:   Else
8:      $\alpha = -\frac{q^{1/2} a_{k+1,k}^{(k)}}{|a_{k+1,k}^{(k)}|}$ 
9:   EndIf
10:   $RSQ = \alpha^2 - \alpha a_{k+1,k}^{(k)}, v_k = 0, v_{k+1} = a_{k+1,k}^{(k)} - \alpha$ 
11:  For j = k + 2, ..., n do
12:     $v_j = a_{jk}^{(k)}$ 
13:  EndFor
14:  For j = k, k + 1, ..., n do

```

```

15:      $u_j = \left(\frac{1}{RSQ}\right) \sum_{i=k+1}^n a_{ji}^{(k)} v_i$ 
16: EndFor
17:      $PROD = \sum_{i=k+1}^n v_i u_i$ 
18: For  $j = k, k+1, \dots, n$  do
19:      $z_j = u_j - \left(\frac{PROD}{2RSQ}\right) v_j$ 
20: EndFor
21: For  $l = k+1, k+2, \dots, n-1$  do
22:     For  $j = l+1, \dots, n$  do
23:          $a_{jl}^{(k+1)} = a_{jl}^{(k)} - v_l z_j - v_j z_l, a_{jl}^{(k+1)} = a_{lj}^{(k+1)}$ 
24:     EndFor
25:      $a_{ll}^{(k+1)} = a_{ll}^{(k+1)} = 0$ 
26: EndFor
27:      $a_{nn}^{(k+1)} = a_{nn}^{(k)} - 2v_n z_n$ 
28: For  $j = k+2, \dots, n$  do
29:      $a_{kj}^{(k+1)} = a_{jk}^{(k+1)} = 0$ 
30: EndFor
31:      $a_{k+1,k}^{(k+1)} = a_{k+1,k}^{(k+1)} - v_{k+1} z_k, a_{k+1,k}^{(k+1)} = a_{k+1,k}^{(k+1)}$ 
32: EndFor
33:  $B = A^{(n-1)}$ 
34: EndFunction
    
```

4.2.2.2. Algoritmo QR para matrices simétricas

Sea una matriz tridiagonal simétrica de $n \times n$, dada por:

$$A = \begin{bmatrix} a_1^{(i)} & b_2^{(i)} & 0 & \dots & 0 \\ b_2^{(i)} & a_2^{(i)} & \ddots & \dots & 0 \\ 0 & \ddots & \ddots & \ddots & \vdots \\ \vdots & \vdots & \ddots & \ddots & b_n^{(i)} \\ 0 & 0 & \dots & b_n^{(i)} & a_n^{(i)} \end{bmatrix}$$

El siguiente algoritmo permite encontrar los autovalores de la matriz tridiagonal simétrica A , este algoritmo es importante en la investigación ya que es el algoritmo en el que está basando el algoritmo propuesto.

```

1:  Function LAMBDA=METQR_SIMETRICO(A)
2:  TOL=0.0001, N=200, k=1, SHIFT=0,
3:  n: dimensión de la matriz A
4:  While k ≤ N do
5:    If |b(n)| ≤ TOL then
6:      LAMBDA(n)=a(n)+SHIFT
7:      n=n-1
8:    EndIf
9:     $b = -(a_{n-1}^{(k)} + a_n^{(k)}), c = a_n^{(k)} a_{n-1}^{(k)} - [b_n^{(k)}]^2, d = (b^2 - 4c)^{1/2}$ 
10:    $\mu_1 = (-b+d)/2, \mu_2 = -(b+d)/2$ 
11:   If n=2 then
12:     LAMBDA (1) =  $\mu_1$  + SHIFT, LAMBDA (2) =  $\mu_2$  + SHIFT
13:     SALIDA (LAMBDA (1), LAMBDA (2)), PARE
14:   EndIf
15:   Elija s talque  $|s - a_n^{(k)}| = \min \{ |\mu_1 - a_n^{(k)}|, |\mu_2 - a_n^{(k)}| \}$ 
16:   Tome SHIFT=SHIFT+s
17:   For j = 1, ..., n do
18:      $d_j = a_j^{(k)} - s$ 
19:   EndFor
20:   Tome  $x_1 = d_1, y_1 = b_2$ 
21:   For j = 2, ..., n do
22:      $z_{j-1} = \sqrt{x_{j-1}^2 + (b_j^{(k)})^2}$ 
23:      $c_j = \frac{x_{j-1}}{z_{j-1}}, s_j = \frac{b_j^{(k)}}{z_{j-1}}, q_{j-1} = c_j y_{j-1} + s_j d_j, x_j = -s_j y_{j-1} + c_j d_j$ 
24:     If j ≠ n then
25:        $r_{j-1} = s_j b_{j+1}^{(k)}, y_j = c_j b_{j+1}^{(k)}$ 
26:     EndIf
27:   EndFor
28:    $z_n = x_n, a_1^{(k+1)} = s_2 q_1 + c_2 z_1, b_2^{(k+1)} = s_2 z_2$ 
29:   For j = 2, 3, ..., n-1 do
30:      $a_n^{(k+1)} = s_{j+1} q_j + c_{j+1} z_j, b_2^{(k+1)} = s_{j+1} z_{j+1}$ 
31:   EndFor
32:    $a_n^{(k+1)} = c_n z_n, k = k + 1$ 
33: EndWhile
34: SALIDA (Número máximo de iteraciones excedido)

```

- 35: PARE
- 36: EndFunction

4.2.2.3. Algoritmo QR Schur para matrices no simétricas

Este algoritmo permite encontrar la descomposición de Schur a partir de una matriz no simétrica. Esta descomposición está dada por $A=QR$, donde Q es una matriz ortogonal y R es una matriz triangular superior.

- 1: Function [Q,R]=METQR_SCHUR(A)
- 2: n: orden de la matriz A
- 3: $Q = I_n$
- 4: $i = 1$
- 5: While $i \leq 50$ do
- 6: [q,r]=FACTOR_QR(A)
- 7: $A = rq$;
- 8: $Q = Qq$;
- 9: $i = i + 1$;
- 10: end
- 11: $R = A$
- 12: EndFunction

4.2.2.4. Algoritmo de potencia inversa con traslación

La entrada del algoritmo es una matriz simétrica cuadrada y un vector LAMBDA que contiene una buena aproximación de sus autovalores y se obtendrá dos matrices Q y R conteniendo los autovectores como columnas en Q y sus autovalores en la diagonal de R .

- 1: Function [Q, R]=INVPOWTRAS(A,LAMBDA)
- 2: n: orden de la matriz A
- 3: $max1 = 100$
- 4: $\varepsilon = 0.001$
- 5: For $i = 1, \dots, n$ do
- 6: $alpha(i) = LAMBDA(i) - 0.0001$
- 7: $B = A - alpha(i)I$


```

8:   AUTOVALORES(i)=0
9:    $X = [1 \ 1 \ \dots \ 1]^T$ 
10:  cnt=0, err=1, state=1
11:  While cnt ≤ maxl y state=1 do
12:     $Y = B^{-1}X$ 
13:    m : máximo componente de Y
14:    c1=m, dc=|AUTOVALORES(i)-c1|
15:     $Y = \frac{1}{c1}Y, dv = \|X - Y\|$ 
16:    err=max {dc,dv}
17:    X=Y, AUTOVALORES(i)=c1, state=0
18:    If err > ε then
19:      state=1
20:    EndIf
21:    cnt=cnt+1
22:  EndWhile
23: EndFor
24: For  $j = 1, \dots, n$  do
25:   R (j, j)=AUTOVALORES(j)
26:    $Q(:, j) = \text{AVECTOR}(:, j) / \|\text{AVECTOR}(:, j)\|$ 
27: EndFor
28: EndFunction

```

4.2.2.5. Algoritmo para la verificación de simetría

Este algoritmo permite verificar si la matriz de entrada es simétrica o no lo que hace es asignar a una variable s el número 0 si A es simétrica y 1 si A es no simétrica.

```

1:  Function [s]=VERIF_SIMETRICA(A)
2:  [n,n]=size(A);
3:  j=1;
4:  While j<n
5:    For i=j+1,...,n
6:      If A(i,j)=A(j,i)
7:        s=0;
8:      Else
9:        s=1;
10:     return;

```

```
11: End
12: End
13: j=j+1;
14: End
15: EndFunction
```

4.2.3. Diseño del algoritmo propuesto

El algoritmo propuesto basado en el método QR está compuesto por los algoritmos internos correspondientes al lenguaje de programación de Scilab y por los siguientes algoritmos propuestos por el autor como son el algoritmo de verificación de simetría para determinar si la matriz de entrada es una matriz simétrica, en caso afirmativo se usa el algoritmo de Householder que sirve para convertir matrices simétricas en matrices tridiagonales, que luego será usadas por el algoritmo QR Simétrico con el que se obtendrán los autovalores para el caso de matrices simétricas. Para el caso no simétrico se utilizará el algoritmo QR Schur que permitirá la descomposición de Schur de donde se obtendrá los autovalores. Para obtener los autovectores se utilizará el algoritmo de potencia inversa con traslación. La metodología divide y vencerás se muestra claramente en la Figura 4.

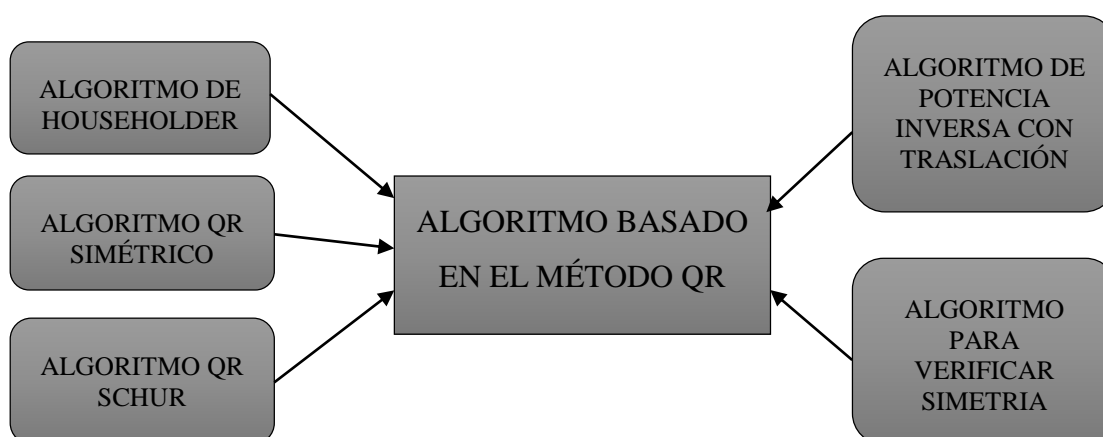


Figura 4. Algoritmo basado en el método QR compuesto por otros algoritmos

4.2.4. Algoritmo basado en el método QR

El siguiente algoritmo es el algoritmo central de la investigación, tiene como entradas a la matriz A y la matriz perturbadora E , donde E tiene una 2-norma muy pequeña. La suma $A+E$ determina una matriz llamada matriz perturbada, el algoritmo determina una Cota 1 para las distancias entre los autovalores y otra Cota 2 para las distancias de los subespacios invariantes determinados por los autovectores de la matriz original A y la matriz perturbada $A+E$. A continuación, se muestra el pseudocódigo respectivo:

```

1:  function [COT2,COTA 1, KAPPA]=ALGORITMO_BASADO_QR(A, E)
2:  n: orden de la matriz A
3:  s=VERIF_SIMETRICA(A)
4:  if s=0 then
5:    H=HOUSE( $E^t \times E$ )
6:    LAMBDA 1=METQR_SIMETRICO(H);
7:    COTA 1 =  $\sqrt{\max(\text{LAMBDA 1})}$ 
8:    B=HOUSE(A)
9:    LAMBDA_2=METQR_SIMETRICO(B)
10:   LAMBDA:=ordenar el vector LAMBDA 2
11:   [Q,R]=INVPOWTRAS(A,LAMBDA);
12:   KAPPA=1
13:   j=0
14:   while j<n
15:     for i=1,..., n-1
16:       D(i) =  $|R(1,1) - R(i+1, i+1)|$ 
17:     end
18:     d(j+1)=minimo(D);
19:     T =  $Q^t \times E \times Q$ 
20:     e=T(2:n,1);
21:     norma(j+1) =  $\sqrt{e^t \times e}$ 
22:     COTA 2 (j+1) =  $\frac{4 \times \text{norma}(j+1)}{d(j+1)}$ 
23:     T(:,1)=Q(:,1);
24:     w=R(1,1);
25:     for p=1,..., n-1
26:       Q(:,p)=Q(:,p+1);
27:       R(p,p)=R(p+1,p+1);

```

```

28:     end
29:     Q(:,n)=T(:,1);
30:     R(n,n)=w;
31:     j=j+1;
32: end
33: else
34: [Q,R]=METQR_SCHUR(A);
35: LAMBDA=diagonal de R
36: [QQ,RR]=INVPOWTRAS(A,LAMBDA);
37: [q1,r1]=METQR_SCHUR(E' x E);
38: NE=sqrt(max(diag(r1)))
39: [q2,r2]=METQR_SCHUR(QQ'*QQ);
40: X=sqrt(max(diag(r2)))
41: [q3,r3]=METQR_SCHUR(inv(QQ)'*inv(QQ));
42: XX=sqrt(max(diag(r3)))
43: COTA 1=X*XX*NE
44: KAPPA=X*XX
45: j=0;
46: while j<n
47:     C=Q' x A x Q
48:     M = C(2:n,2:n) - R(1,1) x I_{n-1}
49:     [q,r]=METQR_SCHUR(M' x M);
50:     D=sqrt(diag(r))
51:     d=min(real(D));
52:     T = Q' x E x Q
53:     e=T(2:n,1);
54:     norma=sqrt(e' x e)
55:     COTA 2(j+1)=frac(4 x norma, d)
56:     T(:,1)=Q(:,1);
57:     w=R(1,1);
58:     for p=1,...,n-1
59:         Q(:,p)=Q(:,p+1);
60:         R(p,p)=R(p+1,p+1);
61:     end
62:     Q(:,n)=T(:,1);
63:     R(n,n)=w;
64:     j=j+1;

```

```
65: end
66: end
67: endfunction
```

4.2.5. Resultado

Se diseñó el algoritmo considerando la técnica “divide y vencerás”, ya que el algoritmo propuesto se compone de otros algoritmos, entre ellos el algoritmo QR (método QR). El diseño está dividido en dos partes la primera corresponde para matrices simétricas y la segunda para matrices no simétricas. Para el caso de una matriz simétrica, el proceso comienza con la entrada de la matriz simétrica a analizar y su matriz perturbadora simétrica, luego mediante el Algoritmo Householder se transforma esta matriz simétrica en una matriz tridiagonal simétrica, de esta matriz se aplica el Algoritmo QR (método QR) para hallar los autovalores, después se utiliza el algoritmo de la potencia inversa con traslación para determinar los autovectores respectivos, finalmente, considerando las funciones internas de Scilab y funciones derivadas de los Teorema 1 y Teorema 2 se logra obtener la Cota 1 y la Cotas 2. Para el caso no simétrico el proceso comienza con las entradas dadas por la matriz a analizar y la matriz perturbadora no simétrica, obtenemos los autovalores directamente mediante el Algoritmo QR-Schur (método QR), en la diagonal de la matriz R , luego se utiliza el algoritmo de la potencia inversa con traslación para hallar los autovectores correspondientes, finalmente, considerando las funciones internas de Scilab y funciones derivadas de los Teorema 3 y Teorema 4 se logra obtener la Cota 1 y las Cotas 2. Las dos partes están dentro del algoritmo propuesto divididas por un algoritmo verificador que establece si la matriz de entrada es simétrica o no simétrica. De todos los algoritmos que forman parte del algoritmo propuesto la más importante son los que son el algoritmo QR simétrico y el algoritmo QR Schur ya que mediante de ellos se utilizó el método QR. Cabe señalar que se hizo una descripción de las ventajas del método QR en comparación con otros métodos dada la importancia de este método en el algoritmo propuesto.

4.2.6. Discusión

La prueba efectuada para comparar los métodos de Jacobi, Leverrier-Laguerre y QR, solo se hizo para una matriz simétrica, sin embargo, este hecho muestra que el método QR es mejor en cuanto a exactitud en comparación con estos métodos. Este resultado es coherente con las investigaciones de Yao (2015) y Panju (2011) que comparan diferentes métodos enfatizando las bondades del método QR a comparación con los demás métodos. Así mismo, la investigación de Laskar y Behera (2017) establecen que los métodos de Jacobi y Givens tienen la limitación de que funcionan eficientemente solo para matrices simétricas, por lo que método QR se muestra como una buena opción para el estudio de las propiedades de autovalores y autovectores. En el mismo sentido, Badia y Vidal (2000) muestran las ventajas y desventajas de considerar un algoritmo basado en el método de Laguerre. También, Aurentz (2017) propone mejoras considerando al método QR como método Base. El algoritmo propuesto no considera la versión más sofisticada del método QR, sin embargo, cumplen los resultados previstos. Este resultado puede servir a investigadores para considerar el método QR una buena herramienta para estudios relacionados con autovalores y autovectores. El algoritmo propuesto se diseñó utilizando el método divide y vencerás, por la ventaja que tiene como los indican Valenzuela (2003) y Cervantez (2014) para algoritmos que se pueden componer de otros más básicos.

4.3. Codificación, ejecución y verificación del algoritmo QR extendido para analizar la sensibilidad de autovalores y autovectores

A continuación, se efectuarán pruebas al algoritmo propuesto con matrices simétricas y no simétricas, para ello se procederá a codificar y ejecutar el pseudocódigo dado en la sección anterior en el programa Scilab.

4.3.1. Primera prueba para una matriz simétrica

Sea la matriz de prueba simétrica y la matriz perturbadora

$$A = \begin{bmatrix} 10.015788 & -1.0095367 & -6.9925099 & 6.1966743 \\ -1.0095367 & 22.933631 & -0.4440989 & 0.600907 \\ -6.9925099 & -0.4440989 & 18.179992 & 5.9007308 \\ 6.1966743 & 0.600907 & 5.9007308 & 19.87059 \end{bmatrix} \text{ y } E = \begin{bmatrix} 0.002 & 0.001 & 0.002 & 0.003 \\ 0.001 & 0.003 & 0 & 0.003 \\ 0.002 & 0 & 0 & 0.003 \\ 0.003 & 0.003 & 0.003 & 0.002 \end{bmatrix}$$

Luego, de aplicar el algoritmo propuesto y el método analítico a las matrices A y E se obtuvo el valor exacto y aproximado de la Cota 1. Ambos, ambos valores son mostrados en la Tabla 11 para su comparación y posterior análisis de sensibilidad de autovalores.

Tabla 11

Resultados para el análisis de sensibilidad para autovalores de la primera prueba

| N° | AUTOVALORES | | DISTANCIA ENTRE LOS AUTOVALORES | COTA 1 MÉTODO ANALÍTICO | COTA 1 ALGORITMO PROPUESTO |
|----|-----------------|-------------------|---------------------------------|-------------------------|----------------------------|
| | MATRIZ ORIGINAL | MATRIZ PERTURBADA | | | |
| 1 | 25 | 25.004543 | 0.004543 | 0.00819652671687365 | 0.00819652671687348 |
| 2 | 23 | 23.002535 | 0.002535 | 0.00819652671687365 | 0.00819652671687348 |
| 3 | 21 | 21.000162 | 0.000162 | 0.00819652671687365 | 0.00819652671687348 |
| 4 | 2 | 1.99976 | 0.00024 | 0.00819652671687365 | 0.00819652671687348 |

La Tabla 11 nos indica que las distancias entre los autovalores $|\bar{\lambda}_i - \lambda_i|$ son menores que la Cota 1 dado por 0.00819652671687348. Además, dado que la Cota 1 solo depende de la norma de E y del número de condición de A que es 1, se deduce que pequeños cambios en la matriz original generan pequeños cambios en los autovalores.

Ahora, de la Cota 2 obtenida por el algoritmo propuesto y por el método analítico se procederá a hacer la comparación dada en la Tabla 12, además en esa tabla se analizará la sensibilidad de los autovectores de A .

Tabla 12

Resultados para analizar la sensibilidad de autovectores de la primera prueba

| N° | AUTOVECTORES | | DISTANCIA ENTRE LOS SUBESPACIOS INVARIANTES | COTA 2 MÉTODO ANALÍTICO | COTA 2 ALGORITMO PROPUESTO |
|----|---|---|--|-------------------------------|----------------------------------|
| | MATRIZ ORIGINAL | MATRIZ PERTURBADA | | | |
| 1 | 0.0087113 0.0785338 0.6441855 0.760777 | 0.0091923 0.0795148 0.6437574 0.7610317 | 0.0012008 | 0.00856137624510381 | 0.00856137624507716 |
| 2 | 0.0785338 -0.9938857 0.0501534 0.0592307 | 0.0779554 -0.9938563 0.0513332 0.0594766 | 0.001337 | 0.0055000184654986 | 0.00550001846547525 |
| 3 | 0.6441855 0.0501534 -0.5886088 0.4858491 | 0.644339 0.0491615 -0.5889787 0.4852983 | 0.0012032 | 0.00821386562081478 | 0.00821386562080804 |
| 4 | 0.760777 0.0592307 0.4858491 -0.4262167 | 0.7607008 0.0592465 0.485845 -0.4263552 | 0.0001589 | 0.00070251173886845 | 0.00070251173886657 |

En la Tabla 12 se muestra la Cotas 2 de las distancias de los subespacios invariantes de la matriz original y de su matriz perturbada correspondiente, la cota para el primer autovector es 0.00856137624507716, para el segundo autovector 0.00550001846547525, para el tercer autovector 0.00821386562080804 y finalmente para el cuarto autovector 0.00070251173886657. Además, se observa que la cota correspondiente al cuarto autovector es más pequeña, esto significa que es menos sensible. Este hecho es coherente con el marco teórico, ya que su autovalor correspondiente es el más aislado.

4.3.2. Segunda prueba para una matriz simétrica

Sea la matriz de prueba simétrica y la matriz perturbadora

$$A = \begin{bmatrix} 29.905214 & -1.0592948 & -0.7596312 & -0.1299213 \\ -1.0592948 & 10.544273 & -2.5284855 & 5.2054425 \\ -0.7596312 & -2.5284855 & 12.725158 & 4.3341416 \\ -0.1299213 & 5.2054425 & 4.3341416 & 6.8253551 \end{bmatrix} \text{ y } E = \begin{bmatrix} 0.002 & 0.002 & 0.001 & 0.002 \\ 0.002 & 0.001 & 0.002 & 0.001 \\ 0.001 & 0.002 & 0.001 & 0.002 \\ 0.002 & 0.001 & 0.002 & 0 \end{bmatrix}$$

Luego, de aplicar el Algoritmo propuesto y el método analítico a las matrices A y E se obtuvo el valor exacto y aproximado de la Cota 1. Ambos, ambos valores son mostrados en la Tabla 13 para su comparación y posterior análisis de sensibilidad de autovalores.

Tabla 13

Resultados para el análisis de sensibilidad para autovalores de la segunda prueba

| N° | AUTOVALORES | | DISTANCIA ENTRE LOS AUTOVALORES | COTA 1 MÉTODO ANALÍTICO | COTA 1 ALGORITMO PROPUESTO |
|----|-----------------|-------------------|---------------------------------|-------------------------|----------------------------|
| | MATRIZ ORIGINAL | MATRIZ PERTURBADA | | | |
| 1 | 30 | 30.001601 | 0.0016006 | 0.00607300031017846 | 0.00607300031017847 |
| 2 | 15 | 15.00324 | 0.0032397 | 0.00607300031017846 | 0.00607300031017847 |
| 3 | 14 | 13.999898 | 0.0001016 | 0.00607300031017846 | 0.00607300031017847 |
| 4 | 1 | 0.9992614 | 0.0007386 | 0.00607300031017846 | 0.00607300031017847 |

La Tabla 13 nos indica que las distancias entre los autovalores $|\bar{\lambda}_i - \lambda_i|$ son menores que la Cota 1 dado por 0.00607300031017847. Además, dado que la cota 1 solo depende de la norma de E y el número de condición de A que es 1, se deduce que pequeños cambios en la matriz original generan pequeños cambios en los autovalores.

Ahora, de la Cota 2 obtenida por el algoritmo propuesto y por el método analítico se procederá a hacer la comparación dada en la Tabla 14, además en esa tabla se analizará la sensibilidad de los autovectores de A .

Tabla 14

Resultados para analizar la sensibilidad de autovectores de la segunda prueba

| N° | AUTOVECTORES | | DISTANCIA ENTRE LOS SUBESPACIOS INVARIANTES | COTA 2 MÉTODO ANALÍTICO | COTA 2 ALGORITMO PROPUESTO |
|----|---|---|--|-------------------------------|----------------------------------|
| | MATRIZ ORIGINAL | MATRIZ PERTURBADA | | | |
| 1 | 0.9972117 -0.055768 -0.0422142 -0.026012 | 0.9972243 -0.0556458 -0.0421526 -0.0258913 | 0.0001829 | 0.00075641082389864 | 0.00075641082389851 |
| 2 | 0.055768 0.1154182 0.8443287 0.5202669 | 0.055729 0.1172375 0.8435899 0.5210622 | 0.0021188 | 0.01187333817943153 | 0.01187333817943301 |
| 3 | 0.0422142 0.8443287 -0.3608757 0.393822 | 0.04197 0.8441324 -0.3626224 0.3926627 | 0.0021197 | 0.01257884780120204 | 0.01257884780120185 |
| 4 | 0.026012 0.5202669 0.393822 -0.7573308 | 0.0260089 0.5201917 0.3938074 -0.7573901 | 0.000097 | 0.00039773157281274 | 0.00039773157281269 |

En la Tabla 14 se muestra la Cotas 2 de las distancias de los subespacios invariantes de la matriz original y de su matriz perturbada correspondiente, la cota para el primer autovector es 0.00075641082389851, para el segundo autovector 0.01187333817943301, para el tercer autovector 0.01257884780120185 y finalmente para el cuarto autovector 0.00039773157281269. Además, se observa que la cota correspondiente al primero y cuarto autovector es más pequeña, esto significa que son menos sensibles. Este hecho es coherente con el marco teórico, ya que sus autovalores correspondiente son más aislados.

4.3.3. Tercera prueba para una matriz no simétrica

Sea la matriz de prueba no simétrica y la matriz perturbadora

$$A = \begin{bmatrix} 5.2857143 & -7.7142857 & -5.7142857 & 18.285714 \\ -9.4285714 & 6.5714286 & -5.4285714 & 18.571429 \\ -9.1428571 & -7.1428571 & 7.8571429 & 18.857143 \\ -5.7142857 & -3.7142857 & -1.7142857 & 23.285714 \end{bmatrix} \text{ y } E = \begin{bmatrix} 0.003 & 0.002 & 0.001 & 0.003 \\ 0.003 & 0.001 & 0 & 0.003 \\ 0.002 & 0.003 & 0.001 & 0.002 \\ 0.002 & 0.002 & 0.001 & 0 \end{bmatrix}$$

Luego, de aplicar el Algoritmo propuesto y el método analítico a las matrices A y E se obtuvo el valor exacto y aproximado de la Cota 1. Ambos, ambos valores son mostrados en la Tabla 15 para su comparación y posterior análisis de sensibilidad de autovalores.

Tabla 15

Resultados para el análisis de sensibilidad para autovalores de la tercera prueba

| N° | AUTOVALORES | | DISTANCIA ENTRE LOS AUTOVALORES | COTA 1 MÉTODO ANALÍTICO | COTA 1 ALGORITMO PROPUESTO |
|----|-----------------|-------------------|---------------------------------|-------------------------|----------------------------|
| | MATRIZ ORIGINAL | MATRIZ PERTURBADA | | | |
| 1 | 15 | 14.99871 | 0.0012905 | 0.05512275625824156 | 0.05512260213847964 |
| 2 | 14 | 14.001294 | 0.0012942 | 0.05512275625824156 | 0.05512260213847964 |
| 3 | 13 | 13.000712 | 0.0007121 | 0.05512275625824156 | 0.05512260213847964 |
| 4 | 1 | 1.0042843 | 0.0042843 | 0.05512275625824156 | 0.05512260213847964 |

La Tabla 15 nos indica que las distancias entre los autovalores $|\bar{\lambda}_i - \lambda_i|$ son menores que la cota 0.05512260213847964. Además, dado que la cota 1 depende de la norma de E y el número de condición de A dado por $\Gamma_2(A) = 7$, esto implica que pequeños cambios en la matriz original generan pequeños cambios en los autovalores.

Ahora, de la Cota 2 obtenida por el algoritmo propuesto y por el método analítico se procederá a hacer la comparación dada en la Tabla 16, además en esa tabla se analizará la sensibilidad de los autovectores de A .

Tabla 16

Resultados para analizar la sensibilidad de autovectores de la tercera prueba

| N° | AUTOVECTORES | | DISTANCIA ENTRE LOS SUBESPACIOS INVARIANTES | COTA 2 MÉTODO ANALÍTICO | COTA 2 ALGORITMO PROPUESTO |
|----|--|--|---|-------------------------|----------------------------|
| | MATRIZ ORIGINAL | MATRIZ PERTURBADA | | | |
| 1 | 0.2773501 0.5547002 0.5547002 0.5547002 | 0.2782407 0.5540741 0.5548662 0.5547139 | 0.0011013 | 0.01261950785560819 | 0.01247327270414476 |
| 2 | 0.5547002 0.2773501 0.5547002 0.5547002 | 0.5543726 0.2783891 0.5544932 0.5547142 | 0.001109 | 0.04839927130540404 | 0.04741664000126998 |
| 3 | 0.5547002 0.5547002 0.2773501 0.5547002 | 0.5544999 0.5551545 0.277196 0.554523 | 0.0005493 | 0.03737888539813459 | 0.03680051919691871 |
| 4 | 0.5547002 0.5547002 0.5547002 0.2773501 | 0.5546764 0.5547607 0.5546972 0.2772827 | 0.0000937 | 0.00041267643383774 | 0.00041267643383762 |

En la Tabla 16 se muestra la Cotas 2 de las distancias de los subespacios invariantes de la matriz original y de su matriz perturbada correspondiente, la cota para el primer autovector es 0.01247327270414476, para el segundo autovector 0.04741664000126998, para el tercer autovector 0.03680051919691871 y finalmente para el cuarto autovector 0.00041267643383762. Además, se observa que la cota correspondiente al cuarto autovector es más pequeña, esto significa que es menos sensible. Este hecho es coherente con el marco teórico, ya que su autovalor correspondiente es el más aislados.

4.3.4. Cuarta prueba para una matriz no simétrica

Sea la matriz de prueba no simétrica y la matriz perturbadora

$$A = \begin{bmatrix} 29.857143 & 4.8571429 & 2.8571429 & -25.142857 \\ 30.571429 & 10.571429 & -1.4285714 & -29.428571 \\ 30.285714 & 0.2857143 & 9.2857143 & -29.714286 \\ 26.285714 & -3.7142857 & -5.7142857 & -8.7142857 \end{bmatrix} \text{ y } E = \begin{bmatrix} 0 & 0.001 & 0.001 & 0.001 \\ 0.002 & 0 & 0.002 & 0.002 \\ 0.001 & 0.001 & 0.001 & 0.002 \\ 0.001 & 0.003 & 0.001 & 0.001 \end{bmatrix}$$

Luego, de aplicar el Algoritmo propuesto y el método analítico a las matrices A y E se obtuvo el valor exacto y aproximado de la Cota 1. Ambos, ambos valores son mostrados en la Tabla 17 para su comparación y posterior análisis de sensibilidad de autovalores.

Tabla 17

Resultados para el análisis de sensibilidad para autovalores de la cuarta prueba

| N° | AUTOVALORES | | DISTANCIA ENTRE LOS AUTOVALORES | COTA 1 MÉTODO ANALÍTICO | COTA 1 ALGORITMO PROPUESTO |
|----|-----------------|-------------------|---------------------------------|-------------------------|----------------------------|
| | MATRIZ ORIGINAL | MATRIZ PERTURBADA | | | |
| 1 | 25 | 24.998714 | 0.001286 | 0.03625184056916531 | 0.0362517990742639 |
| 2 | 11 | 11.001001 | 0.0010005 | 0.03625184056916531 | 0.0362517990742639 |
| 3 | 10 | 9.9980001 | 0.0019999 | 0.03625184056916531 | 0.0362517990742639 |
| 4 | -5 | -4.9957147 | 0.0042853 | 0.03625184056916531 | 0.0362517990742639 |

La Tabla 17 nos indica que las distancias entre los autovalores $|\bar{\lambda}_i - \lambda_i|$ son menor que la Cota 1 dado por 0.0362517990742639. Además, dado que la Cota 1 depende de la norma de E y el número de condición de A dado por $\Gamma_2(A) = 6.9999999$, esto es implica que pequeños cambios en la matriz original generan pequeños cambios en los autovalores.

Ahora, de la Cota 2 obtenida por el algoritmo propuesto y por el método analítico se procederá a hacer la comparación dada en la Tabla 18, además en esa tabla se analizará la sensibilidad de los autovectores de A .

Tabla 18

Resultados para analizar la sensibilidad de autovectores de la cuarta prueba

| N° | AUTOVECTORES | | DISTANCIA ENTRE LOS SUBESPACIOS INVARIANTES | COTA 2 MÉTODO ANALÍTICO | COTA 2 ALGORITMO PROPUESTO |
|----|--------------------|----------------------|--|-------------------------------|----------------------------------|
| | MATRIZ ORIGINAL | MATRIZ PERTURBADA | | | |
| 1 | 0.5547002 | 0.5546677 | 0.0000889 | 0.00064803167630245 | 0.00064803167628939 |
| | 0.5547002 | 0.5547166 | | | |
| | 0.5547002 | 0.5546773 | | | |
| | 0.2773501 | 0.2774279 | | | |
| 2 | 0.5547002 | 0.5546299 | 0.0001355 | 0.01123723619504494 | 0.01275489006532648 |
| | 0.5547002 | 0.5547166 | | | |
| | 0.2773501 | 0.2774648 | | | |
| | 0.5547002 | 0.5546968 | | | |
| 3 | 0.5547002 | 0.5545765 | 0.0003514 | 0.02066454578697921 | 0.01722406015695867 |
| | 0.2773501 | 0.2771316 | | | |
| | 0.5547002 | 0.5549457 | | | |
| | 0.5547002 | 0.5546875 | | | |
| 4 | 0.2773501 | 0.2773275 | 0.0000312 | 0.00059829751053004 | 0.00059829751053004 |
| | 0.5547002 | 0.5547023 | | | |
| | 0.5547002 | 0.5547193 | | | |
| | 0.5547002 | 0.5546904 | | | |

En la Tabla 18 se muestra la Cotas 2 de las distancias de los subespacios invariantes entre la matriz original y la matriz perturbada correspondiente, la cota para el primer autovector es 0.00064803167628939, para el segundo autovector 0.01275489006532648, para el tercer autovector 0.01722406015695867 y finalmente para el cuarto autovector 0.00059829751053004. Además, se observa que la cota correspondiente al primero y cuarto autovector es más pequeña, esto significa que son menos sensibles. Este hecho es coherente con el marco teórico, ya que su autovalor correspondiente es el más aislados.

4.3.5. Resultado

El algoritmo propuesto se codificó y ejecutó en el programa Scilab 6.0.2 considerando el seudocódigo dado en la sección de diseño, mostrado en el anexo. La verificación se hizo en dos tablas, la primera tabla corresponde a la sensibilidad de los autovalores, de ahí se observó que la Cota 1 obtenida por el método analítico y el obtenido con el método propuesto tienen valores muy aproximados hasta más de 10 cifras decimales exactas. Similarmente, de la segunda tabla que corresponde a la sensibilidad de los autovectores, se observa que las entradas de la matriz columna Cota 2 calculadas por los métodos analíticos y la Cota 2 calculada por el algoritmo propuesto, tienen valores muy aproximados hasta más de 10 cifras decimales exactas.

Finalmente, de esos resultados se analizó la sensibilidad de autovalores y autovectores de una matriz real bien condicionada de manera satisfactoria.

4.3.6. Discusión

Las pruebas se aplicaron a matrices de orden 4×4 , esto debido a que la ecuación característica solo tiene solución exacta (mediante radicales) para ecuación de grado menores o iguales a 4, esto no representa un problema, ya que los resultados observados de estas pruebas también ocurren para matrices de órdenes superiores, dado que estas son consecuencias del Teorema 1 y Teorema 2 para el caso de matrices simétricas y los Teorema 3 y Teorema 4 para el caso de matrices no simétricas de orden $n \times n$. En estas pruebas se consideró matrices bien condicionadas, es decir, matrices que tienen número de condición no muy grande. En cuanto a las tablas de resultados, las cotas obtenidas por el algoritmo basado en el método QR son muy aproximados en comparación con los obtenidos con el método analítico (exacto), hasta más de 10 cifras decimales exactas. Estos resultados reafirman la investigación hecha por Cho e Ipsen (2008) en el estudio de las propiedades de sensibilidad de matrices y en particular de aquellas con autovalor único repetido. Así mismo, los resultados obtenidos concuerdan con las obtenidas por Bona (1999) en cuanto a la sensibilidad de autovalores y autovectores con matrices reales. El estudio de sensibilidades de autovalores también depende del tamaño de la matriz y la forma como están dados sus entradas, en ese sentido Ballard et al. (2018) estudia el uso de programación paralela para calcular las descomposiciones QR en matrices de ordenes grandes y Yeralan et al. (2017) en el aprovechamiento de procesadores GPUs para abordar matrices grandes y densas. Además, estas pruebas fueron realizadas en Scilab 6.0.2, los códigos utilizados se encuentran en el anexo.

CONCLUSIONES

Se analizó el problema de sensibilidad de autovalores y autovectores reales de matrices reales bien condicionadas. En este análisis se identificó como entrada del problema a las matrices cuadradas de orden arbitrario con entradas reales que tenga números condición pequeña (cercano a uno) y otra matriz de igual orden con entradas reales que tiene 2-norma pequeña denominada matriz perturbadora. En cuanto a la salida se identificó a un número real denominado Cota 1 y una matriz columna con entras reales denominado Cotas 2. Para obtener estas salidas a partir de las matrices de entrada, se tiene que seguir un proceso determinado por los resultados matemáticos dados por el Teorema 1 y Teorema 2 para el caso de matrices simétricas y el Teorema 3 y Teorema 4 para el caso de matrices no simétricas.

Se diseñó un algoritmo basado en el método QR siguiendo el método denominado divide y vencerás, esto es, el algoritmo propuesto se compone de algoritmos básicos, entre ellos tenemos: el algoritmo Householder, el algoritmo de potencia inversa con traslación, el algoritmo QR para matrices simétricas y el algoritmo QR con descomposición de Schur y funciones preestablecidas de Scilab. Para luego obtener el algoritmo propuesto con una eficiencia del orden $O(n^4)$.

Se codificó y ejecutó el algoritmo propuesto en el lenguaje de programación de Scilab, luego se verificó el algoritmo haciendo pruebas a matrices especificadas cuyos resultados obtenidos se mostraron en tablas, donde se compararon las cotas obtenidas mediante el algoritmo propuesto y los obtenidos por el método analítico, donde se

observaron valores muy aproximados hasta más de 10 cifras decimales exactas. Finalmente, de esos resultados se analizó la sensibilidad de autovalores y autovectores de una matriz real bien condicionada de manera satisfactoria.

RECOMENDACIONES

Se recomienda para futuras investigaciones considerar matrices con entradas complejas, con autovalores repetidos y mal condicionadas.

Se recomienda utilizar métodos mejorados del método QR en cuanto a su eficiencia para la obtención de los autovalores. Así mismo, se recomienda otros diseños de programación y la utilización de la programación paralela.

En cuanto a la codificación se recomienda utilizar otros entornos de programación, así mismo se sugiere la implementación de interfaces de usuario. También, se recomienda utilizar la presente investigación para casos reales como son en el estudio de imágenes digitales y sistemas oscilatorios.

BIBLIOGRAFÍA

- Aurentz, J. L., Mach, T., Vandebril, R., & Watkins, D. S. (2015). Fast and stable unitary QR algorithm. *Electron. Trans. Numer. Anal.*, *44*, 327–341.
- Aurentz, J. L., Mach, T., Vandebril, R., & Watkins, D. S. (2017). Computing the eigenvalues of symmetric tridiagonal matrices via a Cayley transformation. *Electronic Transactions on Numerical Analysis*, *46*, 447–459.
- Badia Contelles, J. M., & Vidal Maciá, A. M. (2000). Cálculo de los valores propios de matrices tridiagonales simétricas mediante la iteración de Laguerre. *Revista Internacional de Métodos Numéricos*.
- Ballard, G., Demmel, J., Grigori, L., Jacquelin, M., & Knight, N. (2018). A 3D Parallel Algorithm for QR Decomposition. *ArXiv Preprint ArXiv:1805.05278*.
- Barchini, G. E. (2005). Métodos I+ D de la Informática. *Revista de Informática Educativa y Medios Audiovisuales*, *2*(5), 16–24.
- Bona, F. (1999.). *Estudio de la sensibilidad de autovalores y autovectores*. Departamento de computacion. Universidad de Buenos Aires.
- Burden, R. L. (1998). *Análisis Numérico* (1st ed.). Thomson Editores.
- Cervantes-Sánchez, J. J., & de Jesús Pérez-Lona, J. (2018). Método de análisis de vibraciones en sistemas mecánicos multicuerpo. *Cervantes*, *6*, 88.

- Cervantez, J. (2014). *Introducción al análisis y al diseño de algoritmos* (Vol. 1). Publidisa Mexicana.
- Cho, G. E., & Ipsen, I. C. F. (1998). If a Matrix Has a Single Eigenvalue, How Sensitive is This Eigenvalue. *CRSC Technical Report*, (CRSC-TR98-8), 1–11.
- de Lima, P. E., & de Souza, L. de F. R. (2013). Autovalores e autovetores: conceitos e uma aplicação a um sistema dinâmico. *Revista Eletrônica de Educação e Ciência*, 3(1), 22–28.
- Francis, J. G. F. (1961). The QR transformation a unitary analogue to the LR transformation—Part 1. *The Computer Journal*, 4(3), 265–271.
- Garrett, C. K., Bai, Z., & Li, R.-C. (2016). A nonlinear QR algorithm for banded nonlinear eigenvalue problems. *ACM Transactions on Mathematical Soft (TOMS)*, 43(1), 4.
- Golub, G. (1996). *Matrix computations* (3rd ed.). The Johns Hopkins University Press.
- González Marrero, C., & otros. (2018). *Álgebra Lineal Numérica*.
- Grossman, S. I. (1996). *Álgebra lineal* (4th ed.). Mc GrawHill.
- Joyanes, L. (1996). *Fundamentos de Programacion* (2nd ed.). Mc GrawHill.
- Laskar, A. H., & Behera, S. (2017). A Comparative Study of Jacobi Method and Givens Method for Finding Eigenvalues and Eigenvectors of a Real Symmetric Matrices. *Rn*, 55, 7.
- Lay, R. (2007). *Algebra Lineal y sus Aplicaciones* (3rd ed.). PEARSON.
- Mathews, J. H. (2000). *Métodos Numéricos con matlab* (3rd ed.). Prentice Hall.
- Oliveira, D. E. de. (2006). *Sobre um método assemelhado ao de Francis para a determinação de autovalores de matrizes*.
- Panju, M. (2011). Iterative methods for computing eigenvalues and eigenvectors. *ArXiv*

Preprint ArXiv:1105.1185.

Stewart, G. W. (1998). *Matrix Algorithms: Volume 1: Basic Decompositions* (Vol. 1). Siam.

Valenzuela, V. (2003). *Manual de Analisis y Diseño de Algoritmos* (1st ed.). INACAP.

Vazquez L. y Jimenz, S. (2009). *Metodos numericos para la fisica y la ingenieria* (1st ed.). Mc Graw Hill.

Yao, D. (2015). Some Methods of Evaluating Eigenvalues of symmetric Matrix. *4th International Conference on Mechatronics*, 30(2).

Yeralan, S. N., Davis, T. A., Sid-Lakhdar, W. M., & Ranka, S. (2017). Algorithm 980: Sparse QR factorization on the GPU. *ACM Transactions on Mathematical Software (TOMS)*, 44(2), 17.

ANEXO

Anexo 1: Código en Scilab del algoritmo propuesto

```

1: function [COT2, LAMBDA, COT1, KAPPA]=ALGORITMO BASADO QR(A, E)
2: [n,n]=size(A);
3: format('v',20);
4: v=format();
5: format(20);
6: s=VERIF_SIMETRICA(A)
7: if s==0 then
8:     //Se calcula la cota 1
9:     H=HOUSE(E'*E);
10:    LAMBDA_1=METQR_SIMETRICO(H);
11:    COT1=sqrt(max(LAMBDA_1));
12:    //Se calcula la cota 2
13:    B=HOUSE(A);
14:    LAMBDA_2=METQR_SIMETRICO(B);
15:    LAMBDA=gsort(LAMBDA_2);
16:    [Q,R]=INVPOWTRAS(A,LAMBDA);
17:    KAPPA=1;
18:    j=0;
19:    while j<n
20:        for i=1:n-1
21:            D(i)=abs(R(1,1)-R(i+1,i+1));
22:        end
23:        d(j+1)=min(D);
24:        T=Q'*E*Q;
25:        e=T(2:n,1);
26:        norma(j+1)=sqrt(e'*e);
27:        COT2(j+1)=(4*norma(j+1))/d(j+1);
28:        T(:,1)=Q(:,1);
29:        w=R(1,1);
30:        for p=1:n-1
31:            Q(:,p)=Q(:,p+1);
32:            R(p,p)=R(p+1,p+1);
33:        end
34:        Q(:,n)=T(:,1);
35:        R(n,n)=w;
36:        j=j+1;
37:    end
38: else
39:     //Se calcula la cotal
40:     [Q,R]=METQR_SCHUR(A);
41:     LAMBDA=diag(R);
42:     [QQ,RR]=INVPOWTRAS(A,LAMBDA);
43:     [q_1,r_1]=METQR_SCHUR(E'*E);
44:     NE=sqrt(max(diag(r_1)));
45:     [q_2,r_2]=METQR_SCHUR(QQ'*QQ);
46:     X=sqrt(max(diag(r_2)));

```

```

47:     [q_3,r_3]=METQR_SCHUR(inv(QQ)'*inv(QQ));
48:     XX=sqrt(max(diag(r_3)));
49:     COT1=X*XX*NE;
50:     KAPPA=X*XX;
51:     //Se calcula la cota2
52:     j=0;
53:     while j<n
54:         G=Q'*A*Q;
55:         M=G(2:n,2:n)-R(1,1)*eye(n-1,n-1);
56:         [q,r]=METQR_SCHUR(M'*M);
57:         D=sqrt(diag(r));
58:         d=min(real(D));
59:         T=Q'*E*Q;
60:         e=T(2:n,1);
61:         norma=sqrt(e'*e);
62:         COT2(j+1)=(4*norma)/d;
63:         //nuevo Q y nuevo R
64:         T(:,1)=Q(:,1);
65:         w=R(1,1);
66:         for p=1:n-1
67:             Q(:,p)=Q(:,p+1);
68:             R(p,p)=R(p+1,p+1);
69:         end
70:         Q(:,n)=T(:,1);
71:         R(n,n)=w;
72:         j=j+1;
73:     end
74: end
75: endfunction

```

Anexo 2: Código en Scilab del algoritmo Householder

```

1: function [B]=HOUSE(A)
2: [n n]=size(A);
3: if n==2 then
4:     B=A;
5:     return
6: end
7: for k=1:1:n-2
8:     qq=0;
9:     for j=k+1:1:n
10:         r=A(j,k)*A(j,k);
11:         qq=qq+r;
12:     end
13:     if A(k+1,k)==0
14:         alpha=-sqrt(qq);
15:     else
16:         alpha=-(sqrt(qq)*A(k+1,k))/abs(A(k+1,k));
17:     end
18:     RSQ=alpha^2-alpha*A(k+1,k);
19:     v(k)=0;
20:     v(k+1)=A(k+1,k)-alpha;

```

```

21:     for j=k+2:1:n
22:         v(j)=A(j,k);
23:     end
24:     for j=k:1:n
25:         p=0;
26:         for i=k+1:1:n
27:             p=(A(j,i)*v(i))+p;
28:         end
29:         u(j)=p/RSQ;
30:     end
31:     w=0;
32:     for i=k+1:1:n
33:         w=v(i)*u(i)+w;
34:     end
35:     PROD=w;
36:     for j=k:1:n
37:         z(j)=u(j)-(PROD/(2*RSQ))*v(j);
38:     end
39:     for l=k+1:1:n-1
40:         for j=l+1:1:n
41:             B(j,l)=A(j,l)-v(l)*z(j)-v(j)*z(l);
42:             B(l,j)=B(j,l);
43:         end
44:         B(l,l)=A(l,l)-2*v(l)*z(l);
45:     end
46:     B(n,n)=A(n,n)-2*v(n)*z(n);
47:     for j=k+2:1:n
48:         B(k,j)=0;
49:         B(j,k)=0;
50:     end
51:     B(k+1,k)=A(k+1,k)-v(k+1)*z(k);
52:     B(k,k+1)=B(k+1,k);
53:     B(1,1)=A(1,1);
54:     //Pasar de B a A
55:     for i=1:1:n
56:         for j=1:1:n
57:             A(i,j)=B(i,j);
58:         end
59:     end
60:     u=zeros(1,n);
61:     z=zeros(1,n);
62: end
63: endfunction

```

Anexo 3: Código en Scilab del algoritmo potencia inversa con traslación

```

1: function [Q,R]=INVPOWTRAS(A,LAMBDA)
2: [n n]=size(A);
3: epsilon=0.00001;
4: maxl=100;
5: for i=1:n

```



```

6:     alpha(i)=LAMBDA(i)-0.00001;
7:     B=A-alpha(i)*eye(n,n);
8:     AUTOVALORES(i)=0;
9:     X=ones(n,1);
10:    cnt=0;
11:    err=1;
12:    state=1;
13:    while ((cnt<=max1)&(state==1))
14:        Y=B^-1*X;
15:        [m j]=max(abs(Y));
16:        c1=m;
17:        dc=abs(AUTOVALORES(i)-c1);
18:        Y=(1/c1)*Y;
19:        dv=norm(X-Y);
20:        err=max(dc,dv);
21:        X=Y;
22:        AUTOVALORES(i)=c1;
23:        state=0;
24:        if (err>epsilon)
25:            state=1;
26:        end
27:        cnt=cnt+1;
28:    end
29:    AUTOVALORES(i)=alpha(i)+1/c1;
30:    AVECTOR(:,i)=X;
31: end
32: R=zeros(n);
33: for j=1:1:n
34:     R(j,j)=AUTOVALORES(j);
35:     Q(:,j)=AVECTOR(:,j)/norm(AVECTOR(:,j));
36: end
37: endfunction

```

Anexo 4: Código en Scilab del algoritmo QR simétrico

```

1: function [LAMBDA]=METQR_SIMETRICO(A)
2: TOL=0.00000001;
3: N=2000;
4: [n n]=size(A);
5: for i=1:1:n
6:     a(i)=A(i,i);
7: end
8: for i=1:1:n-1
9:     b(i+1)=A(i+1,i);
10: end
11: k=1;
12: SHIFT=0;
13: while k<=N
14:     if abs(b(n))<=TOL
15:         LAMBDA(n)=a(n)+SHIFT;
16:         n=n-1;

```

```

17:     end
18:     p=- (a (n-1)+a (n) ) ;
19:     ce=a (n) *a (n-1) -b (n) ^2;
20:     de=sqrt (p^2-4*ce) ;
21:     mu1=(-p+de) /2;
22:     mu2=- (p+de) /2;
23:     if n==2
24:         LAMBDA (1) =mu1+SHIFT;
25:         LAMBDA (2) =mu2+SHIFT;
26:         LAMBDA=gsort (LAMBDA, "g", "i" );
27:         return;
28:     end
29:     h=abs (mu1-a (n) ) ;
30:     k=abs (mu2-a (n) ) ;
31:     if h==k
32:         es=mu2;
33:     else
34:         if h<k
35:             es=mu1;
36:         Else
37:             es=mu2;
38:         end
39:     end
40:     SHIFT=SHIFT+es;
41:     for j=1:1:n
42:         d (j) =a (j) -es;
43:     end
44:     x (1) =d (1) ;
45:     y (1) =b (2) ;
46:     for j=2:1:n
47:         z (j-1) =sqrt (x (j-1) ^2+(b (j) ) ^2) ;
48:         c (j) =x (j-1) /z (j-1) ;
49:         s (j) =b (j) /z (j-1) ;
50:         q (j-1) =c (j) *y (j-1) +s (j) *d (j) ;
51:         x (j) =-s (j) *y (j-1) +c (j) *d (j) ;
52:         if j~= n
53:             r (j-1) =s (j) *b (j+1) ;
54:             y (j) =c (j) *b (j+1) ;
55:         end
56:     end
57:     z (n) =x (n) ;
58:     aa (1) =s (2) *q (1) +c (2) *z (1) ;
59:     bb (2) =s (2) *z (2) ;
60:     for j=2:1:n-1
61:         aa (j) =s (j+1) *q (j) +c (j) *c (j+1) *z (j) ;
62:         bb (j+1) =s (j+1) *z (j+1) ;
63:     end
64:     aa (n) =c (n) *z (n) ;
65:     for i=1:1:n
66:         a (i) =aa (i) ;
67:     end
68:     for i=1:1:n-1

```

```

69:         b(i+1)=bb(i+1);
70:     end
71:     k=k+1;
72: end
73: endfunction

```

Anexo 5: Código en Scilab del algoritmo QR con descomposición de Schur

```

13: function [q,r]=METQR_SCHUR(A)
14: [n n]=size(A);
15: q=eye(n,n);
16: i=1;
17: while i<=50 do
18:     [Q,R]=FACTOR_QR(A)
19:     A=R*Q;
20:     q=q*Q;
21:     i=i+1;
22: end
23: r=A;
24: endfunction

```

Anexo 6: Código en Scilab del algoritmo para verificar si es simétrica

```

16: function [s]=VERIF_SIMETRICA(A)
17: [n,n]=size(A);
18: j=1;
19: while j<n
20:     for i=j+1:n
21:         if A(i,j)~=A(j,i)
22:             s=0;
23:         else
24:             s=1;
25:             return;
26:         end
27:     end
28:     j=j+1;
29: end
30: endfunction

```

Anexo 7: Código en Scilab del algoritmo de factorización QR

```

1: function [Q, R]=FACTOR_QR(A)
2: [n n]=size(A);

```

```

3: i=1;
4: Q=eye(n,n);
5: while i<n do
6:     alpha=-sign(A(i,i))*norm(A(i:n,i));
7:     e=[1; zeros(n-i,1)];
8:     v=A(i:n,i)-alpha*e;
9:     H=eye(n-i+1,n-i+1)-(2/norm(v)^2)*v*v';
10:    if i==1 then
11:        P=H;
12:    else
13:        P=zeros(n,n);
14:        P(1:(i-1),1:(i-1))=eye(i-1,i-1);
15:        P(i:n,i:n)=H;
16:    end
17:    A=P*A;
18:    Q=P*Q;
19:    i=i+1;
20: end
21: R=A;
22: Q=Q^-1;
23: endfunction

```

Anexo 8: Método de Jacobi

El método de Jacobi es un algoritmo para calcular todas las parejas autovalor-autovector de una matriz simétrica. Es un método fiable que proporciona respuestas uniformemente precisas y que para matrices de orden menor o igual que 10 es competitivo frente a otros métodos más sofisticados, también es aceptable para matrices de orden menor o igual que 20, si la velocidad de convergencia no es una cuestión muy relevante.

El método de Jacobi funciona para todas las matrices simétricas reales, esta limitación no es muy severa ya que, en la práctica, hay un gran número de problemas en la ingeniería en la matemática aplicada que involucran el cálculo de los autovalores de una matriz simétrica.

Rotaciones planas

Sea X un vector en el espacio n -dimensional y consideremos la aplicación lineal $Y = RX$ donde R es la matriz de orden $n \times n$ dada por:

$$R = \begin{bmatrix} 1 & \dots & 0 & \dots & 0 & \dots & 0 \\ \vdots & & \vdots & & \vdots & & \vdots \\ 0 & \dots & \cos \phi & \dots & \text{sen} \phi & \dots & 0 \\ \vdots & & \vdots & & \vdots & & \vdots \\ 0 & \dots & -\text{sen} \phi & \dots & \cos \phi & \dots & 0 \\ \vdots & & \vdots & & \vdots & & \vdots \\ 0 & \dots & 0 & \dots & 0 & \dots & 1 \end{bmatrix} \begin{matrix} \leftarrow \text{fila } p \\ \\ \leftarrow \text{fila } q \\ \\ \end{matrix}$$

$$\begin{matrix} \uparrow & \uparrow \\ \text{col } p & \text{col } q \end{matrix}$$

En esta expresión todos los términos de R que están fuera de la diagonal son ceros salvo dos que valen $\pm \text{sen} \phi$ y que todos los términos de la diagonal son 1 excepto los dos que valen $\cos \phi$. El efecto de esta transformación $Y = RX$ puede verse fácilmente:

$$\begin{aligned} y_j &= x_j && \text{cuando } j \neq p \text{ y } j \neq q \\ y_p &= x_p \cos \phi + x_q \text{sen} \phi \\ y_q &= -x_p \text{sen} \phi + x_q \cos \phi \end{aligned}$$

Tenemos entonces que esta aplicación lineal es una rotación del ángulo ϕ en el plano coordenado $X_p O X_q$. Eligiendo adecuadamente el ángulo ϕ , podemos conseguir que $y_p = 0$ o bien que $y_q = 0$ en el vector imagen. La transformación inversa $X = R^{-1}Y$ es una rotación del ángulo $-\phi$ en el mismo plano coordenado $X_p O X_q$. Se observa que R es una matriz ortogonal, o sea,

$$R^{-1} = R^t \text{ o bien } R^t R = I$$

Semejanza de matrices y transformaciones ortogonales

Consideremos el problema de autovalores

$$AX = \lambda X \quad (23)$$

Supongamos que K es una matriz invertible y definamos B mediante

$$B = K^{-1}AK \quad (24)$$

Multiplicando por $K^{-1}X$ a la derecha de la relación (24) obtenemos

$$\begin{aligned} BK^{-1}X &= K^{-1}AKK^{-1}X = K^{-1}AX \\ &= K^{-1}\lambda X = \lambda K^{-1}X \end{aligned} \quad (25)$$

Se hace ahora el cambio de variable

$$Y = K^{-1}X \text{ o bien } X = KY \quad (26)$$

Que al sustituirlo en (25), produce el problema de autovalores

$$BY = \lambda Y \quad (27)$$

Comparando los problemas (23) y (27), se ve que la transformación de semejanza (24) conserva el autovalor λ y que, aunque los autovectores son diferentes, están relacionados por el cambio de variable (26).

Supongamos que R es ortogonal y que C se define mediante

$$C = R'AR \quad (28)$$

Multiplicando por $R'X$ a la derecha de la relación (28) obtenemos

$$CR'X = R'ARR'X = R'AX = R'\lambda X = \lambda R'X \quad (29)$$

Se hace ahora el cambio de variable

$$Y = R^t X \text{ o } X = RY \quad (30)$$

que, al sustituirlo en (29), produce el problema de autovalores

$$CY = \lambda Y \quad (31)$$

Como antes, los autovalores de (23) y (31) son lo mismos. Sin embargo en el problema (31) el cambio de variables (30) es más fácil de deshacer ya que $R^{-1} = R^t$.

Si además suponemos que la matriz A es simétrica (es decir, $A = A^t$), entonces tenemos

$$C^t = (R^t A R)^t = R^t A (R^t)^t = R^t A R = C \quad (32)$$

Así que C es también una matriz simétrica. En consecuencia, si A es una matriz simétrica y R una matriz ortogonal, entonces la transformación de A en C dada por (28) conserva la simetría y los autovalores y la relación entre los autovectores viene dada por el cambio de variable (30).

Sucesión de transformaciones de Jacobi

Empezando con una matriz simétrica A , construimos una sucesión de matrices ortogonales: R_1, R_2, \dots , de la siguiente manera:

$$\begin{aligned} D_0 &= A, \\ D_j &= R_j^t D_{j-1} R_j, \quad j = 1, 2, \dots \end{aligned} \quad (33)$$

Se muestra como hay que construir la sucesión $\{R_j\}$ de manera que:

$$\lim_{j \rightarrow \infty} D_j = D = \text{diag}\{\lambda_1, \lambda_2, \dots, \lambda_n\} \quad (34)$$

En la practica el proceso se detiene cuando los elementos que están fuera de la diagonal son suficientemente pequeños, de manera que:

$$D_k \approx D \quad (35)$$

Siendo

$$D_k = R_k^t R_{k-1}^t \dots R_1^t A R_1 \dots R_{k-1} R_k \quad (36)$$

Si definimos

$$R = R_1 R_2 \dots R_{k-1} R_k \quad (37)$$

entonces $R^{-1}AR = D_k$, así que

$$AR = RD_k \approx R \text{diag}\{\lambda_1, \lambda_2, \dots, \lambda_n\} \quad (38)$$

Si expresamos las columnas de R como vectores: X_1, X_2, \dots, X_n , entonces podemos expresar R como un vector fila de vectores columna:

$$R = [X_1 \quad X_2 \quad \dots \quad X_n] \quad (39)$$

Esto nos permite escribir la relación (38) columna a columna:

$$[AX_1 \quad AX_2 \quad \dots \quad AX_n] \approx [\lambda_1 X_1 \quad \lambda_2 X_2 \quad \dots \quad \lambda_n X_n] \quad (40)$$

De las relaciones (39) y (40) se deduce que X_j , que es la j -ésima columna de R , es una aproximación del autovector correspondiente al autovalor λ_j .

El paso general

El objetivo en cada paso del método de Jacobi es reducir a cero dos de los términos simétricos que están fuera de la diagonal, digamos las que ocupan las posiciones (p,q) y (q,p) . Denotemos por R_1 la primera matriz ortogonal que vamos a usar, de manera que en la matriz

$$D_1 = R_1' A R_1 \tag{41}$$

los elementos d_{pq} y d_{qp} sean cero, y donde R_1 es de la forma

$$R_1 = \begin{bmatrix} 1 & \dots & 0 & \dots & 0 & \dots & 0 \\ \vdots & & \vdots & & \vdots & & \vdots \\ 0 & \dots & c & \dots & s & \dots & 0 \\ \vdots & & \vdots & & \vdots & & \vdots \\ 0 & \dots & -s & \dots & c & \dots & 0 \\ \vdots & & \vdots & & \vdots & & \vdots \\ 0 & \dots & 0 & \dots & 0 & \dots & 1 \end{bmatrix} \begin{matrix} \\ \\ \leftarrow \text{fila } p \\ \\ \leftarrow \text{fila } q \\ \\ \end{matrix} \tag{42}$$

$$\begin{matrix} & & \uparrow & & \uparrow & & \\ & & \text{col } p & & \text{col } q & & \end{matrix}$$

Aquí todos los elementos de R_1 que están fuera de la diagonal son cero excepto: el que vale s , colocado en la fila p y columna q , y el que vale $-s$, colocado en la fila q , columna p . Por otro lado, todos los elementos diagonales valen 1 excepto los dos que valen c , en las posiciones p -ésima y q -ésima de la diagonal. Esta matriz es la matriz de rotación plana, en la que se utilizó la notación $c = \cos \phi$ y $s = \sin \phi$.

Debemos verificar que la transformación (41) solo altera los elementos de las filas y las columnas p -ésima y q -ésima. Se considera la multiplicación de A por R_1 , o sea, el producto $B = AR_1$:

$$B = \begin{bmatrix} a_{11} & \cdots & a_{1p} & \cdots & a_{1q} & \cdots & a_{1n} \\ \vdots & & \vdots & & \vdots & & \vdots \\ a_{p1} & \cdots & a_{pp} & \cdots & a_{pq} & \cdots & a_{pn} \\ \vdots & & \vdots & & \vdots & & \vdots \\ a_{q1} & \cdots & a_{qp} & \cdots & a_{qq} & \cdots & a_{qn} \\ \vdots & & \vdots & & \vdots & & \vdots \\ a_{n1} & \cdots & a_{np} & \cdots & a_{nq} & \cdots & a_{nn} \end{bmatrix} \begin{bmatrix} 1 & \cdots & 0 & \cdots & 0 & \cdots & 0 \\ \vdots & & \vdots & & \vdots & & \vdots \\ 0 & \cdots & c & \cdots & s & \cdots & 0 \\ \vdots & & \vdots & & \vdots & & \vdots \\ 0 & \cdots & -s & \cdots & c & \cdots & 0 \\ \vdots & & \vdots & & \vdots & & \vdots \\ 0 & \cdots & 0 & \cdots & 0 & \cdots & 1 \end{bmatrix}$$

Aplicando la regla de multiplicación de matrices, filas de la primera por columnas de la segunda, observamos que no se producen cambios en las columnas primera a $(p-1)$ -ésima, $(p+1)$ -ésima a $(q-1)$ -ésima y $(q+1)$ -ésima a n -ésima, solo se alteran las columnas p -ésima y q -ésima:

$$\begin{aligned} b_{jm} &= a_{jm}, && \text{cuando } m \neq p \text{ y } m \neq q \\ b_{jp} &= ca_{jp} - sa_{jq}, && \text{para } j = 1, 2, \dots, n \\ b_{jq} &= sa_{jp} + ca_{jq}, && \text{para } j = 1, 2, \dots, n \end{aligned} \tag{43}$$

Un argumento parecido prueba que al multiplicar R_1^t por A solo se alteran las filas p -ésima y q -ésima. En consecuencia, la transformación

$$D_1 = R_1^t AR_1 \tag{44}$$

solo altera los elementos de las filas y las columnas p -ésima y q -ésima. Los elementos d_{jk} de la matriz D_1 que no son iguales que los correspondientes de A se calculan mediante las formulas

$$\begin{aligned}
 d_{jp} &= ca_{jp} - sa_{jq} && \text{para } j \neq p \text{ y } j \neq q, \\
 d_{jq} &= sa_{jp} + ca_{jq} && \text{para } j \neq p \text{ y } j \neq q, \\
 d_{pp} &= c^2 a_{pp} + s^2 a_{qq} - 2csa_{pq}, \\
 d_{qq} &= s^2 a_{pp} + c^2 a_{qq} + 2csa_{pq}, \\
 d_{pq} &= (c^2 - s^2)a_{pq} + cs(a_{pp} - a_{qq}),
 \end{aligned} \tag{45}$$

y los demás se hallan por simetría.

Como hacer d_{pq} y d_{qp} iguales a cero

El objetivo en cada paso del método de Jacobi es reducir a cero los elementos d_{pq} y d_{qp} que están fuera de la diagonal. La estrategia obvia es tomar

$$c = \cos \phi \text{ y } s = \sin \phi \tag{46}$$

siendo ϕ el ángulo de rotación que produce el efecto deseado. Sin embargo, para calcular este ángulo hay que realizar algunas manipulaciones ingeniosas con las identidades trigonométricas. Usando (46) para calcular la cotangente del ángulo doble obtenemos

$$\theta = \cot 2\phi = \frac{c^2 - s^2}{2cs} \tag{47}$$

Supongamos que $a_{pq} \neq 0$ y que deseamos obtener $d_{pq} = 0$. Entonces usando la última ecuación de (45), obtenemos

$$0 = (c^2 - s^2)a_{pq} + cs(a_{pp} - a_{qq}) \tag{48}$$

Ordenando un poco los términos nos queda $(c^2 - s^2)/(cs) = (a_{qq} - a_{pp})/a_{pq}$ que, junto con (47), nos permite calcular θ :

$$\theta = \frac{a_{qq} - a_{pp}}{2a_{pq}} \quad (49)$$

Aunque podemos usar las formula (49) junto con la formulas (46) y (47) para calcular c y s , puede probarse que se propaga menos error de redondeo si calculamos $\tan \phi$ y usamos ese valor en cálculos posteriores. Así pues tomamos

$$t = \tan \phi = \frac{s}{c} \quad (50)$$

Dividimos el numerador y el denominador de (47) entre c^2 obtenemos

$$\theta = \frac{1 - s^2/c^2}{2s/c} = \frac{1 - t^2}{2t}$$

que nos da la ecuación

$$t^2 + 2t\theta - 1 = 0 \quad (51)$$

Puesto que $t = \tan \phi$, la menor de la raíces de la ecuación (51) corresponde al menor al menor ángulo de rotación que podemos tomar y que cumple $|\phi| \leq \pi/4$. La forma especial que tiene la ecuación de segundo grado nos permite trabajar con la siguiente fórmula para hallar la menor de las raíces

$$t = -\theta \pm (\theta^2 + 1)^{1/2} = \frac{\text{sign}(\theta)}{|\theta| + (\theta^2 + 1)^{1/2}}, \quad (52)$$

Siendo $\text{sign}(\theta) = 1$ cuando $\theta \geq 0$ y $\text{sign}(\theta) = -1$ cuando $\theta < 0$. Un vez hallado t , los valores de c y s se calculan usando las formulas

$$c = \frac{1}{(t^2 + 1)^{1/2}} \text{ y } s = ct$$

Algoritmo de Jacobi

```

1: function [V,D]=jacobil(A,epsilon)
2:     D=A;
3:     [n,n]=size(A);
4:     V=eye(n);
5:     state=1;
6:     [m1 p]=max(abs(D-diag(diag(D)))));
7:     [m2 q]=max(m1);
8:     p=p(q);
9:     while (state==1)
10:         theta=(D(q,q)-D(p,p))/2*D(p,q);
11:         t=sign(theta)/(abs(theta)+sqrt(theta^2+1));
12:         c=1/sqrt(t^2+1);
13:         s=c*t;
14:         R=[c s;-s c];
15:         D([p q],:)=R'*D([p q],:);
16:         D(:,[p q])=D(:,[p q])*R;
17:         V(:,[p q])=V(:,[p q])*R;
18:         [m1 p]=max(abs(D-diag(diag(D)))));
19:         [m2 q]=max(m1);
20:         p=p(q);
21:         if (abs(D(p,q))<epsilon*sqrt(sum(diag(D).^2)/n));
22:             state=0;
23:         end
24:     end
25:     D=diag(diag(D));
26: End

```

Anexo 9: Método Leverrier-Laguerre

Este método para hallar autovalores está compuesto de dos etapas, la primera corresponde al método de Leverrier para determinar los coeficientes del polinomio característico y la segunda llamada método de Laguerre que permite determinar mediante los coeficientes las raíces de polinomio característico.

Método de Leverrier

Dada una matriz cuadrada A de orden $n \times n$. El polinomio característico de la matriz es

$$p(\lambda) = p_0 \lambda^n + p_1 \lambda^{n-1} + p_2 \lambda^{n-2} + \dots + p_{n-1} \lambda + p_n$$

Los coeficientes se hallan mediante las siguientes relaciones

$$\begin{aligned}
 p_0 &= 1 \\
 p_1 &= -s_1 \\
 p_2 &= -\frac{1}{2}(s_2 + p_1s_1) \\
 p_3 &= -\frac{1}{3}(s_3 + p_1s_2 + p_2s_1) \\
 p_4 &= -\frac{1}{4}(s_4 + p_1s_3 + p_2s_2 + p_3s_1) \\
 &\vdots \\
 p_n &= -\frac{1}{n}(s_n + p_1s_{n-1} + \dots + p_{n-1}s_1)
 \end{aligned}$$

Los valores s_1, s_2, \dots, s_n son las trazas de las potencias de la matriz cuadrada A

$$\begin{aligned}
 s_1 &= \text{traza de } A \\
 s_2 &= \text{traza de } A^2 \\
 s_3 &= \text{traza de } A^3 \\
 &\vdots \\
 s_n &= \text{traza de } A^n
 \end{aligned}$$

Luego, efectuando el cambio en las notaciones

$$p_0 = a_1, p_1 = a_2, \dots, p_n = a_{n+1}$$

Se tiene el siguiente polinomio de grado n dado por

$$p_n(x) = a_1x^n + a_2x^{n-1} + \dots + a_nx + a_{n+1},$$

donde los coeficientes a_i pueden ser números reales o complejos. Antes de presentar como abordar la resolución de este problema se presenta dos algoritmos auxiliares que son de gran utilidad.

Deflación de un polinomio

Si se ha calculado o se conoce una raíz r de $p_n(x) = 0$, se representa el polinomio por

$$p_n(x) = (x - r)p_{n-1}(x)$$

El haberlo hecho supone una deflación o división sintética. Si se consigue, las raíces del polinomio que quedan por calcular serían de las de $p_{n-1}(x)$. Una deflación gradual, que calcule consecutivamente una raíz de los $p_{n-1}(x)$ que se vayan obteniendo progresivamente es lo que conseguirá calcular todas las raíces del polinomio que se estudia. Si se tiene que

$$p_{n-1}(x) = b_1x^{n-1} + b_2x^{n-2} + \dots + b_{n-1}x + b_n$$

La ecuación anterior queda

$$a_1x^n + a_2x^{n-1} + \dots + a_nx + a_{n+1} = (x-r)(b_1x^{n-1} + b_2x^{n-2} + \dots + b_{n-1}x + b_n)$$

Igualando los coeficientes de la misma potencia de x en ambos miembros se obtiene que

$$b_1 = a_1, b_2 = a_2 + rb_1, \dots, b_n = a_n + rb_{n-1}$$

Lo que conduce al algoritmo de deflación de Horner que se indica el siguiente algoritmo:

```

1: function b=deflpoly(a,r)
2: n=length(a)-1;
3: b=zeros(n,1);
4: b(1)=a(1);
5: for i=2:n
6:     b(i)=a(i)+r*b(i-1);
7: end
8: end

```

Evaluación de un polinomio y sus derivadas

Recordemos la regla de Horner que para evaluar un polinomio $p_n(x)$ lo mejor es proceder secuencialmente

$$\begin{aligned}
 p_0(x) &= a_1 \\
 p_1(x) &= a_2 + xp_0(x) \\
 p_2(x) &= a_3 + xp_1(x) \\
 p_3(x) &= a_4 + xp_2(x) \\
 &\vdots \\
 p_n(x) &= a_{n+1} + xp_{n-1}(x)
 \end{aligned}$$

Es decir, para un polinomio de grado n

$$\begin{aligned}
 p_0(x) &= a_1 \\
 p_i(x) &= a_{i+1} + xp_{i-1}(x), \quad i = 1, 2, \dots, n
 \end{aligned}$$

Las derivadas primera y segunda saldrían de estas expresiones

$$\begin{aligned}
 p'_0(x) &= 0, & p'_i(x) &= p_{i-1}(x) + xp'_{i-1}(x), & i &= 1, 2, \dots, n \\
 p''_0(x) &= 0, & p''_i(x) &= 2p'_{i-1}(x) + xp''_{i-1}(x), & i &= 1, 2, \dots, n
 \end{aligned}$$

Si se escriben en código Scilab resulta lo que se indica a continuación:

```

1: function [p, dp, ddp]=evalpoly(a, x)
2:     n=length(a)-1;
3:     p=a(1);
4:     dp=0.0;
5:     ddp=0.0;
6:     for i=1:n
7:         ddp=ddp*x+2*dp;
8:         dp=dp*x+p;
9:         p=p*x+a(i+1);
10:    end
11: End

```

Método de Laguerre para determinar una raíz

Debe su nombre al trabajo de Edmond Laguerre, Francia (1834-1886). A continuación se describirá el método, para lo cual se considerara un caso especial de polinomio de grado n en el que una raíz es $x=r$ y las $n-1$ restantes es una raíz múltiple $x=q$, dicho polinomio podría expresarse así

$$p_n(x) = (x-r)(x-q)^{n-1}$$

Si derivamos su derivada con respecto a x se tiene que

$$p'_n(x) = (x-q)^{n-1} + (n-1)(x-r)(x-q)^{n-2} = p_n(x) \left(\frac{1}{x-r} + \frac{n-1}{x-q} \right)$$

Es decir,

$$\frac{p'_n(x)}{p_n(x)} = \frac{1}{x-r} + \frac{n-1}{x-q} \quad (53)$$

Si derivamos de nuevo,

$$\frac{p''_n(x)}{p_n(x)} - \left(\frac{p'_n(x)}{p_n(x)} \right)^2 = -\frac{1}{(x-r)^2} - \frac{n-1}{(x-q)^2} \quad (54)$$

Si introducimos la notación que sigue

$$g(x) = \frac{p'_n(x)}{p_n(x)} \text{ y } h(x) = g^2(x) - \frac{p''_n(x)}{p_n(x)}$$

Las ecuaciones (53) y (54) quedan

$$g(x) = \frac{1}{x-r} + \frac{n-1}{x-q}$$

$$h(x) = \frac{1}{(x-r)^2} + \frac{n-1}{(x-q)^2}$$

Si despejamos de la primera de estas dos ecuaciones $x-q$ y lo sustituimos con la segunda, obtenemos una función cuadrática de $x-r$, la solución de la cual es denominada fórmula de Laguerre. Tiene la siguiente expresión

$$x - r = \frac{n}{g(x) \pm \sqrt{(n-1)[nh(x) - g^2(x)]}}$$

```

1: function [x,i]=laguerre(a,tol)
2: x=randn;
3: n=length(a)-1;
4: for i=1:50
5:     [p,dp,ddp]=evalpoly(a,x);
6:     if abs(p)<tol, return, end
7:     g=dp/p;
8:     h=g*g-ddp/p;
9:     f=sqrt((n-1)*(n*h-g*g));
10:    if abs(g+f)>=abs(g-f)
11:        dx=n/(f+g);
12:    else
13:        dx=n/(g-f);
14:    end
15:    x=x-dx;
16:    if abs(dx)<tol,
17:        Return;
18:    end
19: end
20: error('tambien muchas iteraciones de Laguerre')
21: end
22: end

```

Algoritmo Leverrier-Laguerre

Finalmente, el algoritmo que permite encontrar los autovalores de una matriz cuadrada de orden n , está dado por:

```

1: function LAMBDA=laguerre_leverrier(A)
2: [n n]=size(A);
3: COEF=leverrier(A)
4: [root,ite]=polyroots(COEF,0.001);
5: LAMBDA=root;
6: End

```