

UNIVERSIDAD NACIONAL DE INGENIERÍA
FACULTAD DE CIENCIAS
ESCUELA PROFESIONAL DE MATEMÁTICA



**El Problema del Valor Propio Generalizado y
Algunas de sus Aplicaciones**

por

Marlene Cecilia Ruíz Díaz

Tesis para Optar
el Título Profesional de
LICENCIADO en MATEMÁTICA

Prof. William Carlos Echegaray Castillo
Asesor

UNI, 25 Noviembre del 2008

A mis padres

Agradezco al Profesor William Carlos Echegaray Castillo por la orientación y su paciencia para la culminación del presente trabajo y también deseo agradecer a todas aquellas personas que de una u otra forma me ayudaron a terminar este trabajo.

Resumen

En el capítulo 2: Damos a conocer conocimientos básicos que nos servirán para el desarrollo de este presente trabajo, además veremos problemas de estabilidad para ecuaciones diferenciales, estabilidad de un sistema no homogéneo, problema vibratorio, resultados clásicos en localización de valores propios, tales como el teorema del Disco Geršgorin y por último en este capítulo describimos el método de la potencia, método de la potencia inversa, la iteración Cociente Rayleigh el cual encuentra un número de valores propios y sus correspondientes vectores propios ya que es dificultoso de calcular los valores propios de una matriz vía el polinomio característico

En el capítulo 3: Desarrollamos el estudio de la sensibilidad de los valores propios y vectores propios, denotando el mas importante teorema llamado teorema de Bauer-Fike.

En el capítulo 4: Daremos a conocer el método de iteración QR, además modificaremos dicho método en el caso complejo y también mostramos sus aplicaciones.

En el capítulo 5: Estudiaremos el problema $Ax = \lambda Bx$ llamado el problema de valor propios generalizado, donde $A, B \in \mathbb{C}(n, n)$, $x \in \mathbb{C}^n$ y $\lambda \in \mathbb{C}$, en este trabajo mostramos los siguientes puntos:

- 1) El teorema de Schur para una matriz pencil.
- 2) Algoritmo QZ: Este algoritmo es ampliamente utilizado para el problema del valor propio generalizado, en la cual la forma generalizada de Schur es aplicada.

Indice general

1. Introducción	1
2. Conceptos Preliminares	3
2.1. Problemas de Estabilidad para Ecuaciones Diferenciales	10
2.2. Estabilidad de un Sistema No Homogéneo	12
2.3. Problema Vibratorio	15
2.4. Localización de los Valores Propios	15
2.5. Inercia de una Matriz	19
2.6. Importancia de los Valores propios y Vectores Propios Dominantes en un Sistema Dinámico	28
2.6.1. <i>Cálculo de valores propios</i>	29
2.6.2. Método de la Potencia con Desplazamiento	35
2.6.3. Método de la Potencia Inversa	36
2.6.4. El Cociente de Rayleigh	39
2.6.5. Iteración del Cociente de Rayleigh	40
2.6.6. Cálculo de los Valores Propios Subdominantes y Vectores Propios: Deflación	41
3. Sensibilidad de los Valores Propios	46
3.1. Implicaciones del Teorema	47
3.2. Sensibilidad de los Valores Propios Individuales	48
4. Iteración QR	51
4.1. Iteración Básica QR	51

4.2. Iteración QR de Hessenberg	54
4.3. Desplazamiento Simple en la Iteración QR	55
4.4. El Doble Desplazamiento de la Iteración QR	58
5. El Problema de Valor Propio Generalizado	64
5.1. Introducción	64
5.1.1. Valores Propios de una Matriz Pencil	65
5.1.2. Descomposición Generalizada de Schur	66
5.1.3. Algoritmo QZ	67
5.1.4. Reducción a una Forma Hessenberg - Triangular .	67
5.2. El Problema de Valor Propio Generalizado Definido Simétrico	78
5.2.1. Ortogonalidad de los Vectores Propios	79
5.3. El Método QZ para el Pencil Definida Simétrica .	79
5.3.1. El Algoritmo Cholesky QR	79
6. Algunas Aplicaciones	83
7. Conclusiones de la Tesis	90
A. Programas elaborados	92
A.1. Total .	92
A.2. Paso1	100
A.3. Mgivens	101
A.4. Pgivens	102
A.5. Paso2	102
A.6. PQA	103
A.7. QA .	104
A.8. PBZ	105
A.9. BZ .	106
A.10. Minversa .	106
A.11. Paso3 .	108
A.12. Nuevo9 .	108
A.13. Xgivens	110

A.14. Ygивens	111
A.15. ST	112
A.16. Vectores Propios-QZ	113
A.17. Gauss-Jordan	114
A.18. CholeskyQR	116
A.19. Cholesky	118
A.20. Hessenberg	119
A.21. QRsimetrico	120
A.22. Householder	121
A.23. Signo	122
A.24. Vectores Propios	123
A.25. Tridiagonal	124
Bibliografía	125

Indice de figuras

6.1. Sistema Masa - Resorte	84
6.2. Edificio de 4 Pisos	87
6.3. Primer Modo de Vibración de un Edificio de 4 Pisos	88
6.4. Segundo Modo de Vibración de un Edificio de 4 Pisos	89

Capítulo 1

Introducción

En el presente trabajo damos a conocer algunos algoritmos que nos permiten determinar los **valores propios generalizados** del sistema $Ax = \lambda Bx$, donde $A, B \in \mathbb{R}(n, n)$, y como consecuencia ha sido desarrollado de la siguiente manera:

En el capítulo 2 damos a conocer los conceptos básicos que nos permitan usarlos como herramientas para el desarrollo de los siguientes capítulos.

Además vemos aplicaciones del problema $Ax = \lambda x$, donde $A \in \mathbb{R}(n, n)$, $x \in \mathbb{R}^n$ llamado vector propio de A asociado al valor propio λ , también la importancia de los valores propios dominantes, damos además a conocer algunos métodos de cálculo de valores y vectores propios, como por ejemplo el método de la potencia. También vemos el cociente de Rayleigh $R_q = \frac{x^T Ax}{x^T x}$ ($x \neq 0$) el cual nos permita dar una estimación del valor propio λ a partir de la estimación del correspondiente vector propio.

En el capítulo 3 vemos la sensibilidad de los valores propios de una matriz, el cual nos permite determinar si la matriz está bien condicionada o no.

En el capítulo 4 desarrollamos algunos métodos de iteración QR, el cual nos permite calcular los valores propios de una matriz. Para una matriz real, el algoritmo básicamente construye iterativamente la forma real de Schur de la matriz por

similaridad ortogonal. La convergencia de la iteración del algoritmo Hessenberg QR simple, puede ser lenta por la presencia de algún valor propio múltiple ó la presencia de valores propios complejos a esto es que estudiamos la iteración QR con cambio simple o cambio doble para esos casos.

En el capítulo 5 vemos el problema del valor propio generalizado $Ax = \lambda BX$, donde $A, B \in \mathbb{R}(n, n)$, que con ayuda del algoritmo QZ que es una generalización del algoritmo de Schur de (A, B) . Este algoritmo está formado por dos etapas:

En la etapa 1, el par (A, B) es reducido a un par de Hessenberg triangular.

En la etapa 2, el par de Hessenberg triangular es reducido a la forma generalizada real de Schur, usando iteración implícita QR para AB^{-1} donde no es necesario hallar la matriz B^{-1} . Además vemos un caso particular llamado el Problema del valor propio definido simétrico, el cual se presenta en diversos problemas de ingeniería estructural y de vibración de la forma $Kx = \lambda MX$, donde M es una matriz simétrica definida positiva y K es una matriz simétrica. El algoritmo utilizado para este tipo de problema es el algoritmo de Cholesky.

Capítulo 2

Conceptos Preliminares

Presentamos algunos conceptos y notaciones que nos servirán para el desarrollo del presente trabajo.

1. Consideremos $x \in \mathbb{C}^n$ un vector columna
2. Definamos $\mathbb{C}(m, n) = \{A/A \text{ es una matriz compleja de orden } m \times n\}$
Sea $A \in \mathbb{C}(m, n)$, entonces su transpuesta es denotado por A^* y es definido por:

$$A^* = (\overline{A})^T.$$

3. Diremos que $A \in \mathbb{C}(n, n)$ es hermitiana si $A^* = A$.
4. Una matriz hermitiana $A \in \mathbb{C}(n, n)$ es definida positiva si, y solo si, $x^*Ax > 0 \quad \forall x \in \mathbb{C}^n \setminus \{0\}$.
5. Una matriz $A \in \mathbb{C}(n, n)$ es unitaria si $A^*A = AA^* = I$.

Definición 2.0.1. Diremos que una matriz $A \in \mathbb{C}(n, n)$

1. es **triangular superior** si $a_{ij} = 0 \quad \forall i > j$
2. es **triangular inferior** si $a_{ij} = 0 \quad \forall i < j$

Definición 2.0.2. Sea $A \in \mathbb{C}(n, n)$, entonces $\lambda \in \mathbb{C}$ es un **valor propio** de A si existe $x \in \mathbb{C}^n \setminus \{0\}$ tal que $Ax = \lambda x$ donde x es llamado **vector propio** de A asociado con el

valor propio λ . $P_A(\lambda) = \det(A - \lambda I)$ es llamado el **polinomio característico de A**.
 Si $P_A(\lambda) = 0$ entonces las raíces de $P_A(\lambda)$ son los valores propios de la matriz A .

Definición 2.0.3. El conjunto de valores propios de una matriz A es llamado el **espectro de A** y lo denotamos $\sigma(A)$.

Teorema 2.0.1. Los valores propios de una matriz triangular (superior ó inferior) son los elementos de la diagonal.

Prueba: [Ver ([2])]

Teorema 2.0.2. Sea: $A =$

$$\begin{pmatrix} A_{11} & A_{12} & \cdot & \cdot & \cdot & A_{1k} \\ 0 & A_{22} & \cdot & \cdot & \cdot & A_{2k} \\ 0 & 0 & \cdot & \cdot & \cdot & A_{3k} \\ \cdot & \cdot & \cdot & \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdot & \cdot & \cdot & \cdot \\ 0 & 0 & & & & A_{kk} \end{pmatrix}$$

donde cada A_{ii} es una matriz cuadrada. Entonces el espectro de A es igual a la unión de los espectros de A_{11}, \dots, A_{kk} .

Prueba:[Ver ([2])]

Teorema 2.0.3. Si dos matrices son similares, entonces tienen los mismos valores propios.

Prueba:[Ver ([2])]

Observación Del teorema anterior, no se cumple la otra implicancia, es decir, si dos matrices tienen los mismos valores propios no implica que sean similares, como lo

muestra el siguiente ejemplo.

Ejemplo. Sean:

$$A = \begin{pmatrix} 1 & 2 \\ 0 & 1 \end{pmatrix} \quad B = \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix}$$

los valores propios de A son 1 de multiplicidad 2, los valores propios de B son 1 de multiplicidad 2, veamos ahora que no son similares.

Supongamos que A y B sean similares, entonces existe T no singular tal que:

$$\begin{pmatrix} 1 & 2 \\ 0 & 1 \end{pmatrix} = A = T^{-1}BT = T^{-1}IT = I = \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix}.$$

lo cual es una contradicción, por lo tanto A y B no son similares a pesar de tener los mismos valores propios.

Teorema 2.0.4. *Sea $A \in \mathbb{C}(n, n)$ tal que posee n valores propios distintos, entonces los correspondientes vectores propios son linealmente independiente.*

Prueba:[Ver ([13])]]

Teorema 2.0.5. *Una matriz arbitraria A es similar a la matriz diagonal D si y sólo si A tiene un conjunto de vectores propios linealmente independientes.*

Prueba:[Ver ([13])]]

Corolario 2.0.1. *Si todos los valores propios de A son distintos, entonces A es similar a la matriz diagonal.*

Prueba:[Ver ([13])]]

Como todos los valores propios de A son distintos, entonces por el teorema (2.0.4) sus correspondientes vectores propios son linealmente independientes, y finalmente por el teorema (2.0.5), A es similar a una matriz diagonal.

Teorema 2.0.6 (Teorema de Triangularización de Schur). Para cualquier matriz compleja A de orden $n \times n$ existe una matriz unitaria U tal que

$$U^*AU = T \quad (2.1)$$

es una matriz triangular superior. Los valores propios de A son los elementos de la diagonal de T .

Prueba:

Probaremos usando inducción sobre n :

Para $n = 1$ el teorema es verdadero.

Supongamos que para $(n - 1)$ el teorema es verdadero, $n \geq 2$.

Veamos que para n el teorema también es cierto:

Sea u un vector propio de A con $\|u\|_2 = 1$ asociado al valor propio λ_1 . Entonces podemos elegir una matriz V de orden $n \times (n - 1)$ tal que:

$$U_1 = (u, V)$$

sea unitaria.

Como $AU_1 = A(u, V) = (Au, AV) = (\lambda_1 u, AV)$ tenemos:

$$A_1 = U_1^*AU_1 = \begin{pmatrix} u^* \\ V^* \end{pmatrix} (\lambda_1 u, AV) = \left(\begin{array}{c|ccc} \lambda_1 & * & * & * \\ \hline 0 & & & \\ \vdots & & \hat{A} & \\ 0 & & & \end{array} \right) \quad (2.2)$$

donde $\hat{A} = V^*AV$ es una matriz de orden $(n - 1) \times (n - 1)$.

Se observa que los valores propios de A y \hat{A} son los mismos, excepto λ_1 . Por hipótesis existe una matriz unitaria V_1 de orden $(n - 1) \times (n - 1)$ tal que:

$$\hat{T} = V_1^* \hat{A} V_1$$

sea triangular superior.

Luego, definimos

$$U_2 = \left(\begin{array}{c|ccc} 1 & 0 & 0 & 0 \\ \hline 0 & & & \\ \vdots & & V_1 & \\ 0 & & & \end{array} \right)$$

Como V_1 es unitaria, entonces U_2 es unitaria y

$$\begin{aligned} U_2^* A_1 U_2 &\stackrel{(2.2)}{=} U_2^* U_1^* A U_1 U_2 \\ &= U^* A U \end{aligned}$$

donde $U = U_1 U_2$.

Luego,

$$U^* A U = \left(\begin{array}{c|ccc} \lambda_1 & * & \cdots & * \\ \hline 0 & & & \\ \vdots & & & \\ 0 & & & \end{array} \right) = T$$

donde \hat{T} es triangular superior, entonces $U^* A U$ también es triangular superior.

Como los valores propios de una matriz triangular son los elementos de su diagonal que son justamente los valores propios de A . ■

Teorema 2.0.7. Sean $\lambda_1, \dots, \lambda_n$ los valores propios de una matriz $A \in \mathbb{C}(n, n)$, entonces los valores propios de A^m son $\lambda_1^m, \dots, \lambda_n^m$. En general, si P es un polinomio de grado n , es decir:

$$P(x) = c_0 x^n + c_1 x^{n-1} + \dots + c_n$$

entonces los valores propios de $P(A)$ son $P(\lambda_1), \dots, P(\lambda_n)$.

Prueba:[Ver ([2])]

Teorema 2.0.8. Sea A una matriz hermitiana. Entonces:

1. Existe una matriz unitaria U tal que $U^* A U = D$
2. Los valores propios de A son reales
3. Los vectores propios de A pueden ser escogidos ortonormales.

Prueba:[Ver ([14])]

Corolario 2.0.2. Si A es una matriz simétrica real, entonces existe una matriz ortogonal U tal que $U^T A U = D$, donde D es diagonal.

Prueba:[Ver ([18])]

Teorema 2.0.9. Una matriz hermitiana A es definida positiva si, y sólo si, sus valores propios son positivos.

Prueba:

(\Rightarrow) Sea λ un valor propio de A , entonces $\exists x \neq 0$ tal que $Ax = \lambda x$.

$$\Rightarrow x^* Ax = \lambda x^* x \quad (2.3)$$

como A es hermitiana definida positiva, entonces $x^* Ax > 0$ y además

$$x^* x = |x_1|^2 + \dots + |x_n|^2 > 0$$

donde $x = (x_1, \dots, x_n)^T \in \mathbb{R}^n$. Luego, de (2.3) se tiene:

$$\lambda = \frac{x^* Ax}{x^* x} > 0 \Rightarrow \lambda > 0$$

(\Leftarrow) Como A es hermitiana, del teorema (2.0.8) tenemos que existe una matriz unitaria U tal que:

$$U^* A U = D = \text{diag}(\lambda_1, \dots, \lambda_n) \quad (2.4)$$

Sea $x \in \mathbb{R}^n \setminus \{0\}$ cualquiera, definamos $y \in \mathbb{R}^n / y = U^* x$. Entonces $y \neq 0$. Ahora, reemplazando en (2.4) se obtiene:

$$x^* Ax = x^* U D U^* x = y^* D y = \sum_{j=1}^n \lambda_j |y_j|^2 > 0$$

para cualquier $x \neq 0$, lo cual prueba que A es definida positiva. ■

Teorema 2.0.10 (Forma Canónica de Jordan). Para cualquier matriz $A \in \mathbb{C}(n, n)$ existe una matriz no singular X tal que:

$$X^{-1} A X = \begin{pmatrix} J_1 & & 0 \\ & \ddots & \\ 0 & & J_m \end{pmatrix}$$

donde:

$$J_i = \begin{pmatrix} \lambda_i & 1 & 0 & \dots & 0 \\ & \lambda_i & 1 & \dots & \\ & & \ddots & \ddots & \\ & & & \ddots & 1 \\ 0 & & & & \lambda_i \end{pmatrix}$$

donde los λ_i son los valores propios de A .

Prueba: ([ver [9] pag. 121 - 129])

Observación

Los bloques diagonales anteriores es llamada *la Forma Canónica de Jordan*. Las matrices J_1, \dots, J_m son matrices de Jordan. Si J_i es de grado v_i entonces:

$$v_1 + \dots + v_m = n$$

Definición 2.0.4. *Los polinomios:*

$$\det(J_i - \lambda I) = (\lambda_i - \lambda)^{v_i}$$

son llamados los **divisores elementales** de A . Si $v_1 = v_2 = \dots = v_m = 1$ entonces, los divisores elementales son llamados **divisores elementales lineales**.

Si la i th- columna de X es denotado por x_i entonces los vectores x_1, x_2, \dots, x_{v_1} tenemos que:

$$\begin{aligned} Ax_1 &= \lambda_1 x_1 \\ Ax_{i+1} &= \lambda_1 x_{i+1} + x_i, \quad \forall i = 1, \dots, v_1 - 1 \end{aligned}$$

Relaciones similares cumplen otros vectores. Estos vectores son llamados **vectores principales** o **vectores propios generalizados** de A .

Observaciones 1.

1. Cuando todos los valores propios de A son distintos, cada bloque de Jordan tiene orden 1, además, los divisores elementales son lineales. En efecto, si existen m bloques de Jordan, entonces hay m vectores propios linealmente independientes.

concluyendo así que λ es un valor propio de A y v su correspondiente vector propio.

Si A tiene n vectores propios linealmente independientes entonces la solución general del sistema sería:

$$x(t) = c_1 v_1 e^{\lambda_1 t} + c_2 v_2 e^{\lambda_2 t} + \dots + c_n v_n e^{\lambda_n t}$$

donde $\lambda_1, \dots, \lambda_n$ son los valores propios de A y v_1, \dots, v_n son los correspondientes vectores propios.

En general, la solución general de (2.6) con $x(0) = x_0$ es dado por:

$$x(t) = e^{At} x_0$$

donde:

$$e^{At} = I + At + \frac{A^2 t^2}{2!} + \dots + \frac{A^n t^n}{n!} + \dots \quad (2.7)$$

Definición 2.1.1. Una solución x_e del sistema

$$\dot{x}(t) = Ax(t), \quad x(0) = x_0$$

es el vector x_e llamado de equilibrio si

$$Ax_e = 0$$

Claramente, $x_e = 0$ es una solución de equilibrio y es la única solución de equilibrio si, y sólo si A es no singular.

Definición 2.1.2. Una solución de equilibrio x_e es llamada **estable** si, $\forall \epsilon > 0$, existe un número real $\delta > 0$ tal que $\|x(t) - x_e\| < \epsilon$ siempre que $\|x_0 - x_e\| < \delta$.

Definición 2.1.3. Una solución de equilibrio x_e es **asintóticamente estable** si es estable y existe un $\delta > 0$ tal que $\|x(t) - x_e\| \rightarrow 0$ cuando $t \rightarrow \infty$, siempre que $\|x_0 - x_e\| < \delta$.

Definición 2.1.4. El sistema (2.6) es **asintóticamente estable** si la solución de equilibrio x_e es **asintóticamente estable**.

Definición 2.1.5. Un sistema es llamado **marginalmente estable**, si es estable pero no **asintóticamente estable**.

Definición 2.1.6. Un sistema que no es estable, es llamado inestable.

Teorema 2.1.1. [Teorema de Estabilidad Para un Sistema de Ecuaciones Diferenciales Homogéneo]

1. La solución de equilibrio $x_e = 0$ de (2.6) es asintóticamente estable si y sólo si los valores propios de A tiene parte real negativa.
2. Una solución de equilibrio es inestable si al menos uno de los valores propios tiene parte real positiva.

Prueba:[Por Ver ([2])]

2.2. Estabilidad de un Sistema No Homogéneo

Sea el sistema

$$\dot{x}(t) = Ax(t) + b \quad (2.8)$$

donde b es una constante.

Definiendo:

$$z(t) = x(t) - \bar{x}(t) \quad (2.9)$$

entonces:

$$\begin{aligned} \dot{z}(t) &= \dot{x}(t) - \dot{\bar{x}}(t) \\ \dot{z}(t) &= Ax(t) + b - A\bar{x}(t) - b \\ \dot{z}(t) &= A(x(t) - \bar{x}(t)) \end{aligned}$$

por (2.9) tenemos: $\dot{z} = A(z(t))$ tendríamos un problema similar a (2.6).

Si $x(t) \rightarrow \bar{x}(t) \Leftrightarrow z(t) \rightarrow 0$

Teorema 2.2.1. [Teorema de Estabilidad para un Sistema No Homogéneo de Ecuaciones Diferenciales] Con el cambio de variables tendríamos a (2.6), por lo tanto se cumplen las mismas propiedades anteriores, es decir:

1. Una solución de equilibrio de (2.8) es asintóticamente estable si y sólo si todos los valores propios de A tienen parte real negativa.
2. Una solución de equilibrio es inestable si al menos un valor propio tiene parte real positiva.

El siguiente modelo matemático fue dado por Richardson y es conocido como Modelo de Richardson.

Ejemplo. Una competición de armamentos europeos.

Consideremos la carrera armamentista de 1909 - 1914 entre dos alianzas europeas:

Alianza 1: Francia y Rusia

Alianza 2: Alemania y Austria-Hungría.

Las dos alianzas se enfrentaron. Vamos intentar explicar este hecho histórico con la noción de estabilidad. Primero, consideremos el siguiente modelo matemático de guerra entre los dos países:

$$\begin{aligned} \frac{dx_1}{dt} &= k_1 x_2 - \alpha_1 x_1 + g_1 \\ \frac{dx_2}{dt} &= k_2 x_1 - \alpha_2 x_2 + g_2 \end{aligned} \quad (2.10)$$

donde:

$x_i(t)$ es el potencial de guerra del país i , $i = 1, 2$.

$g_i(t)$ son los ataques de la alianza i ($i = 1, 2$) hacia la alianza j ($j = 1, 2$), con $i \neq j$.

g_i, α_i y k_i , ($i = 1, 2$) son constantes positivas.

$\alpha_i x_i$ es el costo de armamento del país i .

El sistema (2.10) puede ser escrito como

$$\dot{x}(t) = Ax(t) + g$$

donde:

$$A = \begin{pmatrix} -\alpha_1 & k_1 \\ k_2 & -\alpha_2 \end{pmatrix}, \quad x(t) = \begin{pmatrix} x_1(t) \\ x_2(t) \end{pmatrix}, \quad g = \begin{pmatrix} g_1 \\ g_2 \end{pmatrix}$$

Hallemos los valores propios de A :

$$P_A(\lambda) = \begin{vmatrix} -\alpha_1 - \lambda & k_1 \\ k_2 & -\alpha_2 - \lambda \end{vmatrix} = (-\alpha_1 - \lambda)(-\alpha_2 - \lambda) - k_1 k_2$$

$$\Rightarrow P_A(\lambda) = \lambda^2 + (\alpha_1 + \alpha_2)\lambda + \alpha_1\alpha_2 - k_1k_2$$

$$\Rightarrow \lambda = \frac{-(\alpha_1 + \alpha_2) \pm \sqrt{(\alpha_1 + \alpha_2)^2 - 4(\alpha_1\alpha_2 - k_1k_2)}}{2}$$

Luego:

$$\lambda_1 = \frac{-(\alpha_1 + \alpha_2) + \sqrt{(\alpha_1 + \alpha_2)^2 - 4(\alpha_1\alpha_2 - k_1k_2)}}{2}$$

$$\lambda_2 = \frac{-(\alpha_1 + \alpha_2) - \sqrt{(\alpha_1 + \alpha_2)^2 - 4(\alpha_1\alpha_2 - k_1k_2)}}{2}$$

1. Si $\alpha_1\alpha_2 - k_1k_2 > 0$

a) Si $\lambda_1, \lambda_2 \in \mathbb{R}$

$$\Rightarrow -4(\alpha_1\alpha_2 - k_1k_2) < 0$$

$$0 < (\alpha_1 + \alpha_2)^2 - 4(\alpha_1\alpha_2 - k_1k_2) < (\alpha_1 + \alpha_2)^2$$

entonces:

$$0 < \sqrt{\frac{(\alpha_1 + \alpha_2)^2 - 4(\alpha_1\alpha_2 - k_1k_2)}{2}} < (\alpha_1 + \alpha_2)$$

$$\Rightarrow -(\alpha_1 + \alpha_2) < 0$$

$$\Rightarrow \lambda_1, \lambda_2 < 0 \Rightarrow x(t) \text{ asintóticamente estable}$$

b) Si $\lambda_1, \lambda_2 \in \mathbb{C} \Rightarrow -(\alpha_1 + \alpha_2) < 0$ Por lo tanto $x(t)$ es asintóticamente estable.

2. Si $\alpha_1\alpha_2 - k_1k_2 < 0$

$$\Rightarrow (\alpha_1 + \alpha_2)^2 - 4(\alpha_1\alpha_2 - k_1k_2) > 0$$

Además:

$$(\alpha_1 + \alpha_2)^2 - 4(\alpha_1\alpha_2 - k_1k_2) > (\alpha_1 + \alpha_2)^2$$

$$\Rightarrow (\alpha_1 + \alpha_2) < \sqrt{(\alpha_1 + \alpha_2)^2 - 4(\alpha_1\alpha_2 - k_1k_2)}$$

$$\Rightarrow 0 < -(\alpha_1 + \alpha_2) + \sqrt{(\alpha_1 + \alpha_2)^2 - 4(\alpha_1\alpha_2 - k_1k_2)} = \lambda_1$$

entonces $\lambda_1 > 0$. Por lo tanto $x(t)$ es inestable.

2.3. Problema Vibratorio

El problema de vibración de la forma:

$$B\ddot{y} + Ay = 0 \quad (2.11)$$

donde

$$\ddot{y} = \left(\ddot{y}_1(t), \ddot{y}_2(t), \dots, \ddot{y}_n(t) \right)^T$$

Asumamos que la solución es de la forma $y = xe^{i\omega t}$ que reemplazando en (2.11) tenemos:

$$\omega^2 Bx = Ax$$

Sea $\lambda = \omega^2$

$$\Rightarrow Ax = \lambda Bx \quad (2.12)$$

Esta ecuación es llamada **Problema de valor propio generalizado**, λ es llamado valor propio y x es el correspondiente vector propio del problema de valor propio generalizado.

En un problema de vibración, las matrices B y A son llamadas matrices de *masa* y de *rigidez* respectivamente, y son denotadas por M y K . Reemplazando en (2.12) se obtiene:

$$Kx = \lambda Mx$$

Definición 2.3.1. Las cantidades $\omega_i = \sqrt{\lambda_i}$, $i = 1, \dots, n$ son llamados **modos de vibración del sistema**.

2.4. Localización de los Valores Propios

En varias aplicaciones prácticas es importante conocer la región (regiones) donde se encuentra (encuentran) un (algunos) valor propio (valores propios) o dada alguna región saber cuantos valores propios se encuentran en dicha región. Existen varias formas de localizar los valores propios de una determinada matriz A de orden $n \times n$. Comenzaremos con el método de Geršgorin.

Teorema 2.4.1. [Primer Teorema de Geršgorin] Sea $A \in \mathbb{C}(n, n)$ una matriz. Definamos

$$r_i = \sum_{\substack{j=1 \\ j \neq i}}^n |a_{ij}|, \quad i = 1, \dots, n$$

Entonces cada valor propio λ de A satisface al menos una de las siguientes desigualdades:

$$|\lambda - a_{ii}| \leq r_i, \quad i = 1, \dots, n$$

Prueba:[Ver ([11])]

Definición 2.4.1. Los discos $R_i = \{z/|z - a_{ii}| \leq r_i\}$, $i = 1, \dots, n$ son llamados discos de Geršgorin en el plano complejo.

Ejemplo. Sea:

$$A = \begin{pmatrix} 1 & 2 & 3 \\ 3 & 4 & 9 \\ 1 & 1 & 1 \end{pmatrix}$$

Entonces tenemos: $r_1 = \sum_{\substack{j=1 \\ j \neq 1}}^3 |a_{1j}| = 5, \quad a_{11} = 1$

$$r_2 = \sum_{\substack{j=1 \\ j \neq 2}}^3 |a_{2j}| = 12, \quad a_{22} = 4$$

$$r_3 = \sum_{\substack{j=1 \\ j \neq 3}}^3 |a_{3j}| = 2, \quad a_{33} = 1$$

Los discos de Geršgorin son:

$$R_1 = \{z/|z - 1| \leq 5\}$$

$$R_2 = \{z/|z - 4| \leq 12\}$$

$$R_3 = \{z/|z - 1| \leq 2\}$$

□

Teorema 2.4.2 (Segundo Teorema de Geršgorin). Supongamos que se tiene $R_i = \{z/|z - a_{ii}| \leq r_i\}$, $r_i = 1, 2, \dots, r$ discos de Geršgorin tales que $R_i \cap R_j = \emptyset$, $\forall i \neq j$, entonces el valor propio de A , $\lambda_i \in R_i \forall i = 1, 2, \dots, r$.

◦ **Prueba:**[Ver ([2])]

Ejemplo. Sea $A = \begin{pmatrix} 1 & 0.1 & 0.2 \\ 0.2 & 4 & 0.3 \\ 0.4 & 0.5 & 8 \end{pmatrix}$

$$\text{Entonces } R_1 = \{z/|z - 1| \leq 0.3\}$$

$$R_2 = \{z/|z - 4| \leq 0.5\}$$

$$R_3 = \{z/|z - 8| \leq 0.9\}$$

notar que $R_i \cap R_j = \emptyset \forall i \neq j, i, j = 1, 2, 3$, además los valores propios de A son:

$$\lambda_1 = 0.98336253767999 \in R_1,$$

$$\lambda_2 = 3.96709236337626 \in R_2,$$

$$\lambda_3 = 8.04954509894374 \in R_3. \quad \square$$

Además podemos localizar los valores propios de una matriz cuadrada A de orden $n \times n$ mediante los siguientes teoremas:

Teorema 2.4.3. *Sea λ un valor propio de una matriz A . Entonces se tiene*

$$|\lambda| \leq \|A\|.$$

En particular $\rho(A) \leq \|A\|$

Prueba: Sea $x \in \mathbb{R}^n \setminus \{0\}$ un vector propio asociado a λ ,

$$\text{entonces } Ax = \lambda x$$

$$\implies \|Ax\| = \|\lambda x\| = |\lambda| \|x\| \quad (2.13)$$

$$\text{Como } \|Ax\| \leq \|A\| \|x\|$$

$$\xrightarrow{(2.13)} |\lambda| \|x\| \leq \|A\| \|x\|$$

$$\xrightarrow{\|x\| \neq 0} |\lambda| \leq \|A\|$$

$$\implies \max_{\lambda \text{ v.p. } A} (|\lambda|) = \rho(A) \leq \|A\| \quad \blacksquare$$

Corolario 2.4.1. $\rho(A) \leq \|A^T\|$

Prueba:[Ver ([2])]

Teorema 2.4.4.

$$\rho(A) \leq \min \left(\left\{ \max_{1 \leq i \leq n} \left(\left\{ \sum_{j=1}^n |a_{ij}| \right\} \right) \right\}, \max_{1 \leq j \leq n} \left(\left\{ \sum_{i=1}^n |a_{ij}| \right\} \right) \right)$$

Prueba:

$$\text{Sabemos que } \|A\|_\infty = \max_{1 \leq i \leq n} \left(\sum_{j=1}^n |a_{ij}| \right)$$

$$\|A^T\|_\infty = \max_{1 \leq j \leq n} \left(\sum_{i=1}^n |a_{ij}| \right)$$

Como $\rho(A) \leq \|A\|_\infty$ por el teorema 2.4.3 y

$$\rho(A^T) \leq \|A^T\|_\infty \text{ por el corolario 2.4.1}$$

$$\implies \rho(A) \leq \min(\{\|A\|_\infty, \|A^T\|_\infty\})$$

Teorema 2.4.5. Sean $\lambda_1, \dots, \lambda_n$ los valores propios de A , entonces

$$\sum_{i=1}^n |\lambda_i|^2 \leq \|A\|_F^2$$

donde:

$$\|A\|_F = \left(\sum_{j=1}^n \sum_{i=1}^n |a_{ij}|^2 \right)^{\frac{1}{2}}$$

llamada Norma de Frobenius.

Prueba:

Por el teorema (2.0.6) de triangularización existe una matriz unitaria U tal que:

$$U^*AU = T$$

donde T es una matriz triangular superior

$$\begin{aligned} \implies TT^* &= (U^*AU)(U^*AU)^* \\ &= U^*AUU^*A^*U \\ &= U^*AA^*U \end{aligned}$$

entonces AA^* es una matriz semejante unitaria a TT^* .

Además, se sabe que las matrices semejantes tienen las mismas trazas, entonces

$$\text{tr}(TT^*) = \text{tr}(AA^*) = \|A\|_F^2$$

Como T es una matriz triangular superior, entonces se tiene:

$$\begin{aligned} \implies \text{tr}(TT^*) &= \sum_{i=1}^n \sum_{j=1}^n |t_{ij}|^2 \\ \implies \sum_{i=1}^n \sum_{j=1}^n |t_{ij}|^2 &= \|A\|_F^2 \end{aligned}$$

Además, los elementos de la diagonal de T son los valores propios de A

$$\begin{aligned}
 \text{entonces} \quad & \sum_{i=1}^n |t_{ii}|^2 = \sum_{i=1}^n |\lambda_i|^2 \\
 \implies & \sum_{i=1}^n |\lambda_i|^2 = \sum_{i=1}^n |t_{ii}|^2 \leq \sum_{i=1}^n \sum_{j=1}^n |t_{ij}|^2 = \|A\|_F^2 \\
 \implies & \sum_{i=1}^n |\lambda_i|^2 \leq \|A\|_F^2
 \end{aligned}$$

■

2.5. Inercia de una Matriz

Hemos visto anteriormente que la estabilidad del sistema:

$$\begin{cases} \dot{X}(t) = AX(t) \\ X(0) = X_0 \end{cases} \quad (2.14)$$

es conocido si los valores propios de la matriz A tienen parte real negativa.

Un problema más general que el problema de estabilidad es el problema de la inercia.

La inercia de $A \in \mathbb{C}(n, n)$, con respecto a los semiplanos es definido por:

$$In(A) = (\pi(A), \nu(A), \delta(A)) \quad (2.15)$$

donde:

$\pi(A)$ = número de valores propios de A con parte real positiva.

$\nu(A)$ = número de valores propios de A con parte real negativa.

$\delta(A)$ = número de valores propios de A con parte real nula.

Entonces, A es una matriz estable si $In(A) = (0, n, 0)$.

Teorema 2.5.1 (Ley de Inercia de Sylvester). Si $A \in \mathbb{R}(n, n)$ es simétrica y

$X \in \mathbb{R}(n, n)$ no singular, entonces $In(A) = In(X^T A X)$

Antes de demostrar este teorema veamos previamente lo siguiente:

Sabemos que por el teorema de Schur en el caso real, para $A \in \mathbb{R}(n, n)$ simétrica, existe

una matriz ortogonal T tal que:

$$T^T A T = \begin{pmatrix} \lambda_1 & 0 & \dots & \dots & 0 \\ 0 & \lambda_2 & \dots & \dots & 0 \\ \vdots & \vdots & \ddots & \ddots & \vdots \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ 0 & 0 & \dots & \dots & \lambda_n \end{pmatrix} = \Lambda \quad (2.16)$$

donde $\lambda_1, \lambda_2, \dots, \lambda_n$ son los valores propios de A .

Sea $x = Ty$ donde $y = (y_1, y_2, \dots, y_n)^T$, entonces:

$$\langle x, Ax \rangle = \langle Ty, ATy \rangle = \langle y, T^T A T y \rangle = \langle y, \Lambda y \rangle \quad (2.17)$$

además, se tiene que:

$$\langle y, \Lambda y \rangle = \lambda_1 y_1^2 + \lambda_2 y_2^2 + \dots + \lambda_n y_n^2 \quad (2.18)$$

supongamos que $\lambda_1 \geq \lambda_2 \geq \dots \geq \lambda_n$, entonces:

$$\lambda_n (y_1^2 + y_2^2 + \dots + y_n^2) \leq \langle y, Ay \rangle \leq \lambda_1 (y_1^2 + y_2^2 + \dots + y_n^2) \quad (2.19)$$

Notemos que:

$$\lambda_1 = \max_y \left(\left\{ \frac{\langle y, \Lambda y \rangle}{\langle y, y \rangle} \right\} \right) = \max_x \left(\left\{ \frac{\langle x, Ax \rangle}{\langle y, y \rangle} \right\} \right)$$

$$\lambda_n = \min_y \left(\left\{ \frac{\langle y, \Lambda y \rangle}{\langle y, y \rangle} \right\} \right) = \min_x \left(\left\{ \frac{\langle x, Ax \rangle}{\langle y, y \rangle} \right\} \right)$$

En efecto, sea:

$$u_1 = \max_y \left(\left\{ \frac{\langle y, \Lambda y \rangle}{\langle y, y \rangle} \right\} \right) \quad (2.20)$$

por (2.19) se tiene:

$$\langle y, \Lambda y \rangle \leq \lambda_1 (y_1^2 + y_2^2 + \dots + y_n^2) = \lambda_1 \langle y, y \rangle \quad \forall y \in \mathbb{R}^n$$

Luego:

$$\frac{\langle y, \Lambda y \rangle}{\langle y, y \rangle} \leq \lambda_1$$

entonces:

$$\max_y \left(\left\{ \frac{\langle y, \Lambda y \rangle}{\langle y, y \rangle} \right\} \right) \leq \lambda_1$$

y así:

$$u_1 \leq \lambda_1 \quad (2.21)$$

de (2.20) tenemos:

$$u_1 \geq \frac{\langle y, \Lambda y \rangle}{\langle y, y \rangle} \stackrel{(2.18)}{=} \frac{\lambda_1 y_1^2 + \lambda_2 y_2^2 + \dots + \lambda_n y_n^2}{\langle y, y \rangle} \quad \forall y \in \mathbb{R}^n \setminus \{0\}$$

en particular, escogemos: $y_1 \neq 0, y_2 = y_3 = \dots = y_n = 0$ entonces

$$u_1 \geq \frac{\lambda_1 y_1^2}{y_1^2} = \lambda_1 \implies \lambda_1 \leq u_1 \quad (2.22)$$

por lo tanto, de (2.21) y (2.22) se concluye que:

$$\lambda_1 = u_1 = \max_y \left(\left\{ \frac{\langle y, \Lambda y \rangle}{\langle y, y \rangle} \right\} \right)$$

Consideremos

$$v_n = \min_y \left(\left\{ \frac{\langle y, \Lambda y \rangle}{\langle y, y \rangle} \right\} \right) \quad (2.23)$$

de (2.19) tenemos:

$$\begin{aligned} \lambda_n \langle y, y \rangle &= \lambda_n (y_1^2 + y_2^2 + \dots + y_n^2) \leq \langle y, \Lambda y \rangle \quad \forall y \in \mathbb{R}^n \\ \Rightarrow \lambda_n &\leq \frac{\langle y, \Lambda y \rangle}{\langle y, y \rangle} \quad \forall y \in \mathbb{R}^n \setminus \{0\} \\ \Rightarrow \lambda_n &\leq \min_y \left(\left\{ \frac{\langle y, \Lambda y \rangle}{\langle y, y \rangle} \right\} \right) = v_n \implies \lambda_n \leq v_n \end{aligned} \quad (2.24)$$

de (2.23) se tiene:

$$v_n = \frac{\langle y, \Lambda y \rangle}{\langle y, y \rangle} \stackrel{(2.18)}{=} \frac{\lambda_1 y_1^2 + \lambda_2 y_2^2 + \dots + \lambda_n y_n^2}{\langle y, y \rangle} \quad \forall y \in \mathbb{R}^n \setminus \{0\}$$

en particular, hacemos $y_1 = y_2 = \dots = y_{n-1} = 0$ y $y_n \neq 0$, entonces:

$$v_n \leq \frac{\lambda_n y_n^2}{y_n^2} = \lambda_n \implies v_n \leq \lambda_n \quad (2.25)$$

de (2.24) y (2.25) se obtiene:

$$\lambda_n = v_n = \min_y \left(\left\{ \frac{\langle y, \Lambda y \rangle}{\langle y, y \rangle} \right\} \right)$$

por (2.17) se obtiene:

$$\lambda_1 = \max_y \left(\left\{ \frac{\langle y, \Lambda y \rangle}{\langle y, y \rangle} \right\} \right)$$

$$\lambda_n = \min_y \left(\left\{ \frac{\langle y, Ay \rangle}{\langle y, y \rangle} \right\} \right)$$

de donde:

$$\lambda_n \leq \frac{\langle x, Ax \rangle}{\langle x, x \rangle} \leq \lambda_1 \quad \forall x$$

definamos:

$$q(x) = \frac{\langle x, Ax \rangle}{\langle x, x \rangle} \quad (2.26)$$

que es llamado el cociente de Rayleigh

Teorema 2.5.2. Las raíces características $\lambda_1, \dots, \lambda_n$ se pueden expresar como sigue:

$$\begin{aligned} \lambda_1 &= \max_x \left(\left\{ \frac{\langle x, Ax \rangle}{\langle x, x \rangle} \right\} \right) \\ \lambda_2 &= \min_{\langle y, y \rangle=1} \max_{\langle x, y \rangle=0} \left(\left\{ \frac{\langle x, Ax \rangle}{\langle x, x \rangle} \right\} \right) \\ &\vdots \\ \lambda_k &= \min_{\substack{\langle y^i, y^i \rangle=1 \\ i=1, \dots, k-1}} \max_{\langle x, y_i \rangle=0} \left(\left\{ \frac{\langle x, Ax \rangle}{\langle x, x \rangle} \right\} \right) \\ &\vdots \end{aligned}$$

equivalentemente:

$$\begin{aligned} \lambda_n(A) = \lambda_n &= \min_x \left(\left\{ \frac{\langle x, Ax \rangle}{\langle x, x \rangle} \right\} \right) \\ \lambda_{n-1} &= \min_{\langle y, y \rangle=1} \max_{\langle x, y \rangle=0} \left(\left\{ \frac{\langle x, Ax \rangle}{\langle x, x \rangle} \right\} \right) \\ &\vdots \\ \lambda_k &= \min_{\substack{\langle y^i, y^i \rangle=1 \\ i=1, \dots, k-1}} \max_{\langle x, y_i \rangle=0} \left(\left\{ \frac{\langle x, Ax \rangle}{\langle x, x \rangle} \right\} \right) \\ &\vdots \end{aligned}$$

Prueba:

Sea $\lambda_1 = \max_x \left(\left\{ \frac{\langle x, Ax \rangle}{\langle x, x \rangle} \right\} \right)$ y $\lambda_n = \min_x \left(\left\{ \frac{\langle x, Ax \rangle}{\langle x, x \rangle} \right\} \right)$ fué probado antes.

Veamos que:

$$\lambda_2 = \min_{\langle y, y \rangle=1} \max_{\langle x, y \rangle=0} \left(\left\{ \frac{\langle x, Ax \rangle}{\langle x, x \rangle} \right\} \right)$$

para ello, sea: $x = Tz$:

$$\begin{aligned} u_2 &= \min_{\langle y, y \rangle = 1} \max_{\langle x, y \rangle = 0} \left(\left\{ \frac{\langle x, Ax \rangle}{\langle x, x \rangle} \right\} \right) \\ &= \min_{\langle y, y \rangle = 1} \max_{\langle Tz, y \rangle = 0} \left(\left\{ \frac{\langle Tz, ATz \rangle}{\langle Tz, Tz \rangle} \right\} \right) \\ &= \min_{\langle y, y \rangle = 1} \max_{\langle z, T^T y \rangle = 0} \left(\left\{ \frac{\langle z, T^T ATz \rangle}{\langle z, z \rangle} \right\} \right) \end{aligned}$$

Definamos $y' = T^T y$ y (esto es $y = Ty'$), entonces:

$$\begin{aligned} u_2 &= \min_{\langle Ty', Ty' \rangle = 1} \max_{\langle z, y' \rangle = 0} \left(\left\{ \frac{\langle z, \Lambda z \rangle}{\langle z, z \rangle} \right\} \right) \\ \Rightarrow u_2 &= \min_{\langle y', y' \rangle = 1} \max_{\langle z, y' \rangle = 0} \left(\left\{ \frac{\langle z, \Lambda z \rangle}{\langle z, z \rangle} \right\} \right) \end{aligned}$$

como observamos, es suficiente asumir que A es una matriz diagonal.

Sea

$$S = \{z \in \mathbb{R}(n, n)^n / z_3 = z_4 = \dots = z_n = 0 \wedge \langle z, y' \rangle = 0\} \quad y,$$

$$M = \{z \in \mathbb{R}(n, n)^n / \langle z, y \rangle = 0\} \implies S \subset M$$

luego:

$$\begin{aligned} \max_M \left(\left\{ \frac{\langle z, \Lambda z \rangle}{\langle z, z \rangle} \right\} \right) &\geq \max_S \left(\left\{ \frac{\langle z, \Lambda z \rangle}{\langle z, z \rangle} \right\} \right) \quad \forall y' \quad \text{con} \quad \langle z, y' \rangle = 0 \\ \Rightarrow u_2 &= \min_{\langle y, y \rangle = 0} \max_M \left(\left\{ \frac{\langle z, \Lambda z \rangle}{\langle z, z \rangle} \right\} \right) \geq \min_{\langle y, y \rangle = 0} \max_S \left(\left\{ \frac{\langle z, \Lambda z \rangle}{\langle z, z \rangle} \right\} \right) = \quad (2.27) \\ &= \min_{\langle y, y \rangle = 0} \max_S \left(\left\{ \frac{\lambda_1 z_1^2 + \lambda_2 z_2^2}{z_1^2 + z_2^2} \right\} \right) \end{aligned}$$

además como $\lambda_1 \geq \lambda_2 \Rightarrow \lambda_1 z_1^2 > \lambda_2 z_2^2$.

$$\Rightarrow \lambda_1 z_1^2 + \lambda_2 z_2^2 \geq \lambda_2 z_1^2 + \lambda_2 z_2^2 = \lambda_2 (z_1^2 + z_2^2) \frac{\lambda_1 z_1^2 + \lambda_2 z_2^2}{z_1^2 + z_2^2} \geq \lambda_2 \quad \forall z_1, z_2 \quad (2.28)$$

Luego de (2.27) y (2.28) se tiene:

$$u_2 \geq \lambda_2 \quad (2.29)$$

de la definición de u_2 se obtiene:

$$u_2 \leq \max_{\langle z, y \rangle = 0} \left(\left\{ \frac{\langle z, \Lambda z \rangle}{\langle z, z \rangle} \right\} \right) \quad \forall y \quad \text{con} \quad \langle y, y \rangle = 1$$

en particular, elegimos $y_1 = 1, y_2 = y_3 = \dots = y_n = 0$ entonces:

$$\begin{aligned} u_2 &\leq \max_{z_1=0} \left(\left\{ \frac{\lambda_1 z_1^2 + \lambda_2 z_2^2 + \dots + \lambda_n z_n^2}{z_1^2 + z_2^2 + \dots + z_n^2} \right\} \right) \\ &\leq \max_z \left(\left\{ \frac{\lambda_2 z_2^2 + \dots + \lambda_n z_n^2}{z_1^2 + z_2^2 + \dots + z_n^2} \right\} \right) \\ u_2 &\leq \lambda_2 \end{aligned}$$

esto es debido a que:

$$\begin{aligned} \lambda_n &\leq \lambda_2 \Rightarrow \lambda_n z_n^2 &\leq \lambda_2 z_n^2 \\ \lambda_{n-1} &\leq \lambda_2 \Rightarrow \lambda_{n-1} z_{n-1}^2 &\leq \lambda_2 z_{n-1}^2 \\ &\dots &\dots \\ \lambda_3 &\leq \lambda_2 \Rightarrow \lambda_3 z_3^2 &\leq \lambda_2 z_3^2 \\ \lambda_2 &\leq \lambda_2 \Rightarrow \lambda_2 z_2^2 &\leq \lambda_2 z_2^2 \\ \Rightarrow \frac{\lambda_n z_n^2 + \lambda_{n-1} z_{n-1}^2 + \dots + \lambda_2 z_2^2}{z_n^2 + z_{n-1}^2 + \dots + z_2^2} &\leq \lambda_2 \\ &\Rightarrow u_2 \leq \lambda_2 \end{aligned}$$

Luego $u_2 = \lambda_2$.

En general, podemos definir:

$$\begin{aligned} u_k &= \min_{\langle y^i, y^i \rangle = 1} \max_{\langle x, y^i \rangle = 0} \left(\left\{ \frac{\langle x, Ax \rangle}{\langle x, x \rangle} \right\} \right) \\ &= \min_{\langle y^i, y^i \rangle = 1} \max_{\langle z, y^i \rangle = 0} \left(\left\{ \frac{\langle z, \Lambda z \rangle}{\langle z, z \rangle} \right\} \right) \end{aligned}$$

consideremos

$$S_k = \{z / z_{k+1} = z_{k+2} = \dots = z_n = 0 \wedge \langle z, y^i \rangle = 0 \quad i = 1, 2, \dots, k-1\}$$

Notemos que $S_k \subset M$, luego:

$$\begin{aligned} \max_M \left(\left\{ \frac{\langle z, \Lambda z \rangle}{\langle z, z \rangle} \right\} \right) &\geq \max_{S_k} \left(\left\{ \frac{\langle z, \Lambda z \rangle}{\langle z, z \rangle} \right\} \right) \\ \Rightarrow u_k &= \min_{\langle y^i, y^i \rangle = 1} \max_M \left(\left\{ \frac{\langle z, \Lambda z \rangle}{\langle z, z \rangle} \right\} \right) \geq \min_{\langle y^i, y^i \rangle = 1} \max_{S_k} \left(\left\{ \frac{\langle z, \Lambda z \rangle}{\langle z, z \rangle} \right\} \right) \end{aligned}$$

$$= \min_{\substack{\langle y^i, y^i \rangle = 1 \\ i=1, \dots, k-1}} \max_S \left(\left\{ \frac{\lambda_1 z_1^2 + \lambda_2 z_2^2 + \dots + \lambda_k z_k^2}{z_1^2 + z_2^2 + \dots + z_k^2} \right\} \right) \geq \lambda_k \implies u_k \geq \lambda_k$$

De la definición de u_k se tiene:

$$u_k \leq \max_{\substack{\langle z, \Lambda z \rangle \\ \langle z, z \rangle \\ i=1, \dots, k-1}} \left(\left\{ \frac{\langle z, \Lambda z \rangle}{\langle z, z \rangle} \right\} \right) \quad \forall y^i \text{ con } \langle y^i, y^i \rangle = 1 \quad i = 1, \dots, k-1$$

en particular $y^i = e_i = \begin{pmatrix} 0 \\ \vdots \\ 1 \\ \vdots \\ 0 \end{pmatrix} \leftarrow \in \mathbb{R}^n$ 1 está ubicado en la i -ésima posición,

esto implica que $z_1 = z_2 = \dots = z_{k-1} = 0$ entonces:

$$u_k \leq \max_{\substack{z_i=0 \\ i=1, \dots, k-1}} \left(\left\{ \frac{\langle z, \Lambda z \rangle}{\langle z, z \rangle} \right\} \right) \leq \max_z \left(\left\{ \frac{\lambda_1 z_1^2 + \dots + \lambda_k z_k^2}{z_1^2 + \dots + z_k^2} \right\} \right) \leq \lambda_k$$

$$\implies u_k = \lambda_k$$

$$\lambda_n = \min \left(\left\{ \frac{\langle x, Ax \rangle}{\langle x, x \rangle} \right\} \right) \text{ ya fué probado.}$$

Veamos que $\lambda_{n-1} = \max_{\langle y, y \rangle = 1} \min_{\langle x, y \rangle = 0} \left(\left\{ \frac{\langle x, Ax \rangle}{\langle x, x \rangle} \right\} \right)$.

Sea $x = Tz$:

$$\begin{aligned} v_{n-1} &= \max_{\langle y, y \rangle = 1} \min_{\langle x, y \rangle = 0} \left(\left\{ \frac{\langle x, Ax \rangle}{\langle x, x \rangle} \right\} \right) \\ &= \max_{\langle y, y \rangle = 1} \min_{\langle Tz, y \rangle = 0} \left(\left\{ \frac{\langle Tz, ATz \rangle}{\langle Tz, Tz \rangle} \right\} \right) \\ &= \max_{\langle y, y \rangle = 1} \min_{\langle z, T^T y \rangle = 0} \left(\left\{ \frac{\langle z, T^T ATz \rangle}{\langle z, T^T Tz \rangle} \right\} \right) \end{aligned}$$

haciendo $y^n = T^T y \Rightarrow v_{n-1} = \max_{\langle T y^n, T y^n \rangle = 1} \min_{\langle z, y^n \rangle = 0} \left(\left\{ \frac{\langle z, \Lambda z \rangle}{\langle z, z \rangle} \right\} \right)$

$$\Rightarrow v_{n-1} = \max_{\langle y', y' \rangle = 1} \min_{\langle z, y' \rangle = 0} \left(\left\{ \frac{\langle z, \Lambda z \rangle}{\langle z, z \rangle} \right\} \right)$$

Sea $\widehat{S} = \{z/z_1 = z_2 = \dots = z_{n-2} = 0 \wedge \langle z, y^n \rangle = 0\}$, recordar que:

$$M = \{z/\langle z, y^n \rangle = 0\}$$

$$\text{Luego: } \min_M \left(\left\{ \frac{\langle z, \Lambda z \rangle}{\langle z, z \rangle} \right\} \right) \leq \min_{\widehat{S}} \left(\left\{ \frac{\langle z, \Lambda z \rangle}{\langle z, z \rangle} \right\} \right) \quad \forall y^n \quad \text{con } \langle z, y^n \rangle = 0$$

$$\Rightarrow v_{n-1} = \max_{(y^n, y^n)=1} \min_M \left(\left\{ \frac{\langle z, \Lambda z \rangle}{\langle z, z \rangle} \right\} \right) \leq \max_{(y^n, y^n)=1} \min_{\widehat{S}} \left(\left\{ \frac{\langle z, \Lambda z \rangle}{\langle z, z \rangle} \right\} \right)$$

$$\Rightarrow v_{n-1} \leq \max_{(y^n, y^n)=1} \min_{\widehat{S}} \left(\left\{ \frac{\lambda_{n-1} z_{n-1}^2 + \lambda_n z_n^2}{z_{n-1}^2 + z_n^2} \right\} \right) \quad (2.30)$$

además como:

$$\lambda_n \leq \lambda_{n-1}$$

$$\Rightarrow \lambda_n z_n^2 \leq \lambda_{n-1} z_n^2$$

$$\Rightarrow \lambda_n z_n^2 + \lambda_{n-1} z_{n-1}^2 \leq \lambda_{n-1} z_n^2 + \lambda_{n-1} z_{n-1}^2$$

$$\Rightarrow \frac{\lambda_n z_n^2 + \lambda_{n-1} z_{n-1}^2}{z_n^2 + z_{n-1}^2} \leq \lambda_{n-1} \quad \forall z_n, z_{n-1} \quad (2.31)$$

de (2.30) y (2.31) tenemos:

$$v_{n-1} \leq \lambda_{n-1} \quad (2.32)$$

de la definición de v_{n-1} tenemos:

$$v_{n-1} \geq \min_{\langle z, y^n \rangle=0} \left(\left\{ \frac{\langle z, \Lambda z \rangle}{\langle z, z \rangle} \right\} \right) \quad \forall y^n \quad \text{con } \langle z, y^n \rangle = 0$$

en particular escogemos:

$$y_1 = y_2 = \dots = y_{n-1} = 0, \quad y_n = 1$$

entonces:

$$\begin{aligned} v_{n-1} &\geq \min_{z_n=0} \left(\left\{ \frac{\lambda_1 z_1^2 + \lambda_2 z_2^2 + \dots + \lambda_n z_n^2}{z_1^2 + z_2^2 + \dots + z_n^2} \right\} \right) \\ &\geq \min_z \left(\left\{ \frac{\lambda_1 z_1^2 + \lambda_2 z_2^2 + \dots + \lambda_n z_n^2}{z_1^2 + z_2^2 + \dots + z_n^2} \right\} \right) \geq \lambda_{n-1} \\ &\Rightarrow v_{n-1} \geq \lambda_{n-1} \end{aligned} \quad (2.33)$$

de (2.32) y (2.33) tenemos:

$$v_{n-1} = \lambda_{n-1}$$

Prueba del teorema (2.5.1)

Supongamos que para algún r que $\lambda_r(A) > 0$ y definimos el subespacio $S_0 \subseteq \mathbb{R}^n$ por:

$$S_0 = \text{span}\{X^{-1}q_1, \dots, X^{-1}q_r\} \quad q_i \neq 0$$

$$y \in S_0 \Rightarrow y = \sum_{i=1}^r \beta_i X^{-1}q_i$$

donde $Aq_i = \lambda_i(A)q_i$, $i = 1, \dots, r$. De la caracterización de $\lambda_r(X^T AX)$ tenemos:

$$\begin{aligned} \lambda_r(X^T AX) &= \max_{\dim(S)=r} \min_{y \in S} \frac{y^T (X^T AX)y}{y^T y} \\ &\geq \min_{S_0} \frac{y^T (X^T AX)y}{y^T y} \end{aligned}$$

Ahora para cualquier $y \in \mathbb{R}^n$ tenemos:

$$\frac{\langle y, (X^T X)y \rangle}{\langle y, y \rangle} = \frac{y^T (X^T X)y}{y^T y} \geq \sigma_n^2(X), \quad \text{donde } \sigma_n^2(X) = \lambda_n(X^T X)$$

mientras para $y \in S_0$ es claro que:

$$\frac{y^T (X^T AX)y}{y^T y} \geq \lambda_r(A)$$

esto es:

$$\lambda_r(X^T AX) \geq \min_{y \in S_0} \left\{ \frac{y^T (X^T AX)y \ y^T (X^T X)y}{y^T (X^T X)y \ y^T y} \right\} \geq \lambda_r(A) \sigma_n^2(X)$$

con argumentos análogos con los roles de A y $X^T AX$ invertidos muestran que:

$$\lambda_r \geq \lambda_r(X^T AX) \sigma_n^2(X^{-1}) = \frac{\lambda_r(X^T AX)}{\sigma_1^2(X)}$$

esto es, $\lambda_r(A)$ y $\lambda_r(X^T AX)$ tienen el mismo signo y como mostramos que A y $X^T AX$ tienen el mismo número de valores propios positivos. Aplicamos este resultado a $-A$, y concluimos que A y $X^T AX$ tienen el mismo número de valores propios negativos. Es obvio que el mismo número de valores propios nulos para cada matriz es también el mismo. ■

2.6. Importancia de los Valores propios y Vectores Propios Dominantes en un Sistema Dinámico

Se llama λ_1 valor propio dominante de A si cumple:

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

donde $\lambda_1, \lambda_2, \dots, \lambda_n$ son valores propios de A .

Consideremos el sistema de tiempo - discreto homogéneo:

$$x_{k+1} = A x_k \quad k = 0, 1, 2, \dots, n$$

entonces

$$\begin{aligned} x_1 &= A x_0 \\ x_2 &= A^2 x_0 \\ &\vdots \\ x_k &= A^k x_0 \\ &\vdots \\ x_n &= A^n x_0 \end{aligned} \tag{2.34}$$

Supongamos que A tiene un conjunto de n vectores propios linealmente independientes.

Sea (λ_i, v_i) un par propio de A .

$$\begin{aligned} \Rightarrow A v_i &= \lambda_i v_i \\ A^k v_i &= \lambda_i^k v_i \end{aligned} \tag{2.35}$$

Para $x_0 \in \mathbb{R}^n$ se cumple:

$$\begin{aligned} x_0 &= \alpha_1 v_1 + \alpha_2 v_2 + \dots + \alpha_n v_n \\ A^k x_0 &= \alpha_1 A^k v_1 + \alpha_2 A^k v_2 + \dots + \alpha_n A^k v_n \\ A^k x_0 &\stackrel{(2.35)}{=} \alpha_1 \lambda_1^k v_1 + \alpha_2 \lambda_2^k v_2 + \dots + \alpha_n \lambda_n^k v_n \\ x_k &= \alpha_1 \lambda_1^k v_1 + \alpha_2 \lambda_2^k v_2 + \dots + \alpha_n \lambda_n^k v_n \end{aligned} \tag{2.36}$$

Observaciones 2. Si λ_1 es un valor propio dominante de A , entonces cumple:

1.

$$|\lambda_1|^k > |\lambda_i|^k \quad \forall i = 2, 3, \dots, n.$$

En efecto:

Como λ_1 es un valor propio dominante, se cumple:

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

$$\Rightarrow \forall k \in \mathbb{N} : |\lambda_1|^k > |\lambda_2|^k \geq |\lambda_3|^k \geq \dots \geq |\lambda_n|^k$$

$$\Rightarrow |\lambda_1|^k > |\lambda_i|^k \quad i = 2, 3, \dots, n$$

2. Para k grande

$$|\alpha_1 \lambda_1^k| \gg |\alpha_i \lambda_i^k| \quad i = 2, 3, 4, \dots, n; \alpha_1 \neq 0$$

En efecto:

$$\text{Si } \exists i_0/2 \leq i_0 \leq n : |\alpha_1| \leq |\alpha_{i_0}|$$

$$\Rightarrow \exists k_0 \in \mathbb{N} / |\alpha_1 \lambda_1^k| > |\alpha_i \lambda_i^k| \quad \forall k \geq k_0$$

En la segunda observación se concluye que para valores grandes de k , el vector x_k se aproximaría en la dirección del vector v_i correspondiente al valor propio dominante λ_1 .

En el caso que $\alpha_1 = 0$, el segundo valor propio dominante λ_2 y su correspondiente vector propio se denominaría *vector propio dominante*.

En resumen, el comportamiento de un sistema dinámico homogéneo está basado en analizar su primer y segundo valor propio dominante de una matriz y su correspondiente vector propio.

2.6.1. Cálculo de valores propios

Describiremos brevemente algunos métodos para calcular los valores propios dominantes y sus correspondientes vectores propios.

Método de la Potencia

Este método es usado para calcular el mayor valor propio en módulo y su correspondiente vector propio de una matriz A de orden $n \times n$.

Sean los valores propios $\lambda_1, \lambda_2, \dots, \lambda_n$ de una matriz A . Y sea λ_1 valor propio dominante con vector propio v_1 correspondiente. Asumamos que A es diagonalizable. Presentamos el algoritmo del método de la Potencia.

Definamos $\text{máx}(\{v\}) = \text{máx}(\{v_i\})$, donde v_i $i = 1, \dots, n$ son las componentes del vector v .

ALGORITMO 2.6.1. (MÉTODO DE LA POTENCIA):

Paso 1 Elijamos $x_0 \in \mathbb{R}^n \setminus \{0\}$

Paso 2 Para $k = 1, 2, \dots$ hasta que converja

$$\begin{aligned} \hat{x}_k &= Ax_{k-1} \\ x_k &= \frac{\hat{x}_k}{\text{máx}(\{\hat{x}_k\})} \\ \text{END} \end{aligned}$$

Teorema 2.6.1. Sea la sucesión $\{\text{máx}(\{\hat{x}_k\})\}$ que converge a λ_1

y $x_k \rightarrow w^1$ múltiplo de v_1 cuando $k \rightarrow \infty$

Prueba:

Aplicando sucesivamente el algoritmo tenemos:

$$\begin{aligned} k = 1: \quad \hat{x}_1 &= Ax_0 & x_1 &= \frac{Ax_0}{\text{máx}(\{Ax_0\})} \\ k = 2: \quad \hat{x}_2 &= Ax_1 = \frac{A^2x_0}{\text{máx}(\{Ax_0\})} & x_2 &= \frac{A^2x_0}{\text{máx}(\{Ax_0\}) \text{máx}(\{\frac{A^2x_0}{\text{máx}(\{Ax_0\})}\})} \\ & & &= \frac{A^2x_0}{\text{máx}(\{A^2x_0\})} \\ k = 3: \quad \hat{x}_3 &= Ax_2 = \frac{A^3x_0}{\text{máx}(\{A^2x_0\})} & x_3 &= \frac{A^3x_0}{\text{máx}(\{A^2x_0\}) \text{máx}(\{\frac{A^3x_0}{\text{máx}(\{A^2x_0\})}\})} \\ & & &= \frac{A^3x_0}{\text{máx}(\{A^3x_0\})} \\ \vdots & & & \vdots \\ \hat{x}_k &= \frac{A^kx_0}{\text{máx}(\{A^{k-1}x_0\})} & x_k &= \frac{A^kx_0}{\text{máx}(\{A^kx_0\})} \end{aligned}$$

Como A es diagonalizable, entonces los vectores propios $v_i, i = 1, \dots, n$ asociados con $\lambda_i, i = 1, \dots, n$ pueden ser elegidos linealmente independientes.

Luego x^0 puede ser escrito de la forma:

$$\begin{aligned} x_0 &= \alpha_1 v_1 + \alpha_2 v_2 + \dots + \alpha_n v_n, \quad \alpha_1 \neq 0 \\ \implies A^k x_0 &= A^k (\alpha_1 v_1 + \alpha_2 v_2 + \dots + \alpha_n v_n) \\ &\stackrel{Av_i = \lambda_i v_i}{=} \alpha_1 \lambda_1^k v_1 + \alpha_2 \lambda_2^k v_2 + \dots + \alpha_n \lambda_n^k v_n \\ \implies A^k x_0 &= \lambda_1^k \left(\alpha_1 v_1 + \alpha_2 \left(\frac{\lambda_2}{\lambda_1} \right)^k v_2 + \dots + \alpha_n \left(\frac{\lambda_n}{\lambda_1} \right)^k v_n \right) \end{aligned}$$

Como $|\lambda_1| > |\lambda_2| \geq \dots \geq |\lambda_n|$. Entonces $\lim_{k \rightarrow \infty} \left(\frac{\lambda_i}{\lambda_1} \right)^k = 0, i = 2, \dots, n$.

Luego:

$$\begin{aligned} x_k &= \frac{A^k x_0}{\max(\{A^k x_0\})} = \frac{\lambda_1^k (\alpha_1 v_1 + \alpha_2 \left(\frac{\lambda_2}{\lambda_1} \right)^k v_2 + \dots + \alpha_n \left(\frac{\lambda_n}{\lambda_1} \right)^k v_n)}{\max(\{\lambda_1^k (\alpha_1 v_1 + \alpha_2 \left(\frac{\lambda_2}{\lambda_1} \right)^k v_2 + \dots + \alpha_n \left(\frac{\lambda_n}{\lambda_1} \right)^k v_n\})} \\ &= \frac{\alpha_1 v_1 + \alpha_2 \left(\frac{\lambda_2}{\lambda_1} \right)^k v_2 + \dots + \alpha_n \left(\frac{\lambda_n}{\lambda_1} \right)^k v_n}{\max(\{\alpha_1 v_1 + \alpha_2 \left(\frac{\lambda_2}{\lambda_1} \right)^k v_2 + \dots + \alpha_n \left(\frac{\lambda_n}{\lambda_1} \right)^k v_n\})} \rightarrow c v_1 \end{aligned}$$

donde

$$c = \frac{\alpha_1}{\max(\{\alpha_1 v_1 + \alpha_2 \left(\frac{\lambda_2}{\lambda_1} \right)^k v_2 + \dots + \alpha_n \left(\frac{\lambda_n}{\lambda_1} \right)^k v_n\})}$$

es una constante y

$$\begin{aligned} \max(\{\hat{x}_k\}) &= \max \left(\left\{ \frac{A^k x_0}{\max(\{A^{k-1} x_0\})} \right\} \right) = \frac{\max(\{A^k x_0\})}{\max(\{A^{k-1} x_0\})} = \\ &= \frac{\lambda_1^k \max(\{\alpha_1 v_1 + \alpha_2 \left(\frac{\lambda_2}{\lambda_1} \right)^k v_2 + \dots + \alpha_n \left(\frac{\lambda_n}{\lambda_1} \right)^k v_n\})}{\lambda_1^{k-1} \max(\{\alpha_1 v_1 + \alpha_2 \left(\frac{\lambda_2}{\lambda_1} \right)^{k-1} v_2 + \dots + \alpha_n \left(\frac{\lambda_n}{\lambda_1} \right)^{k-1} v_n\})} \\ &= \frac{\lambda_1 \max(\{\alpha_1 v_1 + \alpha_2 \left(\frac{\lambda_2}{\lambda_1} \right)^k v_2 + \dots + \alpha_n \left(\frac{\lambda_n}{\lambda_1} \right)^k v_n\})}{\max(\{\alpha_1 v_1 + \alpha_2 \left(\frac{\lambda_2}{\lambda_1} \right)^{k-1} v_2 + \dots + \alpha_n \left(\frac{\lambda_n}{\lambda_1} \right)^{k-1} v_n\})} \rightarrow \lambda_1. \end{aligned}$$

Observación El Método de la Potencia es utilizado si se cumplen las siguientes hipótesis:

1. $\alpha_1 \neq 0$

2. λ_1 es el único valor propio dominante.

Ejemplo.

$$\text{Sea } A = \begin{pmatrix} 1 & 2 & 3 \\ 2 & 3 & 4 \\ 3 & 4 & 5 \end{pmatrix}$$

y consideremos $x_0 = (1, 1, 1)^T \in \mathbb{R}^3 \setminus \{0\}$

k=1:

$$\hat{x}_1 = Ax_0 = \begin{pmatrix} 1 & 2 & 3 \\ 2 & 3 & 4 \\ 3 & 4 & 5 \end{pmatrix} \begin{pmatrix} 1 \\ 1 \\ 1 \end{pmatrix} = \begin{pmatrix} 6 \\ 9 \\ 12 \end{pmatrix};$$

$$\text{máx}(\{\hat{x}_1\}) = 12$$

$$x_1 = \frac{\hat{x}_1}{\text{máx}(\{\hat{x}_1\})} = (1/2, 3/4, 1)^T = (0.50, 0.75, 1.00)^T$$

k=2:

$$\hat{x}_2 = Ax_1 = \begin{pmatrix} 1 & 2 & 3 \\ 2 & 3 & 4 \\ 3 & 4 & 5 \end{pmatrix} \begin{pmatrix} 1/2 \\ 3/4 \\ 1 \end{pmatrix} = \begin{pmatrix} 5 \\ 29/4 \\ 19/2 \end{pmatrix} = \begin{pmatrix} 5.00 \\ 7.25 \\ 9.50 \end{pmatrix};$$

$$\text{máx}(\{\hat{x}_2\}) = 19/2 = 9.50$$

$$x_2 = \frac{\hat{x}_2}{\text{máx}(\{\hat{x}_2\})} = (10/19, 29/38, 1)^T \approx (0.5263, 0.7632, 1.0000)^T$$

k=3:

$$\hat{x}_3 = Ax_2 = \begin{pmatrix} 1 & 2 & 3 \\ 2 & 3 & 4 \\ 3 & 4 & 5 \end{pmatrix} \begin{pmatrix} 10/19 \\ 29/38 \\ 1 \end{pmatrix} = \begin{pmatrix} 96/19 \\ 279/38 \\ 183/19 \end{pmatrix} \approx \begin{pmatrix} 5.0526 \\ 7.4321 \\ 9.6316 \end{pmatrix}$$

$$\text{máx}(\{\hat{x}_3\}) = 183/19 \approx 9.6316$$

$$x_3 = \frac{\hat{x}_3}{\text{máx}(\{\hat{x}_3\})} = (32/61, 93/122, 1)^T \approx (0.5246, 0.7623, 1.0000)^T$$

k=4:

$$\hat{x}_4 = Ax_3 = \begin{pmatrix} 1 & 2 & 3 \\ 2 & 3 & 4 \\ 3 & 4 & 5 \end{pmatrix} \begin{pmatrix} 32/61 \\ 93/122 \\ 1 \end{pmatrix} = \begin{pmatrix} 308/61 \\ 895/122 \\ 587/61 \end{pmatrix} \approx$$
$$\approx \begin{pmatrix} 5.049180 \\ 7.336066 \\ 9.622951 \end{pmatrix}$$

$$\text{máx}(\{\hat{x}_4\}) = 587/61 \approx 9.622951$$

$$x_4 = \frac{\hat{x}_4}{\text{máx}(\{\hat{x}_4\})} = (308/587, 895/1174, 1)^T \approx$$
$$(0.52470187, 0.76235094, 1.00000000)^T$$

k=5:

$$\hat{x}_5 = Ax_4 = \begin{pmatrix} 1 & 2 & 3 \\ 2 & 3 & 4 \\ 3 & 4 & 5 \end{pmatrix} \begin{pmatrix} 32/61 \\ 93/122 \\ 1 \end{pmatrix} = \begin{pmatrix} 308/587 \\ 895/1174 \\ 1 \end{pmatrix} \approx$$
$$\approx \begin{pmatrix} 5.0494037479 \\ 7.3364565588 \\ 9.6235093697 \end{pmatrix}$$

$$\text{máx}(\{\hat{x}_5\}) = 5649/587 \approx 9.6235093697$$

$$x_5 = \frac{\hat{x}_5}{\text{máx}(\{\hat{x}_5\})} = \begin{pmatrix} 988/1883 \\ 2871/3766 \\ 1 \end{pmatrix} \approx \begin{pmatrix} 0.524694636219 \\ 0.762347318109 \\ 1.000000000000 \end{pmatrix}$$

$$\text{Normalizandolo tenemos: } \frac{x_5}{\|x_5\|} = (0.3850895499, 0.5595101709, 0.7339307919)^T.$$

□

Convergencia del Método de la Potencia

La razón de convergencia del método de potencia está determinado por $\frac{\lambda_2}{\lambda_1}$ como veremos en seguida, de la prueba del teorema (2.6.1) (método de la potencia)

tenemos:

$$x_k = \frac{\alpha_1}{\beta} v_1 + \frac{\alpha_2}{\beta} \left(\frac{\lambda_2}{\lambda_1}\right)^k v_2 + \dots + \frac{\alpha_n}{\beta} \left(\frac{\lambda_n}{\lambda_1}\right)^k v_n \quad \text{donde } \beta = \frac{1}{\lambda_1^k}$$

$$x_k = \alpha'_1 v_1 + \alpha'_2 \left(\frac{\lambda_2}{\lambda_1}\right)^k v_2 + \dots + \alpha'_n \left(\frac{\lambda_n}{\lambda_1}\right)^k v_n$$

donde $\alpha'_i = \frac{\alpha_i}{\beta} \quad \forall i = 1, \dots, n$; entonces:

$$\begin{aligned} \|x_k - \alpha'_1 v_1\| &= \left\| \alpha'_2 \left(\frac{\lambda_2}{\lambda_1}\right)^k v_2 + \dots + \alpha'_n \left(\frac{\lambda_n}{\lambda_1}\right)^k v_n \right\| \\ &\leq |\alpha'_2| \left|\frac{\lambda_2}{\lambda_1}\right|^k \|v_2\| + \dots + |\alpha'_n| \left|\frac{\lambda_n}{\lambda_1}\right|^k \|v_n\| \\ &\leq \left|\frac{\lambda_2}{\lambda_1}\right|^k (|\alpha'_2| \|v_2\| + \dots + |\alpha'_n| \|v_n\|) \end{aligned}$$

(debido a que $|\lambda_1| > |\lambda_2| \geq \dots \geq |\lambda_n|$)

$$\Rightarrow \left|\frac{\lambda_2}{\lambda_1}\right|^k \geq \left|\frac{\lambda_3}{\lambda_1}\right|^k \geq \dots \geq \left|\frac{\lambda_n}{\lambda_1}\right|^k \quad \forall k = 1, 2, 3, \dots$$

$$\|x_k - \alpha'_1 v_1\| \leq \left|\frac{\lambda_2}{\lambda_1}\right|^k (|\alpha'_2| \|v_2\| + \dots + |\alpha'_n| \|v_n\|)$$

Luego tenemos:

$$\|x_k - \alpha'_1 v_1\| \leq \alpha' \left|\frac{\lambda_2}{\lambda_1}\right|^k$$

donde $\alpha' = |\alpha'_2| \|v_2\| + \dots + |\alpha'_n| \|v_n\|$.

Notar que esta prueba es debido a que x_k se aproxima a $\alpha'_1 v_1$ dependiendo de la rapidez como $\left|\frac{\lambda_2}{\lambda_1}\right|^k$ va hacia cero. Si λ_2 está cercano a λ_1 , entonces la convergencia se hace lenta, y si $\lambda_2 \ll \lambda_1$ la convergencia se hace rápida. ■

Ejemplo. Si consideramos $\lambda_1 = 10$ y $\lambda_2 = 9$, y queremos obtener una precisión de seis dígitos exactos, para ello se tiene que cumplir:

$$\left|\frac{9}{10}\right|^k < 10^{-6}$$

de aquí:

$$k > -\frac{\ln(10)}{6 \ln\left(\frac{9}{10}\right)} = 3.6424 \implies k_{min} = 4$$

Ahora, si consideramos $\lambda_1 = 10$ y $\lambda_2 = 1$, y queremos obtener la misma precisión, entonces:

$$\left| \frac{1}{10} \right|^k < 10^{-6}$$

de aquí:

$$k > -\frac{\ln(10)}{6 \ln(\frac{1}{10})} = 0.1667 \implies k_{min} = 1$$

se observa que en el primer caso, se necesitaría un mayor número de iteraciones que en el segundo caso.

2.6.2. Método de la Potencia con Desplazamiento

En algunos casos, la convergencia puede ser mejorada usando un cambio adecuado.

Sea $v \in \mathbb{R}^n$ un vector propio de A asociado al valor propio λ , entonces: $Av = \lambda v$.

Sea σ un cambio adecuado:

$$\Rightarrow (A - \sigma I)v = Av - \sigma v$$

$$(A - \sigma I)v = \lambda v - \sigma v$$

$$(A - \sigma I)v = (\lambda - \sigma)v,$$

por tanto, v es un vector propio de la matriz $A - \sigma I$ asociado al valor propio $(\lambda - \sigma)$.

Entonces, podemos decir que $(\lambda_1 - \sigma)$ es el valor propio dominante de $(A - \sigma I)$, y si el método de la Potencia es aplicado a la matriz de cambio $(A - \sigma I)$ entonces la razón de convergencia es determinado por:

$$\frac{|\lambda_2 - \sigma|}{|\lambda_1 - \sigma|} \text{ en vez de } \frac{\lambda_2}{\lambda_1}$$

Asumiendo que los λ_i son reales, con

$$\sigma = \frac{1}{2}(\lambda_2 + \lambda_n)$$

puede darnos una convergencia rápida, pero no siempre se cumple.

2.6.3. Método de la Potencia Inversa

El siguiente método iterativo es conocido como la **iteración inversa**, es un método efectivo para calcular un vector propio cuando se tiene en forma razonable una buena aproximación a un valor propio conocido.

ALGORITMO 2.6.2. (MÉTODO DE LA POTENCIA INVERSA):

Sea σ una aproximación al valor propio real λ_1 tal que: $|\lambda_1 - \sigma| \leq |\lambda_i - \sigma|$ ($i \neq 1$)

Paso 1 Elijamos x_0 .

Paso 2 Para $k = 1, 2, \dots$ hacemos

Resolvemos $(A - \sigma I)\hat{x}_k = x_{k-1}$, \hat{x}_k es la incógnita a encontrar usando el método de la eliminación gaussiana con pivot parcial.

$$x_k = \frac{\hat{x}_k}{\max(\{\hat{x}_k\})}$$

Parar si $\|(A - \sigma I)x_k\|_\infty < c\mu\|A\|_\infty$ donde c es una constante.

Teorema 2.6.2. *La sucesión $\{x_k\}$ converge a la dirección del vector correspondiente al valor propio λ_1*

Prueba:

Los valores propios de $(A - \sigma I)^{-1}$ son

$$(\lambda_1 - \sigma)^{-1}, (\lambda_2 - \sigma)^{-1}, \dots, (\lambda_n - \sigma)^{-1}$$

y los vectores propios v_1, v_2, \dots, v_n son los mismos como los de A . Luego, como en el caso del método de la potencia, podemos escribir:

$$\begin{aligned} \hat{x}_k &= \frac{c_1}{(\lambda_1 - \sigma)^k} v_1 + \frac{c_2}{(\lambda_2 - \sigma)^k} v_2 + \dots + \frac{c_n}{(\lambda_n - \sigma)^k} v_n \\ &= \frac{1}{(\lambda_1 - \sigma)^k} \left[c_1 v_1 + c_2 \left(\frac{\lambda_1 - \sigma}{\lambda_2 - \sigma} \right)^k v_2 + \dots + c_n \left(\frac{\lambda_1 - \sigma}{\lambda_n - \sigma} \right)^k v_n \right] \end{aligned}$$

como λ_1 es bien cercano a σ que cualquier otro valor propio, entonces el primer término del lado derecho $\frac{1}{(\lambda_1 - \sigma)^k}$ es el dominante.

Entonces $x_k \rightarrow \alpha v_1$ (múltiplo) cuando $k \rightarrow \infty$.

Ejemplo.

$$\text{Sea } A = \begin{pmatrix} 1 & 2 & 3 \\ 2 & 3 & 4 \\ 3 & 4 & 5 \end{pmatrix}$$

y consideremos $x_0 = (1, 1, 1)^T \in \mathbb{R}^3 \setminus \{0\}$ y $\sigma = 9$.

k=1:

$$\hat{x}_1 = (1, 1.5, 2)^T$$
$$x_1 = \frac{\hat{x}_1}{\|\hat{x}_1\|_2} = (0.3714, 0.5571, 0.7428)^T$$

k=2:

$$\hat{x}_2 = (0.6190, 0.8975, 1.1761)^T$$
$$x_2 = \frac{\hat{x}_2}{\|\hat{x}_2\|_2} = (0.3860, 0.5597, 0.7334)^T$$

k=3:

$$\hat{x}_3 = (0.6176, 0.8974, 1.1772)^T$$
$$x_3 = \frac{\hat{x}_3}{\|\hat{x}_3\|_2} = (0.3851, 0.5595, 0.7340)^T$$

k=4:

$$\hat{x}_4 = (0.6176, 0.8974, 1.1772)^T$$
$$x_4 = \frac{\hat{x}_4}{\|\hat{x}_4\|_2} = (0.3851, 0.5595, 0.7340)^T$$

k=5:

$$\hat{x}_5 = (0.6177, 0.8974, 1.1772)^T$$
$$x_5 = \frac{\hat{x}_5}{\|\hat{x}_5\|_2} = (0.3851, 0.5595, 0.7339)^T$$

¿Cómo escoger el vector inicial x_0 ?

Para escoger el vector inicial x_0 realizamos algunas iteraciones del método de la potencia, y luego cambiamos al método de iteración inversa, es decir, con el último vector generado por el método de la potencia como vector inicial para el método de

iteración inversa. Otra alternativa que fue propuesta por Wilkinson para hallar el x_0 es asumir que:

$$Le = Px_0$$

donde P es una matriz de permutación que satisface:

$$P(A - \sigma I) = LU$$

y $e = (1, 1, \dots, 1)^T \in \mathbb{R}^n$, entonces dos iteraciones son recomendadas, asumiendo que σ es una buena aproximación para el valor.

Ejemplo. Sea

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

luego, los valores propios de A son:

$$0, -0.6235, \text{ y } 9.6235$$

Elegimos: $\sigma = 9.1$:

$$A - \sigma I = \begin{pmatrix} -8.1 & 2.0 & 3.0 \\ 2.0 & -6.1 & 4.0 \\ 3.0 & 4.0 & -4.1 \end{pmatrix}$$

y su factorización LU es el siguiente:

$$L = \begin{pmatrix} 1 & 0 & 0 \\ -0.2469 & 1 & 0 \\ -0.3704 & -0.8456 & 1 \end{pmatrix}, \quad U = \begin{pmatrix} -8.1 & 2 & 3 \\ 0 & -5.6062 & 4.7407 \\ 0 & 0 & 0.0200 \end{pmatrix}$$

$$P = \begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix}$$

$$x_0 = (P)^{-1}Le = \begin{pmatrix} 1.000, & 0.7531, & -0.2160 \end{pmatrix}^T$$

Para dos iteraciones tenemos:

$$\begin{aligned}
 k = 1: \hat{x}_1 &= (A - \sigma I)^{-1}x_0 = \begin{pmatrix} 0.4003, & 0.6507, & 0.98040 \end{pmatrix}^T \\
 x_1 &= \frac{\hat{x}_1}{\max(\hat{x}_1)} = \begin{pmatrix} 0.4083, & 0.6637, & 1.0000 \end{pmatrix}^T \\
 k = 2: \hat{x}_2 &= (A - \sigma I)^{-1}x_1 = \begin{pmatrix} 0.9367, & 1.3540, & 1.7624 \end{pmatrix}^T \\
 x_2 &= \begin{pmatrix} 0.5315, & 0.7863, & 1.0000 \end{pmatrix}^T
 \end{aligned}$$

normalizando este vector propio cuatro dígitos se obtiene:

$$\begin{pmatrix} 0.3851, & 0.5595, & 0.7339 \end{pmatrix}^T$$

2.6.4. El Cociente de Rayleigh

Teorema 2.6.3. *Sea A una matriz simétrica y sea $x \in \mathbb{R}^n$ una buena aproximación para un vector propio. Entonces el cociente:*

$$R_q = \sigma = \frac{x^T Ax}{x^T x} \quad (2.37)$$

se aproxima al valor propio λ por lo cual x es el correspondiente vector propio.

Prueba:

Como A es simétrica, por el teorema (2.0.8) existe un conjunto de vectores propios ortonormales v_1, v_2, \dots, v_n , por lo tanto, x se puede expresar como:

$$x = c_1 v_1 + c_2 v_2 + \dots + c_n v_n \quad (2.38)$$

como $v_i \quad i = 1, 2, \dots, n$ es un conjunto de vectores propios ortonormales, es decir:

$$\begin{cases} v_i^T v_j = 1 & \text{si } i = j \\ v_i^T v_j = 0 & \text{si } i \neq j \end{cases}$$

$$\begin{aligned}
 \sigma &= \frac{x^T Ax}{x^T x} = \frac{(c_1 v_1 + c_2 v_2 + \dots + c_n v_n)^T A (c_1 v_1 + c_2 v_2 + \dots + c_n v_n)}{(c_1 v_1 + c_2 v_2 + \dots + c_n v_n)^T (c_1 v_1 + c_2 v_2 + \dots + c_n v_n)} \\
 &= \frac{(c_1 v_1 + c_2 v_2 + \dots + c_n v_n)^T (c_1 \lambda_1 v_1 + c_2 \lambda_2 v_2 + \dots + c_n \lambda_n v_n)}{c_1^2 + c_2^2 + \dots + c_n^2} \\
 &= \frac{\lambda_1 c_1^2 + \lambda_2 c_2^2 + \dots + \lambda_n c_n^2}{c_1^2 + c_2^2 + \dots + c_n^2}
 \end{aligned}$$

$$= \lambda_1 \left[\frac{1 + \left(\frac{\lambda_2}{\lambda_1}\right)\left(\frac{c_2}{c_1}\right)^2 + \dots + \left(\frac{\lambda_n}{\lambda_1}\right)\left(\frac{c_n}{c_1}\right)^2}{1 + \left(\frac{c_2}{c_1}\right)^2 + \dots + \left(\frac{c_n}{c_1}\right)^2} \right] \quad (2.39)$$

como x_1 es una buena aproximación a v_1 , en (2.38) tenemos que c_1 es más grande que los otros c_i $i = 1, 2, \dots, n$.

Entonces de (2.39), la expresión entre corchetes se aproxima a 1, entonces σ se aproxima a λ . ■

2.6.5. Iteración del Cociente de Rayleigh

La idea de aproximar un valor propio puede ser combinado con la iteración inversa para calcular sucesivamente aproximaciones de un valor propio y el correspondiente vector propio en un modelo iterativo conocido como *Iteración del Cociente de Rayleigh*.

En el siguiente algoritmo N es el número mínimo de iteraciones y x_0 es una aproximación inicial para un vector propio.

ALGORITMO 2.6.3. (Iteración del Cociente de Rayleigh):

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

Paso 1 Calcular

$$\sigma_k = \frac{x_k^T A x_k}{x_k^T x_k}$$

Paso 2 Resolver para \hat{x}_{k+1}

$$(A - \sigma_k I) \hat{x}_{k+1} = x_k$$

Paso 3 Calcular:

$$x_{k+1} = \frac{\hat{x}_k}{\max(\hat{x}_{k+1})}$$

Paso 4 Finalizar si el par (σ_k, x_k) es un par valor propio - vector propio aceptable o si $k > N$.

Observación El mejor camino para escoger x_0 puede ser usando el Método de la Potencia con pocas iteraciones y luego usar la última aproximación como x_0 .

Ejemplo. Siendo

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

luego, los valores propios de A son:

$$0, \quad -0.6235, \quad \text{y} \quad 9.6235$$

entonces, después de realizar tres iteraciones en el Método de la Potencia obtenemos como punto inicial para la Iteración del Cociente de Rayleigh:

$$x_0 = \left(0.5246, \quad 0.7622, \quad 1 \right)^T$$

$$\begin{aligned} k = 0 : \quad \sigma_0 &= \frac{x_0^T A x_0}{x_0^T x_0} = 9.6235 & x_1 &= \left(0.5247, \quad 0.7623, \quad 1 \right)^T \\ k = 1 : \quad \sigma_1 &= \frac{x_1^T A x_1}{x_1^T x_1} = 9.6235 & x_2 &= \left(1, \quad 1.4529, \quad 1.9059 \right)^T \end{aligned}$$

el vector propio normalizado asociado a 9.6255 es: $\left(0.3851, \quad 0.5595, \quad 0.7339 \right)^T$

2.6.6. Cálculo de los Valores Propios Subdominantes y Vectores Propios: Deflación

Una vez que el valor propio dominante λ_1 y su correspondiente vector propio v_1 ha sido calculado, el próximo valor propio dominante λ_2 puede ser calculado usando el Método de Deflación.

Deflación de Housholder

Por razones de estabilidad numérica, usaremos matrices de Housholder para este método.

Teorema 2.6.4. Sea λ_1 un valor propio y su correspondiente vector propio v_1 , H una

matriz de Housholder tal que Hv_1 es un múltiplo de e_1 , entonces:

$$A_1 = HAH = \begin{pmatrix} \lambda_1 & * & * & * & \dots & * \\ 0 & & & & & \\ \vdots & & A_2 & & & \\ 0 & & & & & \end{pmatrix} = \left(\begin{array}{c|c} \lambda_1 & b^T \\ \hline 0 & A_2 \end{array} \right)$$

donde A_2 es una matriz de $(n - 1) \times (n - 1)$ y los valores propios de A_2 son los mismos de A excepto λ_1 , en particular si:

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

entonces el valor propio dominante de A_2 es λ_2 , el cual es el segundo valor propio dominante (o también llamado subdominante) de A .

Prueba:

De $Av_1 = \lambda_1 v_1$ tenemos:

$$\begin{aligned} HAHv_1 &= \lambda_1 Hv_1 \quad \text{desde que} \quad H^2 = I \\ HAH(ke_1) &= \lambda_1 ke_1 \quad \text{debido a que} \quad Hv_1 = ke_1 \\ HAH e_1 &= \lambda_1 e_1 \end{aligned}$$

entonces, la primera componente de la primera columna de HAH es λ_1 y los demás son ceros. Luego:

$$\begin{pmatrix} \lambda_1 & * & * & * & \dots & * \\ 0 & & & & & \\ \vdots & & A_2 & & & \\ 0 & & & & & \end{pmatrix}$$

$$\det(HAH - \lambda I) = (\lambda_1 - \lambda) \det(A_2 - \lambda I)$$

se sigue que los valores propios de A_2 son los mismos de A_1 excepto λ_1 . Más aún, si:

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

el valor propio dominante de A_2 es λ_2 , el cual es el segundo valor propio dominante de A .

ALGORITMO 2.6.4. (Deflación de Housholder para calcular el Valor Propio Subdominante):

Paso 1 Calcular el valor propio dominante λ_1 y su correspondiente vector propio v_1 usando los Métodos de la Potencia y la Iteración Inversa.

Paso 2. Encontrar una matriz de Housholder tal que:

$$Hv_1 = \begin{pmatrix} * \\ 0 \\ \vdots \\ 0 \end{pmatrix}$$

Paso 3 Calcular HAH .

Paso 4 Descartar la primera fila y la primera columna de HAH y encontrar el valor propio dominante de la matriz $(n - 1) \times (n - 1)$ obtenida.

Ejemplo. Sea

$$A = \begin{pmatrix} 0.2190 & 0.6793 & 0.5194 \\ 0.0470 & 0.9347 & 0.8310 \\ 0.6789 & 0.3835 & 0.0346 \end{pmatrix}$$

Los valores propios de A son 0.0018, -0.3083 y 1.4947.

1. $\lambda_1 = 1.4947$ y $v_1 = \begin{pmatrix} -0.5552 \\ -0.7039 \\ -0.4430 \end{pmatrix}$

2.

$$u = v_1 - \|v_1\|_2 e_1 = \begin{pmatrix} -1.5552 \\ -0.7039 \\ -0.4430 \end{pmatrix}$$

$$H = I - \frac{2uu^T}{u^T u} = \begin{pmatrix} -0.5552 & -0.7039 & -0.4430 \\ -0.7039 & 0.6814 & -0.2005 \\ -0.4430 & -0.2005 & -0.8738 \end{pmatrix}$$

$$Hv_1 = \begin{pmatrix} 1 \\ 0 \\ 0 \end{pmatrix}$$

3.

$$HAH = \begin{pmatrix} 1.4977 & -0.3223 & -0.3331 \\ 0 & 0.1987 & 0.2672 \\ 0 & 0.3736 & -0.5052 \end{pmatrix} = \left(\begin{array}{c|cc} 1.4977 & -0.3223 & -0.3331 \\ \hline 0 & & A_2 \\ 0 & & \end{array} \right)$$

4. El valor propio dominante de A_2 es -0.3083 , el cual es el valor propio subdominante de A .

Observación Una vez calculado el valor propio subdominante λ_2 , el vector propio v_2 puede ser calculado usando el Método de la iteración inversa.

Cálculo de otros Valores Propios y Vectores Propios Grandes

Una vez calculado (λ_2, v_2) , la matriz A_2 puede ser deflactada, usando este par para calcular el par (λ_3, v_3) , y del par (λ_3, v_3) se calcula el par (λ_4, v_4) y así sucesivamente.

Cálculo de los Valores Propios Pequeños

Sea A una matriz no singular y sean los valores propios ordenados de tal forma:

$$|\lambda_1| > |\lambda_2| \geq |\lambda_3| \cdots \geq |\lambda_{n-1}| > |\lambda_n| > 0$$

donde λ_1 es el valor propio dominante de A y λ_n el valor propio más pequeño.

Entonces, los valores propios de A^{-1} son los valores propios inversos de A y son ordenados como:

$$\left| \frac{1}{\lambda_n} \right| > \left| \frac{1}{\lambda_{n-1}} \right| \geq \left| \frac{1}{\lambda_{n-2}} \right| \geq \cdots \geq \left| \frac{1}{\lambda_1} \right| > 0$$

entonces, $\frac{1}{\lambda_n}$ es el valor propio dominante de A^{-1} . Luego, podemos aplicar el método de la potencia para hallar el valor propio dominante de A^{-1} , que a la vez será el valor propio pequeño para A .

ALGORITMO 2.6.5. (Cálculo de Valores Propios Pequeños en Magnitud):

Paso 1. Aplicar el Método de la Potencia para A^{-1}

Paso 2 Tomar el recíproco del valor propio obtenido en el paso 1.

Ejemplo. Sea $A = \begin{pmatrix} 1 & 4 & 5 \\ 2 & 3 & 3 \\ 1 & 1 & 1 \end{pmatrix} \Rightarrow A^{-1} = \begin{pmatrix} 0 & -1 & 3 \\ -1 & 4 & -7 \\ 1 & -3 & 5 \end{pmatrix}$ aplicando el Método

de la Potencia a A^{-1} con el vector inicial $x_0 = (1, -1, 1)^T$. Realizando 5 iteraciones el valor propio dominante se aproxima a $\lambda = 9.1545$. Entonces, tomando la inversa a λ tendríamos el vector propio pequeño $\frac{1}{\lambda} = 0.1051$.

Capítulo 3

Sensibilidad de los Valores Propios

En este capítulo estudia la sensibilidad de los valores propios y vectores propios, el más importante resultado en este capítulo es el teorema de Bauer-Fiker en la cual empezaremos a definirlo.

Teorema 3.0.5. *[Teorema de Bauer - Fiker] Sea A diagonalizable, es decir, existe una matriz X no-singular tal que $X^{-1}AX = D$ es diagonal. Entonces para un Valor Propio λ de $A + E$ tenemos:*

$$\min_i |\lambda_i - \lambda| \leq \|X\| \|X^{-1}\| \|E\|$$

donde $\|\cdot\|$ es la norma matricial subordinada y $\lambda_1, \lambda_2, \dots, \lambda_n$ son los valores Propios de A .

Prueba:

Consideremos dos casos:

Caso 1: $\lambda = \lambda_i$ para algún i . En teorema es trivialmente verdadero.

Caso 2: $\lambda \neq \lambda_i$ para cualquier i . Entonces los elementos de la diagonal de la matriz diagonal son diferentes de cero, entonces la matriz $(\lambda I - D)$ es no singular.

De $(A + E)x = \lambda x$ tenemos:

$$\begin{aligned} Ex &= (\lambda I - XDX^{-1})x \\ &= X(\lambda I - D)X^{-1}x \end{aligned}$$

Sea $X^{-1}x = y$

$$\Rightarrow Ex = X(\lambda I - D)y$$

$$y = (\lambda I - D)^{-1}X^{-1}Ex \quad (x = Xy)$$

$$y = (\lambda I - D)^{-1}X^{-1}EXy$$

tomando norma subordinada en ambos lados, tenemos:

$$\|y\| = \|(\lambda I - D)^{-1}X^{-1}EXy\|$$

$$1 \leq \|(\lambda I - D)^{-1}\| \|X^{-1}\| \|E\| \|X\|$$

por norma subordinante

$$\|(\lambda I - D)^{-1}\| = \max \frac{1}{|\lambda - \lambda_i|}$$

$$\|(\lambda I - D)^{-1}\| = \frac{1}{\min_i |\lambda - \lambda_i|}$$

$$\Rightarrow 1 \leq \frac{1}{\min_i |\lambda - \lambda_i|} \|X^{-1}\| \|E\| \|X\|$$

$$\min_i |\lambda - \lambda_i| \leq \|X^{-1}\| \|E\| \|X\|$$

3.1. Implicaciones del Teorema

Por el teorema (3.0.5), tenemos que si:

$$\text{Cond}(X) = \|X\| \|X^{-1}\|$$

es grande, entonces un valor propio λ de la matriz perturbada $A + E$ puede ser significativamente diferente de un valor propio λ_i de A .

Ejemplo. Consideremos $A = \begin{pmatrix} 1 & 2 & 3 \\ 0 & 0.9990 & 1 \\ 0 & 0 & 2 \end{pmatrix}$

los valores propios de A son 1, 0.9990 y 2. Ahora:

$$X = \begin{pmatrix} 1 & -1 & 0.9623 \\ 0 & 0.0005 & 0.1923 \\ 0 & 0 & 0.1925 \end{pmatrix},$$

entonces:

$$X^{-1}AX = \text{diag}(1, 0.9990, 2)$$

Sea:

$$E = \begin{pmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ 10^{-5} & 0 & 0 \end{pmatrix}$$

los autovalores de $A + E$ son $0.9995 + 0.004i$, $0.9995 - 0.004i$ y 2. Los valores propios de $A + E$ que han sido variado por 10^{-5} en la posición (3,1) de la matriz A cambian totalmente, esto es debido a que $\text{Cond}(X) = 6.8708 \times 10^3$ es un valor grande.

3.2. Sensibilidad de los Valores Propios

Individuales

El $\text{Cond}(X) = \|X\| \|X^{-1}\|$ da un total enfoque de los cambios de los valores propios con respecto a los cambios en los coeficientes de la matriz siendo así algunos valores propios de A más sensibles que otros. En concreto, algunos podrían ser muy bien condicionados mientras que otros son mal condicionados. Similarmente sucede con los vectores propios.

Sea $X^{-1}AX = \text{diag}(\lambda_1, \dots, \lambda_n)$ entonces los vectores propios normalizados correspondientes a un valor propio λ_i son dados por:

$$x_i = \frac{Xe_i}{\|Xe_i\|} \quad y_i = \frac{(X^{-1})^T e_i}{\|(X^{-1})^T e_i\|_2}$$

Definición 3.2.1. El número $\frac{1}{s_i}$ donde s_i es definido por:

$$s_i = |y_i^T x_i|$$

es llamado el **Número de Condición** del valor propio λ_i . Si este número es grande, entonces λ_i es un valor propio mal condicionado.

Del ejemplo anterior:

$$X = \begin{pmatrix} 1 & -1 & 0.9623 \\ 0 & 0.0005 & 0.1923 \\ 0 & 0 & 0.1925 \end{pmatrix}$$

$$x_1 : (1,0,0)^T$$

$$x_2 : (1,-0.0005,0)^T$$

$$x_3 : (0.9623,0.1923,0.1925)^T$$

$$y_1 : (0.0004,0.7066,-0.7076)^T$$

$$y_2 : (0,-0.7025,0.7068)^T$$

$$y_3 : (0,0,1)^T$$

$$s_1 : |y_1^T x_1| = 3.5329 \times 10^{-4}$$

$$s_2 : |y_2^T x_2| = 3.5373 \times 10^{-4}$$

$$s_3 : |y_3^T x_3| = 0.1925$$

Según la definición, s_1, s_2 son grandes, $\lambda_1 = 1, \lambda_2 = 0.999$ están mal condicionados, mientras $\lambda_3 = 2$ están bien condicionado.

Relación entre s_i y $Cond_2(X)$:

Tenemos que:

$$\begin{aligned} x_i &= \frac{X e_i}{\|X e_i\|_2}, \quad y_i = \frac{(X^{-1})^T e_i}{\|(X^{-1})^T e_i\|_2} \\ \Rightarrow s_i &= |y_i^T x_i| = \frac{e_i^T X^{-1} X}{\|X e_i\|_2 \|(X^{-1})^T e_i\|_2} \\ s_i &= \frac{1}{\|X e_i\|_2 \|(X^{-1})^T e_i\|_2} \end{aligned} \quad (3.1)$$

además:

$$\begin{aligned} \|X e_i\|_2 &\leq \|X\|_2 \|e_i\|_2 = \|X\|_2 \\ \|(X^{-1})^T e_i\|_2 &\leq \|(X^{-1})^T\|_2 \|e_i\|_2 \\ \|(X^{-1})^T e_i\|_2 &= \|(X^{-1})^T\|_2 \end{aligned}$$

entonces reemplazando en (3.1):

$$\frac{1}{s_i} = \|X^T e_i\|_2 \| (X^{-1})^T e_i \|_2 \leq \|X^T\|_2 \| (X^{-1})^T \|_2$$
$$\Rightarrow \frac{1}{s_i} \leq \text{Cond}_2(X)$$

Otra relación:

$$\text{Cond}_2(X) \leq \sum_{i=1}^n |s_i^{-1}|$$

veamos que esta relación también se cumple en el ejemplo anterior:

$$A = \begin{pmatrix} 1 & 2 & 3 \\ 0 & 0.9990 & 1 \\ 0 & 0 & 2 \end{pmatrix}$$

$$\frac{1}{s_1} = 2.8305 \times 10^3, \quad \frac{1}{s_2} = 2.8270 \times 10^3, \quad \frac{1}{s_3} = 5.1940$$

Sea:

$$X = \begin{pmatrix} 53.20 & -53.17 & 2.193 \\ 0 & 0.0266 & 0.4383 \\ 0 & 0 & 0.4388 \end{pmatrix}$$

$$\Rightarrow \text{Cond}_2(X) = 5.6577 \times 10^3 \Rightarrow \frac{1}{s_1} < \text{Cond}_2(X)$$

Capítulo 4

Iteración QR

En este capítulo veremos el método de iteración QR utilizando cambio simple lo cual nos permite obtener una convergencia más rápida en el caso de valores propio reales; pero si existen valores propios complejos en este caso utilizamos el doble cambio.

4.1. Iteración Básica QR

Presentamos primero el Método de la Iteración Básica QR .

Sea $A_0 = A$

Ahora calculemos una sucesión de matrices definida por:

$$A_0 = Q_0 R_0$$

$$A_1 = R_0 Q_0 = Q_1 R_1$$

En general:

$$A_k = R_{k-1} Q_{k-1} = Q_k R_k \quad k = 1, 2, \dots$$

Las matrices en la sucesión A_k poseen propiedades interesantes, tales como que cada

matriz en la sucesión es similarmente ortogonal a la anterior, es decir:

$$\begin{aligned} A_1 &= R_0 Q_0 = Q_0^T A_0 Q_0 \quad (Q_0^T A_0 = R_0) \\ A_2 &= R_1 Q_1 = Q_1^T A_1 Q_1 \\ &\vdots \\ A_k &= R_{k-1} Q_{k-1} = Q_{k-1}^T A_{k-1} Q_{k-1} \end{aligned}$$

Además, cada matriz es similarmente ortogonal a la original, es decir:

$$\begin{aligned} A_2 &= Q_1^T A_1 Q_1 = (Q_1^T Q_0^T A_0 Q_0 Q_1) \\ A_2 &= (Q_0 Q_1)^T A_0 (Q_0 Q_1) \\ A_3 &= Q_2^T A_2 Q_2 = (Q_2^T (Q_0 Q_1)^T A_0 Q_0 Q_1 Q_2) \\ A_3 &= (Q_0 Q_1 Q_2)^T A_0 (Q_0 Q_1 Q_2) \\ &\vdots \\ A_k &= Q_{k-1}^T A_{k-1} Q_{k-1} = Q_{k-1}^T (Q_0 Q_1 Q_2 \cdots Q_{k-2})^T A_0 (Q_0 Q_1 Q_2 \cdots Q_{k-1}) \\ A_k &= (Q_0 Q_1 Q_2 \cdots Q_{k-1})^T A_0 (Q_0 Q_1 Q_2 \cdots Q_{k-1}) \end{aligned}$$

Damos ahora una condición para la convergencia:

Teorema 4.1.1. [Teorema de Convergencia para la Iteración Básica QR]

Sean los valores propios $\lambda_1, \dots, \lambda_n$ tal que: $|\lambda_1| > |\lambda_2| > \dots > |\lambda_n|$, y sea Y la matriz de vectores propios de los vectores propios izquierdos (esto es, $Y = (X^T)^{-1}$) tal que las entradas principales menores no nulas. Entonces $\{A_k\}$ converge a una matriz triangular superior o de la forma real de Schur.

Prueba: [Ver([19])]

La conclusión de este teorema:

Bajo ciertas condiciones, la primera columna de A_k , se aproxima a un múltiplo de e_1 , tal que para k suficientemente grande, obtenemos:

$$A_k = \begin{pmatrix} \lambda_1 & u \\ 0 & \hat{A}_k \end{pmatrix}$$

y luego se aplica otra vez la iteración QR para \hat{A}_k y así sucesivamente hasta que la sucesión converja a una matriz triangular superior.

jemplo. Sea $A = \begin{pmatrix} 1 & 2 \\ 3 & 4 \end{pmatrix}$

Es fácil ver que los valores propios de A son:

$$\lambda_1 = \frac{5 + \sqrt{33}}{2}, \quad \lambda_2 = \frac{5 - \sqrt{33}}{2}, \quad |\lambda_1| > |\lambda_2|$$

$k=0$

$$A_0 = A = Q_0 R_0 \text{ donde}$$

$$Q_0 = \begin{pmatrix} -0.3162 & -0.9487 \\ -0.9487 & 0.3162 \end{pmatrix}, \quad R_0 = \begin{pmatrix} -3.1623 & -4.4272 \\ 0 & -0.6325 \end{pmatrix}$$

$k=1$

$$A_1 = A = R_0 Q_0 = \begin{pmatrix} 5.2 & 1.6 \\ 0.6 & -0.2 \end{pmatrix} = Q_1 R_1 \text{ donde}$$

$$Q_1 = \begin{pmatrix} 0.9934 & -0.1146 \\ -0.11462 & -0.9934 \end{pmatrix}, \quad R_1 = \begin{pmatrix} -5.2345 & -1.5665 \\ 0 & -0.3821 \end{pmatrix}$$

$k=2$

$$A_2 = A = R_1 Q_1 = \begin{pmatrix} 5.3796 & -0.9562 \\ 0.0438 & -0.3796 \end{pmatrix} = Q_2 R_2 \text{ donde}$$

$$Q_2 = \begin{pmatrix} -1 & -0.0082 \\ -0.0081 & 1 \end{pmatrix}, \quad R_2 = \begin{pmatrix} -5.3797 & 0.9593 \\ 0 & -0.3718 \end{pmatrix}$$

$k=3$

$$A_3 = A = R_2 Q_2 = \begin{pmatrix} 5.3718 & 1.0030 \\ 0.0030 & -0.3718 \end{pmatrix} = Q_3 R_3 \text{ donde}$$

$$Q_3 = \begin{pmatrix} -1 & -0.0006 \\ -0.006 & 1 \end{pmatrix}, \quad R_3 = \begin{pmatrix} -5.3718 & -1.0028 \\ 0 & -0.3723 \end{pmatrix}$$

$k=4$

$$A_4 = A = R_3 Q_3 = \begin{pmatrix} 5.3723 & -0.4998 \\ 0.0002 & -0.3723 \end{pmatrix}$$

Aproximándose así a una matriz triangular superior y sus elementos de la diagonal

de esta matriz se aproxima a los valores propios de la matriz original.

4.2. Iteración QR de Hessenberg

Al realizar la factorización QR de una matriz se requiere $O(n^3)$ operaciones, en consecuencia, las n iteraciones del método QR requerirán $O(n^4)$ operaciones, haciéndose así este método insuficiente. Afortunadamente existe una solución para evitar demasiadas operaciones, para ello realizaremos entonces la Iteración de Hessenberg:

Reducir la matriz arbitraria a una matriz Hessenberg por similitud ortogonal antes de empezar con las iteraciones QR . Supongamos que A_k es una matriz Hessenberg superior irreducibles y las rotaciones de Givens son usadas para el factor $A_k = Q_k R_k$, donde $Q_k = (J(n-1, n, \theta) \cdots J(2, 3, \theta) J(1, 2, \theta))$ así, $A_{k+1} = R_k Q_k$ sigue siendo una matriz Hessenberg superior.

Así con este método el número de iteraciones es menor que el método anterior.

Ejemplo. Sea $A = A_0 = \begin{pmatrix} 1 & 2 & 3 \\ 2 & 3 & 5 \\ 0 & 1 & 1 \end{pmatrix}$, luego:

$$Q_0 = \begin{pmatrix} -0.4472 & 0.3651 & -0.8165 \\ 0.8944 & -0.1826 & 0.4082 \\ 0 & 0.9129 & 0.4082 \end{pmatrix} \quad R_0 = \begin{pmatrix} -2.2361 & -3.5777 & -5.8138 \\ 0 & 1.0954 & 1.0954 \\ 0 & 0 & 0 \end{pmatrix}$$

$k=1$

$$A_1 = A = R_0 Q_0 = \begin{pmatrix} 4.2000 & -5.4705 & -2.0083 \\ -0.9798 & 0.8000 & 0.8944 \\ 0 & 0 & 0 \end{pmatrix}$$

$$Q_1 = \begin{pmatrix} -0.9739 & -0.2272 & 0 \\ 0.2272 & -0.9739 & 0 \\ 0 & 0 & 0 \end{pmatrix}$$

$$R_1 = \begin{pmatrix} -4.3128 & 5.5092 & 2.1590 \\ 0 & 0.4637 & -0.4148 \\ 0 & 0 & 0 \end{pmatrix}$$

$k=2$

$$A_2 = A = R_1 Q_1 = \begin{pmatrix} 5.4516 & -4.3854 & 2.1590 \\ 0.1054 & -0.4516 & -0.4148 \\ 0 & 0 & 0 \end{pmatrix}$$
$$Q_2 = \begin{pmatrix} -0.9998 & 0.0193 & 0 \\ -0.0193 & -0.9998 & 0 \\ 0 & 0 & 1 \end{pmatrix}$$

Se observa que cada Q_i y A_i son Hessenberg superior, además los valores propios de A_2 son 5.3723, 0, -0.3722 y los de A son 5.3723, 0 y -0.3723.

4.3. Desplazamiento Simple en la Iteración QR

Para implementar un cambio simple en la iteración QR necesitamos tener una aproximación $\hat{\lambda}_i$ del valor propio λ_i . Como $\hat{\lambda}_i$ no está disponible en la práctica, $h_{n,n}^0$ el elemento (n, n) de $H = H_0$ es tomada como aproximación inicial y entonces la iteración es continuada usando sucesivamente los elementos (n, n) como un nuevo cambio. Veamos entonces la modificación del Método de Iteración QR conocido como *desplazamiento simple en la Iteración QR*

Previamente veamos el siguiente algoritmo:

ALGORITMO 4.3.1. (Reducción a una matriz Hessenberg superior por el Método de Givens, donde A es cualquier matriz):

Paso 1 Para $p = 1, 2, \dots, n - 2$ hacer

Para $q = p + 2, \dots, n$ hacer

a) *Encontrar $c = \cos(\theta)$ y $s = \sin(\theta)$ tal que:*

$$\begin{pmatrix} c & s \\ -s & c \end{pmatrix} \begin{pmatrix} a_{p+1} \\ a_{qp} \end{pmatrix}$$

b) *Guardar s y c y los índices p y q .*

c) $A = J(p + 1, q, \theta) A J(p + 1, q, \theta)^T$

ALGORITMO 4.3.2. (Desplazamiento Simple):

Paso 1 Transformar A a una matriz Hessenberg superior H .

Paso 2 Sea $H_0 = H$

Para $k = 0, 1, 2, \dots$ hacer hasta que $h_{n,n-1}^k$ converja a cero

$$H_k - h_{nn}^{(k)}I = Q_k R_k$$

$$H_{k+1} = R_k Q_k + h_{nn}^{(k)}I$$

$h_{ij}^{(k)}$ es el elemento (i, j) de H_k

Ejemplo. Sea $H = H_0 = \begin{pmatrix} 1 & 1 & 1 \\ 1 & 2 & 3 \\ 0 & 1 & 1 \end{pmatrix}$

$k=0$

$$H_0 - h_{33}^{(0)}I = \begin{pmatrix} 0 & 1 & 1 \\ 1 & 1 & 3 \\ 0 & 1 & 0 \end{pmatrix} = Q_0 R_0$$

$$Q_0 = \begin{pmatrix} 0 & 0.7071 & -0.7071 \\ -1 & 0 & 0 \\ 0 & 0.7071 & 0.7071 \end{pmatrix}$$

$$R_0 = \begin{pmatrix} -1 & -1 & -3 \\ 0 & 1.4142 & 0.7071 \\ 0 & 0 & -0.7071 \end{pmatrix}$$

$$H_1 = R_0 Q_0 + h_{33}^0 I = \begin{pmatrix} 2 & -2.8284 & -1.4142 \\ -1.4142 & 1.5 & 0.5 \\ 0 & -0.5 & 0.5 \end{pmatrix}$$

k=1

$$H_1 - h_{33}^{(1)}I = Q_1 R_1$$

$$Q_1 = \begin{pmatrix} -0.7276 & -0.6342 & -0.2615 \\ 0.6860 & -0.6727 & -0.2774 \\ 0 & -0.3812 & 0.9245 \end{pmatrix}$$

$$R_1 = \begin{pmatrix} -2.0616 & 2.7440 & 1.3720 \\ 0 & 1.3117 & 0.5606 \\ 0 & 0 & 0.2311 \end{pmatrix}$$

$$H_2 = R_1 Q_1 + h_{33}^1 I = \begin{pmatrix} 3.8824 & -1.0613 & 1.0464 \\ 0.8998 & -0.5960 & 0.1544 \\ 0 & -0.0881 & 0.7137 \end{pmatrix}$$

k=2

$$H_2 - h_{33}^{(2)}I = Q_2 R_2$$

$$Q_2 = \begin{pmatrix} -0.9620 & 0.2721 & 0.0247 \\ -0.2732 & -0.9580 & -0.0870 \\ 0 & -0.0905 & 0.9959 \end{pmatrix}$$

$$R_2 = \begin{pmatrix} -3.2940 & 1.3787 & -1.0488 \\ 0 & 0.9740 & 0.1367 \\ 0 & 0 & 0.0124 \end{pmatrix}$$

$$H_3 = R_2 Q_2 + h_{33}^2 I = \begin{pmatrix} 3.5057 & -2.1221 & -1.2459 \\ -0.2661 & -0.2318 & 0.0514 \\ 0 & -0.0011 & 0.7260 \end{pmatrix}$$

$k=3$

$$H_3 - h_{33}^{(3)}I = Q_3 R_3$$

$$Q_3 = \begin{pmatrix} -0.9955 & -0.0953 & -0.0001 \\ 0.0953 & -0.9955 & -0.0010 \\ 0 & -0.0010 & 1 \end{pmatrix}$$

$$R_3 = \begin{pmatrix} -2.7924 & 2.0212 & 1.2451 \\ 0 & 1.1556 & 0.0675 \\ 0 & 0 & 0.0001 \end{pmatrix}$$

$$H_4 = R_3 Q_3 + h_{33}^3 I = \begin{pmatrix} 3.6983 & -1.7472 & 1.2434 \\ 0.1101 & -0.4244 & 0.0664 \\ 0 & 0 & 0.7261 \end{pmatrix}$$

en donde $h_{33}^{(4)}$ converge al autovalor 0.7261

4.4. El Doble Desplazamiento de la Iteración QR

Si los valores propios de una matriz son reales, la iteración de desplazamiento simple QR funciona bien, sin embargo, puede darse el caso que una matriz real pueda tener valores propios complejos. En tal caso:

1. En alguna etapa de la iteración podríamos encontrar que la submatriz 2×2 del lado derecho posea valores propios complejos.
2. El elemento (n, n) que se encuentra en la submatriz 2×2 la cual es real no sería una buena aproximación.
3. Es usual usar los valores propios de la submatriz 2×2 como parámetros de desplazamiento, la cual nos lleva a una iteración QR de doble desplazamiento.

Un paso de la iteración de doble cambio QR (complejo).

Sean los valores propios de la submatriz 2×2 de lado derecho de la matriz Hessenberg k_1 y $k_2 = k_1$, entonces un paso de la iteración de doble cambio QR es:

$$H_s - k_1 I = Q_s R_s \quad H_{s+1} = R_s Q_s + k_1 I$$

$$H_{s+1} - k_2 I = Q_{s+1} R_{s+1} \quad H_{s+2} = R_{s+1} Q_{s+1} + k_2 I$$

Ejemplo. Sea $H = H_0 = \begin{pmatrix} 1 & 2 & 2 \\ 0 & 0 & 1 \\ 0 & -1 & 0 \end{pmatrix}$ $k_1 = i, \quad k_2 = -i$

$$H_0 - k_1 I = Q_0 R_0$$

$$Q_0 = \begin{pmatrix} -1 & 0 & 0 \\ 0 & -0.7071 & -0.7071i \\ 0 & 0.7071i & 0.7071 \end{pmatrix}$$

$$R_0 = \begin{pmatrix} -1 + i & -2 & -2 \\ 0 & 1.4142i & -1.4142 \\ 0 & 0 & 0 \end{pmatrix}$$

$$H_1 = R_0 Q_0 + k_1 I = \begin{pmatrix} 1 & 1.4142 - 1.4142i & 1.4142 + 1.4142i \\ 0 & -i & 0 \\ 0 & 0 & i \end{pmatrix}$$

$$H_1 - k_2 I = Q_1 R_1$$

$$Q_1 = \begin{pmatrix} -1 & 0 & 0 \\ 0 & -0.8165 & -0.5744i \\ 0 & 0.5744i & 0.8165 \end{pmatrix}$$

$$R_1 = \begin{pmatrix} -1 - i & -1.4142 + 1.4142i & 1.4142 - 1.4142i \\ 0 & 0 & 1.1547 \\ 0 & 0 & 1.6330i \end{pmatrix}$$

$$H_2 = R_1 Q_1 + k_2 I = \begin{pmatrix} 2 & 1.9712 - 0.3382i & 1.9712 - 0.3382i \\ 0 & -0.3333i & 0.9428 \\ 0 & -0.9428 & 0.333i \end{pmatrix}$$

Los valores propios de H_2 son 1, i y $-i$.

Evitando la aritmética compleja en el doble desplazamiento de la iteración QR .

Debido a que k_1 y k_2 pueden ser números complejos en el método anterior, sería un poco complicado, sin embargo con una pequeña modificación se evitaran estos cálculos. Demostremos primero que la matriz H_{s+2} es similarmente ortogonal a H_s . Consideremos la matriz:

$$N = (H_s - k_2 I)(H_s - k_1 I) = H_s^2 - (k_1 + k_2)H_s + k_1 k_2 I$$

como $k_2 = \overline{k_1}$, entonces la matriz N es real. Previamente demostraremos que $(Q_s, Q_{s+1})(R_{s+1}, R_s)$ es la factorización QR de N .

$$\begin{aligned} N &= (H_s - k_2 I)(H_s - k_1 I) \\ &= (H_s - k_2 I)Q_s R_s \\ &= Q_s \underbrace{Q_s^* (H_s - k_2 I) Q_s}_{H_{s+1} - k_2 I} R_s \\ &= Q_s (H_{s+1} - k_2 I) R_s \\ N &= Q_s Q_{s+1} R_{s+1} R_s \end{aligned}$$

Debido a que N es real y $(Q_s Q_{s+1})(R_{s+1} R_s)$ es la factorización QR de N , la matriz $Q_s Q_{s+1}$ puede ser elegida real y ortogonal. Entonces:

$$\begin{aligned} H_{s+2} &= \underbrace{R_{s+1} Q_{s+1}} + k_2 I \\ H_{s+2} &= \underbrace{H_s + 2 - k_2 I + k_2 I}_{Q_{s+1}^* (H_{s+1} - k_2 I) Q_{s+1} + k_2 I} \\ &= Q_{s+1}^* [R_s Q_s + k_1 I - k_2 I] Q_{s+1} + k_2 I \\ &= Q_{s+1}^* [R_s Q_s + (k_1 - k_2) I] Q_{s+1} + k_2 I \\ &= Q_{s+1}^* [\underbrace{H_{s+1} - k_1 I}_{Q_s^* (H_s - k_1 I) Q_s} + (k_1 - k_2) I] Q_{s+1} + k_2 I \\ &= Q_{s+1}^* [Q_s^* (H_s - k_1 I) Q_s + (k_1 - k_2) I] Q_{s+1} + k_2 I \\ &= Q_{s+1}^* Q_s^* H_s Q_s Q_{s+1} \\ H_{s+2} &= (Q_s Q_{s+1})^T H_s (Q_s Q_{s+1}) \end{aligned}$$

Entonces H_{s+2} es similarmente ortogonal a H_s y además, como $Q_s Q_{s+1}$ y H_s son reales, H_{s+2} es real.

k_1 y k_2 no necesitan ser calculado explícitamente debido a que:

$$N = (H_s - k_1 I)(H_s - k_2 I) = H_s^2 - (k_1 + k_2)H_s + k_1 k_2 I$$

$$t = k_1 + k_2 = h_{n-1,n-1} + h_{nn}$$

$$d = k_1 k_2 = h_{n-1,n-1} h_{nn} - h_{n,n-1} h_{n-1,n}$$

ALGORITMO 4.4.1. *Un Paso de la Iteración de Doble Desplazamiento QR (Aritmética Real)*

1. Formar la matriz $N = H_s^2 - tH_s + dI$
2. Encontrar la factorización QR de N : $N = QR$
3. Formar $H_{s+2} = Q^T H_s Q$

Ejemplo. Sea $H = H_0 = \begin{pmatrix} 1 & 2 & 3 \\ 1 & 0 & 1 \\ 0 & -2 & 2 \end{pmatrix}$, $t = 2$, $d = 2$

$$N = H^2 - tH + dI = \begin{pmatrix} 3 & -8 & 5 \\ -1 & 2 & 3 \\ -2 & 0 & 0 \end{pmatrix}$$

encontrando la factorización QR de N :

$$Q = \begin{pmatrix} -0.8018 & -0.5470 & -0.2408 \\ 0.2673 & 0.0322 & -0.9631 \\ 0.5345 & -0.8365 & 0.1204 \end{pmatrix}$$

$$H_2 = Q^T H_0 Q = \begin{pmatrix} -0.8571 & 1.1007 & 2.5740 \\ -1.1867 & 3.0455 & -0.8289 \\ 0.0000 & 1.8437 & 0.8116 \end{pmatrix}$$

Veamos ahora un algoritmo más eficiente que el (4.4.1):

ALGORITMO 4.4.2. *Un Paso de Iteración para el Doble Desplazamiento QR implícito.* Sea $H_{n \times n}$ una matriz de Hessenberg irreducida. Entonces el siguiente

algoritmo, el cual constituye una iteración del Doble Desplazamiento Implícito QR, produce matrices ortogonales P_0, P_1, \dots, P_{n-2} tal que: $Q^T H Q$ donde $Q = P_0 P_1 \dots P_{n-2}$ es una matriz Hessenberg superior. El algoritmo sobrescribe H con $Q^T H Q$.

Paso 1 Calcular los desplazamientos:

$$t = h_{n-1,n-1} + h_{nn}$$

$$d = h_{n-1,n-1} h_{nn} - h_{n,n-1} h_{n-1,n}$$

Paso 2 Calcular las tres primeras entradas de la primera columna de $N = H^2 - tH + dI$

$$x = n_{11} = h_{11}^2 - t h_{11} + d + h_{12} h_{21}$$

$$y = n_{21} = h_{21} (h_{11} + h_{12} - t)$$

$$z = n_{31} = h_{21} h_{32}$$

Paso 3 Calcular las matrices de Householder P_0, P_1, \dots, P_{n-2} tal que la matriz final es Hessenberg superior.

a) For $k = 0, 1, 2, \dots, n - 3$ hacer

Encontrar una matriz de Householder 3×3 , \hat{P}_k , tal que:

$$\hat{P}_k \begin{pmatrix} x \\ y \\ z \end{pmatrix} = \begin{pmatrix} * \\ 0 \\ 0 \end{pmatrix}$$

y entendiendo H con $P_k H P_k^T$, donde:

$$P_k = \begin{pmatrix} I_k & & 0 \\ & \hat{P}_k & \\ 0 & & I_{n-k-3} \end{pmatrix}$$

Tomar:

$$x \equiv h_{k+2,k+1}$$

$$y \equiv h_{k+3,k+1}$$

$$z \equiv h_{k+4,k+1} \quad (\text{si } k < n - 3)$$

b) Encontrar la matriz de Householder \widehat{P}_{n-2} de orden 2 tal que:

$$\widehat{P}_{n-2} \begin{pmatrix} x \\ y \end{pmatrix} = \begin{pmatrix} * \\ 0 \end{pmatrix}$$

Formar:

$$P_{n-2} = \begin{pmatrix} I_{n-2} & 0 \\ 0 & \widehat{P}_{n-2} \end{pmatrix}$$

Capítulo 5

El Problema de Valor Propio Generalizado

5.1. Introducción

En este capítulo consideramos el siguiente problema conocido como el **problema de valor propio generalizado**.

Definición 5.1.1. Sean $A, B \in \mathbb{C}(n, n)$. Para el problema $Ax = \lambda Bx$, λ es llamado *valor propio generalizado de A (si existe)* y el vector x es un *vector propio asociado con λ* .

Observación. Si λ existe, entonces es una raíz de la ecuación característica:

$$\det(A - \lambda B) = 0$$

Definición 5.1.2. La matriz $A - \lambda B$ es llamada *una matriz pencil*. Es conveniente denotar por (A, B) .

El par (A, B) es llamado *regular* si $\det(A - \lambda B) \neq 0$, y es llamado *singular* si $\det(A - \lambda B) = 0$.

5.1.1. Valores Propios de una Matriz Pencil

Proposición 5.1.1. *Sea (A, B) una matriz pencil regular. Entonces:*

1. *Si B es no singular, entonces todos los valores propios de (A, B) son finitos y son los mismos para AB^{-1} y para $B^{-1}A$.*

2. *Si B es singular, entonces:*

el $\text{grad}(P(\lambda)) = \text{grad}(\det(A - \lambda B)) = r \leq n$. Los r ceros de $P(\lambda)$ son los valores propios del par (A, B) .

Denotaremos los $(n - r)$ valores propios sobrantes como ∞ .

Prueba:

[1]:

Como B es no singular, entonces $Ax = \lambda Bx$

$$\underbrace{B^{-1}Ax}_{A'x} = \lambda x$$

Estamos en un problema $Ax = \lambda x$, donde λ es finito. Ahora veamos que los valores propios de AB^{-1} y $B^{-1}A$ son los mismos.

En efecto, de:

$$AB^{-1} = BB^{-1}AB^{-1}$$

se observa que las matrices AB^{-1} y $B^{-1}A$ son similares, por lo tanto, tienen los mismos valores propios.

[2]

Prueba: [Ver ([2])]

Definición 5.1.3. *Si A y B son matrices simétricas reales, y además si B es definida positiva, entonces el problema de valor propio generalizado $Ax = \lambda Bx$ es llamada el problema del valor propio generalizado definida simétrica.*

5.1.2. Descomposición Generalizada de Schur

Para resolver el problema del valor propio generalizado, debemos reducir simultáneamente las matrices A y B en la forma más conveniente, de la cual, el problema del valor propio generalizado sea fácilmente extraído.

El siguiente teorema ilustra la extracción del valor propio generalizado desde que las matrices han sido reducidas en forma conveniente.

Teorema 5.1.1. Sean $A, B \in \mathbb{C}(n, n)$ y el par (A, B) regular. Sean U_1, U_2 matrices unitarias tal que:

$$U_1 A U_2 = T_1 = \begin{pmatrix} t_{11} & & & \\ & \ddots & & \\ & & * & \\ & & & \ddots & \\ & & & & t_{nn} \end{pmatrix}$$

$$U_1 B U_2 = T_2 = \begin{pmatrix} t'_{11} & & & \\ & \ddots & & \\ & & * & \\ & & & \ddots & \\ & & & & t'_{nn} \end{pmatrix}$$

entonces los autovalores generalizados $\lambda_i \quad i = 1, \dots, n$ de la pencil regular (A, B) son dados por:

$$\begin{cases} \lambda_i = \frac{t_{ii}}{t'_{ii}} & \text{si } t'_{ii} \neq 0 \\ \lambda_i = \infty & \text{si } t_{ii} \neq 0, t'_{ii} = 0 \end{cases}$$

Prueba:

De $Ax = \lambda Bx$ tenemos:

$$U_1 A U_2 U_2^* x = \lambda U_1 B U_2 U_2^* x$$

definimos:

$$\begin{aligned} y &= U_2^* x \\ U_1 A U_2 y &= \lambda U_1 B U_2 y \\ \Rightarrow T_1 y &= \lambda T_2 y \end{aligned}$$

de tal manera que λ es un valor propio del (A, B) si y sólo si es un valor propio del par (T_1, T_2) .

Como T_1 y T_2 son triangulares:

$$\det(T_1 - \lambda T_2) = \prod_{i=1}^n (t_{ii} - \lambda t'_{ii})$$

luego:

$$\det(T_1 - \lambda T_2) = 0 \Leftrightarrow (t_{ii} - \lambda t'_{ii}) = 0 \Leftrightarrow \begin{cases} \lambda_i = \frac{t_{ii}}{t'_{ii}} & \text{si } t'_{ii} \neq 0 \\ \lambda_i = \infty & \text{si } t_{ii} \neq 0, t'_{ii} = 0 \end{cases}$$

5.1.3. Algoritmo QZ

El algoritmo de iteración QZ es similar al de iteración QR para el cálculo de valores propios.

Asumamos que la matriz B es no singular. Entonces, la idea básica es aplicar la iteración QR a la matriz $C = B^{-1}A$ ó AB^{-1} sin calcular independientemente a B^{-1} y a C .

La mejor forma es transformar a A y a B a su forma reducida.

Teorema 5.1.2. *Descomposición Generalizada de Schur Sean $A, B \in \mathbb{R}(n, n)$, existen matrices ortogonales Q y Z tal que:*

$$\begin{aligned} A' &= Q^T A Z, & A' &= \text{matriz real de Schur superior} \\ B' &= Q^T B Z, & B' &= \text{matriz triangular superior} \end{aligned}$$

el par (A', B') es llamado la forma generalizada de Schur y tiene los mismos valores propios del par (A, B) .

5.1.4. Reducción a una Forma Hessenberg - Triangular

Sea $A, B \in \mathbb{C}(n, n)$, entonces:

1. Encontrar una matriz U tal que $U^T B$ sea triangular superior.
Calcular $U^T A$

2. Reducir a A a la forma Hessenberg, y a la vez preservar la estructura triangular de B

Los pasos a seguir son:

$$B := U^T B = \begin{pmatrix} * & * & \dots & * \\ 0 & * & \dots & * \\ \vdots & \vdots & & \vdots \\ 0 & * & \dots & * \\ 0 & 0 & \dots & 0 & * \end{pmatrix} \quad A := U^T A = \begin{pmatrix} * & * & \dots & * \\ * & * & \dots & * \\ \vdots & \vdots & & \vdots \\ * & * & \dots & * \\ * & * & \dots & * \end{pmatrix}$$

El elemento $(n, 1)$ de A se hace cero, aplicando una rotación $Q_{n-1,n}$.

$$A := Q_{n-1,n} A = \begin{pmatrix} * & * & \dots & * \\ * & * & \dots & * \\ \vdots & \vdots & & \vdots \\ * & * & \dots & * \\ 0 & * & \dots & * \end{pmatrix}$$

multipliquemos esta rotación a B por el lado izquierdo, la cual la posición $(n, n - 1)$ se hace diferente de cero.

$$B \equiv Q_{n-1,n} B = \begin{pmatrix} * & * & \dots & * \\ 0 & * & \dots & * \\ \vdots & \vdots & & \vdots \\ 0 & * & \dots & * \\ 0 & 0 & \dots & * & * \end{pmatrix}$$

la rotación $Z_{n-1,n}$ se le multiplica a B por el lado derecho para hacer el elemento $(n, n - 1)$ de B cero. Afortunadamente esta rotación cuando se multiplica por el lado derecho a A no afecta a los ceros producidos anteriormente, es decir:

$$B := B Z_{n-1,n} = \begin{pmatrix} * & * & \dots & * \\ 0 & * & \dots & * \\ \vdots & \vdots & & \vdots \\ 0 & * & \dots & * \\ 0 & 0 & \dots & 0 & * \end{pmatrix}$$

$$A \equiv AZ_{n-1,n} = \begin{pmatrix} * & * & \dots & * \\ * & * & \dots & * \\ \vdots & \vdots & & \vdots \\ * & * & \dots & * \\ 0 & * & \dots & * \end{pmatrix}$$

luego, sucesivamente se hacen cero a los elementos $(n-1, 1), (n-2, 1), \dots, (3, 1)$, como sigue:

Primero una rotación apropiada es multiplicado por el lado izquierdo para ser esa posición cero.

Segundo, la misma rotación es multiplicada por el lado izquierdo a B .

Finalmente, otra rotación apropiado es multiplicada por el lado derecho a B .

Al final se obtiene la matriz A Hessenberg en su primera columna y B es triangular superior.

Luego, el proceso es realizado para las siguientes columnas de A hasta obtener una Hessenberg superior y B sea una triangular superior.

Ejemplo. Sea $A = \begin{pmatrix} 1 & 2 & 3 \\ 1 & 3 & 4 \\ 1 & 3 & 3 \end{pmatrix}$, $B = \begin{pmatrix} 1 & 1 & 1 \\ 0 & 1 & 2 \\ 0 & 0 & 2 \end{pmatrix}$

Paso 1: Formar Q_{23} al hacer a_{31} cero:

$$Q_{23} = \begin{pmatrix} 1 & 0 & 0 \\ 0 & 0.7071 & 0.7071 \\ 0 & -0.7071 & 0.7071 \end{pmatrix}$$

$$A \equiv A^{(1)} = Q_{23}A = \begin{pmatrix} 1 & 2 & 3 \\ 1.4142 & 4.2426 & 4.9497 \\ 0 & 0 & -0.7071 \end{pmatrix}$$

Paso 2: Actualizar B :

$$B \equiv B^{(1)} = Q_{23}B = \begin{pmatrix} 1 & 1 & 1 \\ 0 & 0.7071 & 2.8284 \\ 0 & -0.7071 & 0 \end{pmatrix}$$

Paso 3: Formar Z_{23} al hacer b_{32} cero:

$$Z_{23} = \begin{pmatrix} 1 & 0 & 0 \\ 0 & 0 & -1 \\ 0 & 1 & 0 \end{pmatrix}$$

$$B \equiv B^{(1)}Z_{23} = Q_{23}BZ_{23} = \begin{pmatrix} 1 & 1 & -1 \\ 0 & 2.8284 & -0.7071 \\ 0 & 0 & 0.7071 \end{pmatrix}$$

Paso 4: Actualizar A :

$$A \equiv A^{(1)}Z_{23} = Q_{23}AZ_{23} = \begin{pmatrix} 1 & 3 & -2 \\ 1.4142 & 4.9497 & -4.2426 \\ 0 & -0.7071 & 0 \end{pmatrix}$$

obteniéndose así una Hessenberg superior y B de la forma triangular superior.

ALGORITMO 5.1.1. [Reducción a la forma generalizada de Schur]

Dada A , una matriz hessenberg superior irreducida y B una matriz triangular superior. Los siguientes pasos construyen matrices ortogonales Q y Z tal que $Q^T AZ$ es una matriz Hessenberg superior y $Q^T BZ$ es una matriz triangular superior.

Paso 1 : Escoger los desplazamientos α_1 y α_2 .

Paso 2 : Calcular la primera columna de:

$$N = (c - \alpha_1 I)(c - \alpha_2 I)$$

donde $c = AB^{-1}$, sin formar explícitamente B^{-1} .

Sea (c_1, c_2) las dos primeras columnas de c , entonces:

$$(c_1, c_2) = (a_1, a_2) \begin{pmatrix} b_{11} & b_{12} \\ 0 & b_{22} \end{pmatrix}^{-1}$$

Los tres elementos diferentes de cero de la primera columna de N son dados por:

$$\begin{aligned} x &= (c_{11} - \alpha_1)(c_{11} - \alpha_2) + c_{12}c_{21} \\ y &= c_{21}(c_{11} - \alpha_2) + c_{21}(c_{22} - \alpha_1) + c_{12}c_{21} \\ z &= c_{21}c_{32} \end{aligned}$$

La primera columna de N es $n_1 = \begin{pmatrix} x \\ y \\ z \\ 0 \\ \vdots \\ 0 \end{pmatrix}$

Paso 3 : Encontrar una matriz Housholder Q tal que $Qn_1 = \begin{pmatrix} * \\ 0 \\ \vdots \\ 0 \end{pmatrix}$

Paso 4 : Calcular QA y QB .

Paso 5 : Transformar QA y QB a una matriz Hessenberg superior A_1 y una matriz triangular B , por equivalencias ortogonales.

ALGORITMO 5.1.2. [El Algoritmo QZ]

Dada las matrices reales A y B .

Paso 1: Transformar (A,B) a una matriz triangular Hessenberg por equivalencias ortogonales:

$A \equiv Q^T AZ$: una matriz Hessenberg superior

$B \equiv Q^T BZ$: una matriz triangular superior

Paso 2: Iterar con el algoritmo (5.1.1) para obtener $\{A_k\}$ y $\{B_k\}$, eligiendo desplazamientos para cada iteración como se describió anteriormente.

Paso 3: Observar la convergencia de las sucesiones $\{A_k\}$ y $\{B_k\}$:

$\{A_k\} \rightarrow R$ quasi - triangular

$\{B_k\} \rightarrow T$ triangular

ALGORITMO 5.1.3. [La Iteración Inversa Para Vectores Propios de la Pencil (A, B)]

Cada vez que una aproximación de un valor propio λ es calculado, el correspondiente vector propio B puede ser calculado usando la Iteración Inversa como antes.

Paso 1: Elegir un autovector inicial v_0 .

Paso 2: Para $k = 1, 2, \dots$ hacer hasta que converja:

$$\begin{aligned} & \text{Resolver } (A - \lambda B)\hat{v}_k = Bv_{k-1} \\ v_k &= \frac{\hat{v}_k}{\|\hat{v}_k\|_2} \end{aligned}$$

Observación Para resolver el sistema $(A - \lambda B)\hat{v}_k = Bv_{k-1}$, el par (A, B) es reducido como una parte del algoritmo QZ.

Ejemplo.

Aplicaremos el algoritmo QZ para las siguientes matrices A y B :

$$A = \begin{pmatrix} 1 & 5 & 3 & 5 & 4 & -2 & -1 \\ -5 & 1 & 3 & -4 & 3 & 2 & -6 \\ 2 & -7 & -6 & -5 & 5 & -3 & -3 \\ 1 & 2 & 9 & -3 & -8 & 1 & 2 \\ -7 & 9 & 6 & 5 & -3 & -2 & 5 \\ -5 & 2 & 6 & -3 & 4 & -2 & -7 \\ -6 & 2 & 1 & -1 & 5 & -4 & -3 \end{pmatrix}$$
$$B = \begin{pmatrix} 8 & 5 & 3 & 5 & 4 & 8 & 5 \\ -5 & 1 & -3 & -4 & 3 & 5 & 4 \\ 2 & -7 & -6 & 5 & 5 & -2 & -5 \\ 1 & -2 & 9 & -3 & -8 & 7 & -9 \\ -7 & 9 & 4 & 5 & -3 & -8 & 2 \\ 5 & -4 & 7 & -3 & 9 & -2 & 1 \\ 2 & -6 & 2 & -4 & -1 & 4 & -5 \end{pmatrix}$$

1. Hallamos una matriz ortogonal Q tal que $B = Q^T B$ sea triangular superior, para esto aplicamos el programa "paso1QZ.m" (ver apéndice) y se obtiene:

$$Q = \begin{pmatrix} 0.6100 & -0.6390 & -0.1291 & 0.0332 & 0.0232 & 0.4218 & 0.1530 \\ -0.3812 & 0.0905 & -0.0963 & -0.4473 & 0.3962 & 0.6910 & -0.0512 \\ 0.1525 & 0.4579 & -0.3576 & 0.7044 & 0.1632 & 0.3303 & -0.0855 \\ 0.0762 & 0.1167 & 0.6934 & 0.1191 & -0.4295 & 0.4207 & -0.3522 \\ -0.5337 & -0.4422 & 0.3144 & 0.5272 & 0.2330 & 0.0368 & 0.2951 \\ 0.3812 & 0.1341 & 0.4652 & -0.0284 & 0.7512 & -0.2006 & -0.1225 \\ 0.1525 & 0.3830 & 0.2239 & -0.0986 & -0.1131 & 0.1300 & 0.8605 \end{pmatrix}$$

luego, actualizamos A y B :

$$A = Q^T A = \begin{pmatrix} 3.8125 & -1.9825 & -0.3050 & -0.3812 & 5.3374 & -2.6687 & -4.4225 \\ 0.0679 & -9.0221 & -4.8088 & -9.1927 & 2.8501 & -0.7138 & -5.3433 \\ -5.5394 & 7.3560 & 12.6114 & -0.6000 & -6.1035 & -0.6231 & 0.8106 \\ 0.8407 & -0.4828 & -1.5022 & 0.8957 & -0.8286 & -3.5582 & 3.9065 \\ -6.7695 & 1.8842 & 2.2054 & -1.9716 & 7.2740 & -1.6893 & -7.5031 \\ -1.9864 & 1.5192 & 4.2902 & -2.9130 & 1.7830 & -0.2241 & -3.5191 \\ -6.7303 & 4.7402 & -0.4553 & 3.4370 & 5.7757 & -4.2914 & -0.5419 \end{pmatrix}$$

$$B = Q^T B = \begin{pmatrix} 13.1149 & -5.7949 & 3.5837 & 0.6862 & 6.3287 & 7.3199 & -1.3725 \\ -0.0000 & -13.3573 & -3.9504 & -5.7629 & 1.2222 & 0.0435 & -8.8376 \\ -0.0000 & -0.0000 & 13.2496 & -4.8473 & -5.1211 & 1.5049 & -5.5091 \\ 0.0000 & 0.0000 & -0.0000 & 8.2354 & -0.3792 & -7.1008 & -4.6984 \\ 0.0000 & 0.0000 & 0.0000 & 0.0000 & 11.7083 & -4.9849 & 6.5332 \\ 0.0000 & -0.0000 & -0.0000 & 0.0000 & -0.0000 & 9.7407 & -1.3420 \\ 0 & -0.0000 & 0 & 0.0000 & -0.0000 & 0.0000 & 0.3230 \end{pmatrix}$$

2. Ahora, transformamos la matriz A a su forma Hessemberg manteniendo la estructura triangular de B , para esto aplicamos el programa "paso2QZ.m" (ver apéndice) y se obtiene:

$$A = \begin{pmatrix} 3.8125 & 3.0452 & -3.3226 & -0.9603 & -4.0312 & -4.3885 & -1.6102 \\ -11.2457 & 7.4388 & 3.0151 & -10.1202 & -0.1618 & -2.6750 & -4.9581 \\ 0.0000 & 6.5173 & -11.0905 & 4.2389 & -6.6369 & -7.1637 & 4.4832 \\ -0.0000 & -0.0000 & -6.0322 & 8.1213 & -2.1098 & -0.1959 & -5.6755 \\ 0.0000 & 0.0000 & 0.0000 & -3.1752 & 0.1072 & 6.9082 & -4.9140 \\ -0.0000 & -0.0000 & -0.0000 & 0 & 1.0968 & 1.8004 & -5.5178 \\ -0.0000 & -0.0000 & -0.0000 & 0 & 0 & 0.3583 & -3.4808 \end{pmatrix}$$

$$B = \begin{pmatrix} 13.1149 & -3.0836 & -8.9229 & -6.1131 & -1.8324 & -0.5838 & 3.4954 \\ 0.0000 & -0.3009 & -1.5608 & -6.1974 & 3.6089 & -4.5608 & -0.3648 \\ 0.0000 & -0.0000 & -13.1800 & 7.9498 & 0.9463 & -2.3897 & -1.5774 \\ -0.0000 & -0.0000 & -0.0000 & 7.0545 & -3.5840 & -2.8459 & -6.6586 \\ 0.0000 & -0.0000 & -0.0000 & 0.0000 & 12.8017 & 5.0540 & -7.5811 \\ 0.0000 & -0.0000 & -0.0000 & 0.0000 & 0.0000 & 10.0400 & -0.1681 \\ 0.0000 & 0.0000 & 0.0000 & -0.0000 & -0.0000 & -0.0000 & -14.9303 \end{pmatrix}$$

3. Para aceptar que un elemento $A_{i,i-1}$ es computacionalmente cero, aplicaremos el siguiente criterio sugerido en la bibliografía. (Datta)

$$|A_{i,i-1}| \leq \epsilon (|A_{i,i}| + |A_{i-1,i-1}|) \quad (5.1)$$

donde ϵ es una tolerancia dada. Este criterio nos servirá para decidir si un valor propio es real o complejo. Continuando con el ejemplo, tenemos:

a) Observar: $A_{2,1} = -11.2457$ no cumple la condición (5.1). Iterando con el algoritmo QZ, se obtiene:

$$A = \begin{pmatrix} 8.4482 & 0.9367 & -9.6300 & 3.9607 & 2.6008 & 10.6868 & 0.8043 \\ 0.0001 & 19.6255 & 2.4144 & -1.7300 & 1.3157 & -2.4753 & -3.7515 \\ -0.0000 & 0.0335 & -1.0283 & -0.8193 & 7.1137 & 1.1939 & -3.4662 \\ 0.0000 & -0.0000 & 1.4488 & -1.6798 & 0.3264 & 6.0254 & 1.1805 \\ -0.0000 & -0.0000 & 0.0000 & -0.8018 & -1.3326 & -0.5242 & 5.7838 \\ 0.0000 & -0.0000 & 0.0000 & -0.0000 & -0.3565 & 2.5373 & 8.8750 \\ 0.0000 & -0.0000 & -0.0000 & -0.0000 & 0.0000 & -0.3170 & -4.2093 \end{pmatrix}$$

$$B = \begin{pmatrix} -0.6504 & -8.5746 & -12.1400 & -4.1584 & -4.1342 & 1.2394 & 0.7316 \\ 0.0000 & 12.7088 & 0.4861 & -3.7474 & -3.3911 & -2.3644 & 2.0240 \\ -0.0000 & 0.0000 & 6.4543 & -4.2082 & 9.7642 & 6.2235 & -7.5972 \\ -0.0000 & -0.0000 & -0.0000 & 11.2391 & -1.3948 & 5.2406 & 6.1203 \\ 0.0000 & -0.0000 & -0.0000 & -0.0000 & 7.6207 & -3.5985 & 4.2659 \\ -0.0000 & 0.0000 & 0.0000 & 0.0000 & 0.0000 & 10.3066 & 1.7067 \\ 0.0000 & 0.0000 & 0.0000 & -0.0000 & -0.0000 & 0 & -14.9511 \end{pmatrix}$$

observar que ahora $A_{2,1} = 0.0001$ cumple la condición (5.1), entonces, obtenemos un autovalor real λ_1 que se calcula de la manera siguiente:

$$\lambda_1 = \frac{A_{11}}{B_{11}} = -12.9892$$

b) Ahora, procediendo por deflación:

$$A = \begin{pmatrix} 19.6255 & 2.4144 & -1.7300 & 1.3157 & -2.4753 & -3.7515 \\ 0.0335 & -1.0283 & -0.8193 & 7.1137 & 1.1939 & -3.4662 \\ -0.0000 & 1.4488 & -1.6798 & 0.3264 & 6.0254 & 1.1805 \\ -0.0000 & 0.0000 & -0.8018 & -1.3326 & -0.5242 & 5.7838 \\ -0.0000 & 0.0000 & -0.0000 & -0.3565 & 2.5373 & 8.8750 \\ -0.0000 & -0.0000 & -0.0000 & 0.0000 & -0.3170 & -4.2093 \end{pmatrix}$$

$$B = \begin{pmatrix} 12.7088 & 0.4861 & -3.7474 & -3.3911 & -2.3644 & 2.0240 \\ 0.0000 & 6.4543 & -4.2082 & 9.7642 & 6.2235 & -7.5972 \\ -0.0000 & -0.0000 & 11.2391 & -1.3948 & 5.2406 & 6.1203 \\ -0.0000 & -0.0000 & -0.0000 & 7.6207 & -3.5985 & 4.2659 \\ 0.0000 & 0.0000 & 0.0000 & 0.0000 & 10.3066 & 1.7067 \\ 0.0000 & 0.0000 & -0.0000 & -0.0000 & 0 & -14.9511 \end{pmatrix}$$

se procede como en el caso (a) anterior, obteniéndose sucesivamente los autovalores reales: $\lambda_2 = 1.5447$, $\lambda_3 = -0.4006$, $\lambda_4 = 0.1409$, luego entre las últimas iteraciones obtenemos las siguientes matrices:

$$A = \begin{pmatrix} -2.1974 & 0.7945 & 1.9506 & 4.2275 \\ -5.1060 & -0.1426 & 4.1356 & -4.5685 \\ 0 & 0.0000 & 4.3772 & 8.4168 \\ 0 & 0 & 0.0000 & -2.0293 \end{pmatrix}$$

$$B = \begin{pmatrix} -10.3184 & -3.8845 & 0.7940 & 1.6043 \\ -0.0000 & -7.1244 & 7.7365 & -2.7380 \\ 0.0000 & 0.0000 & 11.1923 & 2.9997 \\ -0.0000 & -0.0000 & 0 & -14.4048 \end{pmatrix}$$

- c) Se observa que no cumple la condición (a) es decir el elemento $A_{2,1} = -5.1060$ no cumple la condición (5.1) en este caso los valores propios son complejos entonces para ello aplicamos el programa "ST.m" obteniéndose los autovalores $\lambda_5 = -0,0184 + 0,2431i$, $\lambda_6 = -0,0184 - 0,2431i$.
- d) Finalmente procediendo por deflacion obtenemos:

$$A = \begin{pmatrix} 4.3772 & 8.4168 \\ 0.0000 & -2.0293 \end{pmatrix} \quad B = \begin{pmatrix} 11.1923 & 2.9997 \\ 0 & -14.4048 \end{pmatrix}$$

Observar ahora que $A_{2,1} = 0$ cumple la condición (5.1) entonces finalmente obtenemos un autovalor real:

$$\lambda_7 = \frac{A_{11}}{B_{11}} = 0.3911$$

Vectores propios Reales.

Una vez hallados los valores propios de el par (A,B) es posible hallar sus correspondientes vectores propios mediante el programa “autovectoresQZ.m” en las cuales son los siguientes:

$$\begin{aligned}
 V_1 &= \begin{pmatrix} 0.2322 \\ 0.6270 \\ -0.2534 \\ -0.3647 \\ 0.2249 \\ -0.0971 \\ -0.5438 \end{pmatrix} & V_2 &= \begin{pmatrix} -0.4337 \\ -0.2996 \\ 0.6434 \\ 0.3073 \\ -0.2632 \\ 0.0829 \\ 0.3710 \end{pmatrix} \\
 V_3 &= \begin{pmatrix} -0.3301 \\ -0.6440 \\ 0.3031 \\ 0.4803 \\ 0.1209 \\ 0.3003 \\ 0.2215 \end{pmatrix} & V_4 &= \begin{pmatrix} 0.4669 \\ 0.7735 \\ -0.2026 \\ -0.2540 \\ 0.1719 \\ 0.1892 \\ -0.1130 \end{pmatrix} & V_5 &= \begin{pmatrix} -0.2112 \\ -0.7321 \\ 0.4649 \\ 0.3636 \\ 0.2007 \\ 0.0105 \\ 0.1754 \end{pmatrix}
 \end{aligned}$$

Vectores propios complejos

$$\begin{aligned}
 V_6 &= \begin{pmatrix} 0.2133 + 0.0377i \\ 0.3962 + 0.2781i \\ -0.2562 - 0.2702i \\ -0.0966 - 0.1764i \\ -0.3213 - 0.2621i \\ -0.3216 - 0.2034i \\ -0.4703 - 0.0417i \end{pmatrix} & V_7 &= \begin{pmatrix} 0.0377 + 0.2133i \\ 0.2781 + 0.3962i \\ -0.2702 - 0.2562i \\ -0.1764 - 0.0966i \\ -0.2621 - 0.3213i \\ -0.2034 - 0.3216i \\ -0.0417 - 0.4703i \end{pmatrix}
 \end{aligned}$$

Además se puede verificar que los resultados cumple: $Ax = \lambda Bx$

5.2. El Problema de Valor Propio Generalizado Definido Simétrico

Muchas aplicaciones reales llevan a resolver problemas de valor propio generalizado definido simétrico de la forma:

$$Ax = \lambda Bx \quad \text{y} \quad ABx = \lambda x$$

donde A y B son simétricas, y al menos una de ellas es definida positiva. Asumiremos que B es definida positiva y consideraremos solamente el problema $Ax = \lambda Bx$.

Cuando B es definida positiva, podemos escribir:

$$ABx = \lambda x$$

en la forma

$$(A - \lambda B^{-1})(Bx) = 0,$$

como B^{-1} es también definida positiva, el problema $ABx = \lambda x$ es de la misma forma que el problema $Ax = \lambda Bx$.

El problema definido simétrico $Ax = \lambda Bx$ es comunmente resuelto en vibración de análisis de estructura.

Empezaremos con una importante propiedad de pencil definido simétrico.

Teorema 5.2.1. *El pencil definido simétrico $(A - \lambda B)$ tiene valores propios reales y vectores propios linealmente independientes.*

Prueba:

Desde que B es simétrico definido positivo, admite la descomposición de Cholesky:

$$B = LL^T$$

Luego:

$$Ax = \lambda Bx = \lambda LL^T x$$

el cual da:

$$L^{-1}A(L^T)^{-1}L^T x = \lambda L^T x \quad \text{o} \quad Cy = \lambda y \quad \text{donde} \quad y = L^T x$$

donde la matriz $C = L^{-1}A(L^T)^{-1}$ es simétrica, además, λ es real. La afirmación acerca de los vectores propios es obvia, debido a que una matriz simétrica tiene un conjunto de n vectores propios independientes según el teorema (2.0.8).

5.2.1. Ortogonalidad de los Vectores Propios

Si los vectores propios y_1, y_2, \dots, y_n de C son elegidos ortonormales, entonces es fácil ver que los vectores propios x_i son B -ortonormales, esto es:

$$\begin{aligned} x_i^T B x_i &= 1 & \text{si } i = 1, \dots, n \\ x_i^T B x_j &= 0 & \text{si } i \neq j \end{aligned}$$

Más aún, si x_i es un vector propio normalizado, $x_i^T A x_i = c_{ii}$, $i = 1, \dots, n$ y:

$$x_i^T A x_j = 0 \quad i \neq j$$

para probar

$$x_i^T B x_j = x_i^T L L^T x_j = (L^T x_i)^T L^T x_j = y_i^T y_j = 0$$

(desde que y_1, \dots, y_n son ortogonales). Las otras relaciones pueden ser probadas similarmente.

5.3. El Método QZ para el Pencil

Definida Simétrica

El algoritmo QZ descrito en la sección previa para pencil $A - \lambda B$ regular también puede ser aplicado a pencil definida simétrica. Sin embargo la desventaja es que tanto la simetría como lo de definida positiva se perdería, para ello describimos el siguiente algoritmo.

5.3.1. El Algoritmo Cholesky QR

La demostración del teorema (5.2.1) nos da previamente un método para encontrar los valores propios generalizados de una pencil definida positiva simétrica.

Antes de ver este método veamos previamente los siguientes algoritmos:

Para el Cálculo de Valores Propios

Cuando A es simétrica, hacemos la transformación $PAP^t = H$ Hessenberg que es simétrica, además se le puede llevar a una matriz simétrica tridiagonal.

Para aplicar el método de iteración QR , a una matriz inicial es una matriz simétrica a tridiagonal T , entonces cada matriz:

$$T_k - w_k I = Q_k R_k$$

Para hallar w_k (conocido como el **cambio Wilkinson**) se utiliza la submatriz de T_k inferior que es dada por:

$$\begin{pmatrix} t_{n-1,n-1}^{(k)} & t_{n,n-1}^{(k)} \\ t_{n,n-1}^{(k)} & t_{n,n}^{(k)} \end{pmatrix}$$
$$\Rightarrow w_k = t_{nn}^{(k)} + r - \text{sign}(r) \sqrt{r^2 + (t_{n,n-1}^{(k)})^2}$$

donde:

$$r = (t_{n-1,n-1}^{(k)} - t_{nn}^{(k)})/2$$

En efecto, el primer algoritmo está dado por:

ALGORITMO 5.3.1.

Paso 1: Transformar a una matriz simétrica tridiagonal T usando transformaciones similares ortogonales:

$$PAP^t = T$$

Paso 2: Aplicar desplazamiento simple en la iteración QR a T con el cambio de Wilkinson w .

Para $T_1 = T$ calcular w_1 .

Para $k = 1, 2, \dots$ hacer hasta que converja.

$$T_k - w_k I = Q_k R_k$$

$$T_{k+1} = R_k Q_k + w_k I$$

Escoger el nuevo w_k

Ahora veamos un segundo algoritmo el cual consiste en hallar los vectores propios correspondientes a los valores propios hallados en el algoritmo (5.3.1).

ALGORITMO 5.3.2.

Paso 1: Reducir la matriz A a una matriz Hessenberg superior H :

$$P^t AP = H$$

Paso 2: Calcular un valor propio λ

Paso 3: Aplicar la iteración inversa

Para $k = 1, 2, \dots$ hacer hasta que converja.

Resolver:

$$(H - \lambda I)z^{(k)} = y^{(k-1)}$$

$$y^{(k)} = \frac{z^{(k)}}{\max(z^{(k)})}$$

Parar si $\| (H - \lambda I)y^{(k)} \| \leq cu \| A \|_\infty$

donde c es una constante de orden 1 y

$$\max(\{x\}) = \max_{1 \leq i \leq n} (\{|x_i|\}) \text{ con } x = (x_1, x_2, \dots, x_n)^t \in \mathbb{R}(n, n)^n$$

Paso 4: Hallar el vector propio x

$$x = Py^{(k)}$$

donde $y^{(k)}$ es la aproximación del vector propio y obtenida en el paso 3.

Ahora veamos el algoritmo de Cholesky QR para una pencil simétrica definida. Este algoritmo calcula los valores propios λ_i y los correspondientes vectores propios x_i , ($i = 1, 2, \dots, n$) de la pencil simétrica definida (A, B) .

ALGORITMO 5.3.3 (Algoritmo de Cholesky).

Paso 1: Encontrar la factorización Cholesky B .

$$B = LL^t$$

Paso 2: Formar $C = L^{-1}A(L^t)^{-1}$ tomando ventaja de la simetría de A .

Paso 3: Calcular los valores propios λ_i , $i = 1, 2, \dots, n$ y los vectores propios ortonormales y_i , $i = 1, 2, \dots, n$ de la matriz simétrica C , usando el algoritmo de iteración simétrica QR , (5.3.1), y el algoritmo de iteración Hessemberg inversa, (5.3.2), ($\lambda_1, \lambda_2, \dots, \lambda_n$ son los valores propios del par (A, B))

Paso 4: Calcular los vectores propios x_i correspondientes a los valores propios λ_i de la pencil (A, B) resolviendo $L^t x_i = y_i$, $i = 1, \dots, n$.

Capítulo 6

Algunas Aplicaciones

Casi todos los problemas de valor propio que surgen en las vibraciones de estructuras son problemas de valor propio generalizados definido simétrico.

Veamos los siguientes casos:

1.- Vibración de un Sistema Masa - Resorte:

Consideremos un sistema donde las masas m_1 , m_2 y m_3 y las constantes de rigidez k_1 , k_2 y k_3 como se muestra en la figura (6.1), (Las matrices rígidas muy a menudo son determinados usando las características físicas de los sistemas, los valores y vectores propios son usados para estudiar el comportamiento de dicho sistema).

Estudiaremos la configuración del sistema correspondiente a los diferentes modos de vibración.

Para ello seguiremos los siguientes pasos:

i) Formularemos el problema como un problema de valor propio generalizado definido simétrico. Sean los desplazamientos de las masas desde su posición de equilibrio estático definido por las coordenadas generalizadas y_1, y_2, y_3 respectivamente. las ecuaciones de movimiento para el sistema puede ser escrito como:

$$\begin{aligned}m_1\ddot{y}_1 + (k_1 + k_2)y_1 - k_2y_2 &= 0 \\m_2\ddot{y}_2 - k_2y_1 + (k_2 + k_3)y_2 - k_3y_3 &= 0 \\m_3\ddot{y}_3 - k_3y_2 + k_3y_3 &= 0\end{aligned}\tag{6.1}$$

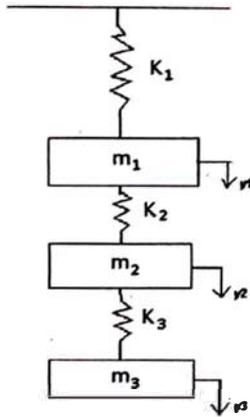


Figura 6.1: Sistema Masa - Resorte

este sistema es equivalente a:

$$\begin{pmatrix} m_1 & 0 & 0 \\ 0 & m_2 & 0 \\ 0 & 0 & m_3 \end{pmatrix} \begin{pmatrix} \ddot{y}_1 \\ \ddot{y}_2 \\ \ddot{y}_3 \end{pmatrix} + \begin{pmatrix} k_1 + k_2 & -k_2 & 0 \\ -k_2 & k_2 + k_3 & -k_3 \\ 0 & -k_3 & k_3 \end{pmatrix} \begin{pmatrix} y_1 \\ y_2 \\ y_3 \end{pmatrix} = \begin{pmatrix} 0 \\ 0 \\ 0 \end{pmatrix}$$

es decir:

$$M\ddot{y} + Ky = 0$$

donde: $M = \text{diag}(m_1, m_2, m_3)$ y

$$K = \begin{pmatrix} k_1 + k_2 & -k_2 & 0 \\ -k_2 & k_2 + k_3 & -k_3 \\ 0 & -k_3 & k_3 \end{pmatrix}$$

Asumiendo movimiento armónico podemos escribir:

$$y_1 = x_1 e^{i\omega t}$$

$$y_2 = x_2 e^{i\omega t}$$

$$y_3 = x_3 e^{i\omega t}$$

donde x_1, x_2 y x_3 son respectivamente las amplitudes de las masas m_1, m_2, m_3 y ω es la frecuencia natural.

Sustituyendo y_1, y_2, y_3 en la ecuación (6.1) y además:

$$\ddot{y}_k = -\omega^2 x_k e^{i\omega t}, \quad k = 1, 2, 3$$

tenemos:

$$\begin{aligned} -m_1 x_1 \omega^2 + (k_1 + k_2)x_1 - k_2 x_2 &= 0 \\ -m_2 x_2 \omega^2 - k_2 x_1 + (k_2 + k_3)x_2 - k_3 x_3 &= 0 \\ -m_3 x_3 \omega^2 - k_3 x_2 + k_3 x_3 &= 0 \end{aligned}$$

y llevándolo a su forma matricial queda:

$$\begin{pmatrix} k_1 + k_2 & -k_2 & 0 \\ -k_2 & k_2 + k_3 & -k_3 \\ 0 & -k_3 & k_3 \end{pmatrix} \begin{pmatrix} x_1 \\ x_2 \\ x_3 \end{pmatrix} = \omega^2 \begin{pmatrix} m_1 & 0 & 0 \\ 0 & m_2 & 0 \\ 0 & 0 & m_3 \end{pmatrix} \begin{pmatrix} x_1 \\ x_2 \\ x_3 \end{pmatrix}$$

denotemos por $KX = \lambda MX$, $\lambda = \omega^2$.

Los valores propios $\lambda_i = \omega_i^2$, $i = 1, 2, 3$ son los cuadrados de las frecuencias naturales de la primera, segunda y tercer modo de vibración respectivamente.

ii) Resolveremos el problema del valor propio generalizado usando el algoritmo de Cholesky QR .

Sea $m_1 = 20000$, $m_2 = 30000$, $m_3 = 40000$ y $k_1 = k_2 = k_3 = 1.5 \times 10^9$.

Entonces tenemos que:

$$\begin{aligned} M &= \text{diag}(20000, 30000, 40000) \\ K &= 10^9 \begin{pmatrix} 3 & -1.5 & 0 \\ -1.5 & 3 & -1.5 \\ 0 & -1.5 & 1.5 \end{pmatrix} \\ L &= \begin{pmatrix} 141.4214 & 0 & 0 \\ 0 & 173.2051 & 0 \\ 0 & 0 & 200.00 \end{pmatrix} \\ C &= 10^5 \begin{pmatrix} 1.5 & -0.6124 & 0 \\ -0.6124 & 1 & -0.4330 \\ 0 & -0.4330 & 0.3750 \end{pmatrix} \end{aligned}$$

Los valores propios de (K, M) son los mismos que los valores propios de C y estos son:

$$10^5(0.0860, 0.8382, 1.9508)$$

y la frecuencia natural ($\lambda_i = \omega^2$) son:

$$10^2(0.9273, 2.8951, 4.4168)$$

los vectores propios son:

$$\begin{pmatrix} 0.0020 \\ 0.0039 \\ 0.0050 \end{pmatrix}, \begin{pmatrix} 0.0071 \\ 0.0062 \\ -0.0051 \end{pmatrix}, \begin{pmatrix} 0.0071 \\ -0.0043 \\ 0.0010 \end{pmatrix}$$

Estos vectores propios pueden ser usados para determinar diferentes configuraciones del sistema para diferentes modos, es decir, se puede usar para ver como las estructuras vibran cuando es excitada a una frecuencia natural dada.

2.-Vibración de un Edificio:

Consideremos un edificio de hormigón armado de 4 pisos como se muestra en la figura. Los pisos y la azotea, que son bastante rígido, están representadas por bloques de masas m_1 a m_4 teniendo un movimiento horizontal causado por la deformación de columnas debido al movimiento y de k_1 a k_4 son constantes de elongación de los resortes que funcionan como resortes en paralelo.

Estudiaremos la configuración cuando el edificio vibra en sus dos primeros modos (correspondientes a los dos valores propios más pequeños).

a) Formulamos el problema como un problema de valores propios generalizados simétrico definido en términos de masa y matrices rígidas:

$$KX = \lambda MX$$

Sea

$$M = \text{diag}(m_1, m_2, m_3, m_4)$$

$$K = \begin{pmatrix} k_1 + k_2 & -k_2 & 0 & 0 \\ -k_2 & k_2 + k_3 & -k_3 & 0 \\ 0 & -k_3 & k_3 + k_4 & -k_4 \\ 0 & 0 & -k_4 & k_4 \end{pmatrix}$$

tomemos:

$$m_1 = 6 \times 10^7, m_2 = 4 \times 10^7, m_3 = 3 \times 10^7, m_4 = 2 \times 10^7$$

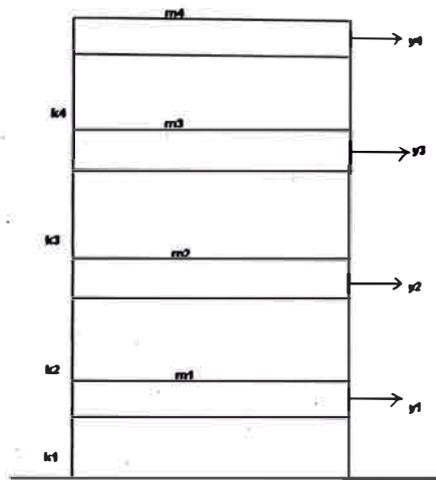


Figura 6.2: Edificio de 4 Pisos

$$k_1 = 10 \times 10^{14}, \quad k_2 = 8 \times 10^{14}, \quad k_3 = 6 \times 10^{14}, \quad k_4 = 4 \times 10^{14}$$

entonces tenemos:

$$K = 10^{14} \begin{pmatrix} -1.8 & -8 & 0 & 0 \\ -8 & 14 & -6 & 0 \\ 0 & -6 & 10 & -4 \\ 0 & 0 & -4 & 4 \end{pmatrix}$$

y

$$M = 10^7 \text{diag}(6, 4, 3, 2)$$

b) Encontramos los valores propios y vectores propios usando el algoritmo de Cholesky QR los valores propios de C en la cual son los valores propios de la pencil (K, M) son:

$$10^7 \begin{pmatrix} 0.7112 \\ 3.1213 \\ 5.7283 \\ -1.0275 \end{pmatrix}$$

los vectores propios aproximados de (K, M) correspondientes a sus valores propios son

$$10^{-3} \begin{pmatrix} -0.0525 \\ 0.0398 \\ 0.1441 \\ 0.2236 \end{pmatrix}, \quad 10^{-3} \begin{pmatrix} 0.0616 \\ -0.1581 \\ -0.1221 \\ 0.2178 \end{pmatrix}$$

$$10^{-3} \begin{pmatrix} -0.0339 \\ 0.1533 \\ -0.1826 \\ 0.0979 \end{pmatrix}, \quad 10^{-3} \begin{pmatrix} 0.1291 \\ 0.0704 \\ 0.0405 \\ 0.0267 \end{pmatrix}$$

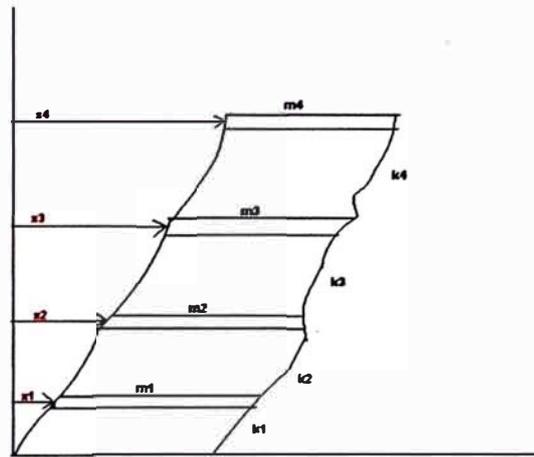


Figura 6.3: Primer Modo de Vibración de un Edificio de 4 Pisos

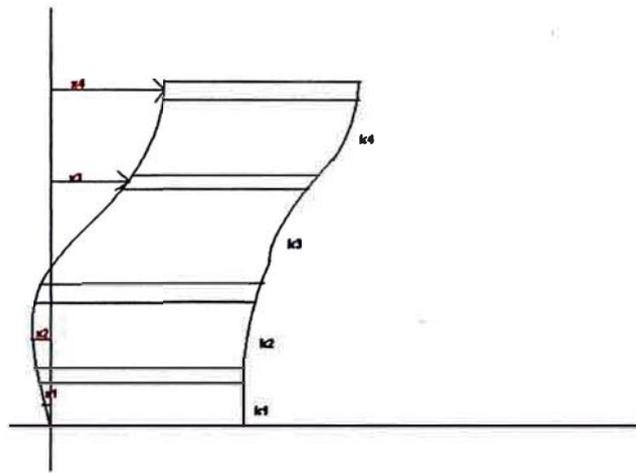


Figura 6.4: Segundo Modo de Vibración de un Edificio de 4 Pisos

Capítulo 7

Conclusiones de la Tesis

Las conclusiones de la presente tesis están en orden de desarrollo:

- El algoritmo QZ nos permite encontrar los valores propios de la matriz $\text{pencil}(A,B)$ y el método de iteración inversa nos permite calcular su correspondiente vector propio.
- En el algoritmo QZ hemos observado que para el cálculo de los valores propios complejos, sólo es necesario calcular los valores de la submatriz de orden 2×2 utilizando del método de deflación evitando la aritmética compleja.
- El algoritmo QZ evita calcular la inversa de la matriz B ($Ax = \lambda Bx$), si es no-singular, aplicando el algoritmo QR para AB^{-1} .
- Los cambios simples y dobles que se utilizan en este algoritmo es recomendado para la elección de la submatriz 2×2 $C = AB^{-1}$, en cada iteración sin calcular la inversa de B .
- Si los valores propios de (A, B) son reales, la subdiagonal de la matriz en cada iteración converge más rápido a cero.
- Cuando una de las matrices es definida simétrica, el algoritmo QZ se simplifica en particular a un método para hallar los valores propios con sus respectivos vectores propios, llamado el método de Cholesky.

- En este trabajo también se muestra como el problema $Ax = \lambda Bx$ ayuda a resolver diferentes fenómenos que pueden suceder en la ingeniería estructural y vibración.

Apéndice A

Programas elaborados

Damos a conocer la programación de algunos de los algoritmos dados a lo largo de este texto, esta programación ha sido realizada en el lenguaje de programación Matlab.

A.1. Total

Este programa elabora el algoritmo qz que consiste resolver el problema $Ax=\lambda Bx$

```
B = [8  5  3  5  4  8  5 ;  
     -5  1 -3 -4  3  5  4 ;  
      2 -7 -6  5  5 -2 -5 ;  
      1 -2  9 -3 -8  7 -9 ;  
     -7  9  4  5 -3 -8  2 ;  
      5 -4  7 -3  9 -2  1 ;  
      2 -6  2 -4 -1  4 -5 ]
```

```
A = [1  5  3  5  4 -2 -1 ;  
     -5  1  3 -4  3  2 -6 ;  
      2 -7 -6 -5  5 -3 -3 ;  
      1  2  9 -3 -8  1  2 ;
```

```

-7  9  6  5 -3 -2  5 ;
-5  2  6 -3  4 -2 -7 ;
-6  2  1 -1  5 -4 -3 ]

AA = A;
BB = B;
CC = A*inv(B);

[n,n] = size(B);
[Q,R] = paso1(n,B);
B = Q'*B;
A = Q'*A;
[A,B] = paso2(n,A,B); %Transforma a A en una matriz Hesenberg y B en una
%matriz triangular
subm = 0
m = n;

ar = 0; %Contabiliza los autovalores reales
vr = [];
ac = 0; %Contabiliza los autovalores complejos
vc = [];
b = zeros(2,2);
while subm<(m-2)
    %subm = subm + 1;
    AH = A;
    BT = B;
    exito = 0;

```

```

tol = 1e-5;
Nmax = 200;
iter = 0;
while (exito==0 & iter<Nmax)
    iter = iter + 1;
    i=2;
    if abs(A(i,i-1))<tol*(abs(A(i,i)) + abs(A(i-1,i-1)))
        exito = 1;
    else
        %Aqui se calcula la matriz inversa de B_2x2
        B
        b_matriz = [B(1,1) B(1,2);0 B(2,2)]
        dB = B(1,1)*B(2,2)
        b(1,1) = B(2,2)/dB;          b(1,2) = -B(1,2)/dB;
        b(2,1) = 0/dB;              b(2,2) = B(1,1)/dB;
        b
        %Aqui calculamos las 2 primeras columnas de
        %N = (C-alpha_1*I)(C-alpha_2*I)
        %donde C = A*B^-1
        C = [A(:,1) A(:,2)]*b;
        %Calculamos las tres entradas NO NULAS de N = [x y z 0 0 ... 0]'
        %Para esto debemos de ingresar valores adecuados de alpha_1
        %y alpha_2
        %%%%%%%%%%%%%%
        cB = B(n-2:n,n-2:n);
        cB = minversa(3,cB);
        cA = A(n-2:n,n-2:n);

```

```

cC = cA*cB;
det_C = cC(1,1)*cC(2,2) - cC(1,2)*cC(2,1);
tr_C = cC(1,1) + cC(2,2);

alpha_1 = (tr_C + sqrt(tr_C^2 - 4*det_C))/2;
alpha_2 = (tr_C - sqrt(tr_C^2 - 4*det_C))/2;
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%Luego:
N = zeros(n,1);
%x
N(1) = (C(1,1) - alpha_1)*(C(1,1) - alpha_2) + C(1,2)*C(2,1);
%y
N(2) = C(2,1)*(C(1,1) - alpha_2) + C(2,1)*(C(2,2) - alpha_1);
%z
N(3) = C(2,1)*C(3,2);
N;
[u,Q1] = paso3(n,N);
A = Q1*A;
B = Q1*B;

j = 3;
i = 2;
[B,c,s] = PBZ(n,i,j,B);
[A] = BZ(n,i,j,c,s,A);

j = 3;
i = 1;

```

```

    [B,c,s] = PBZ(n,i,j,B);
    [A] = BZ(n,i,j,c,s,A);

    j = 2;
    i = 1;
    [B,c,s] = PBZ(n,i,j,B);
    [A] = BZ(n,i,j,c,s,A);

    A;
    B;

    [A1,B1,Q,Z] = nuevo9(n,A,B);
    A = A1;
    B = B1;
end

end

A;
B;
if exito ==1
    A;
    B;
    ar = ar + 1;
    vr = [vr;ar A(1,1)/B(1,1)];
    subm = subm + 1;
    A = A(2:n,2:n);
    B = B(2:n,2:n);
    n = n - 1;

```

```

else
    A;
    B;
    subm = subm + 2;
    S = A(1:2,1:2);
    T = B(1:2,1:2);
    lambda = ST(S,T);
    vc = [vc;ac+1 lambda(1); ac+2 lambda(2)];
    %subm = subm + 1;
    ac = ac + 2;
    A = A(3:n,3:n);
    B = B(3:n,3:n);
    n = n - 2;
end
A;
B;
end

i = 2;
if abs(A(i,i-1))<tol*(abs(A(i,i)) + abs(A(i-1,i-1)))
    exito = 1;
end

if exito ==1
    vr = [vr;ar+1 A(1,1)/B(1,1); ar+2 A(2,2)/B(2,2)];

```

```

    ar = ar + 2;
else
    S = A(1:2,1:2);
    T = B(1:2,1:2);
    lambda = ST(S,T);
    ac = ac + 2;
    vc = [vc;ac+1 lambda(1); ac+2 lambda(2)];
end

%Autovalores calculados con Matlab
vectores = eig(CC)

%Autovalores calculados con el algoritmo QZ implementado
if ar==0
    fprintf('La pencil (A,B) no tiene autovalores reales')
else
    autovalores_reales = vr(:,2)

    fprintf('Autovalores Reales de la pencil (A,B)')
    autovalores_reales = vr(:,2)
    fprintf('Autovectores Reales')
    V = autovectores_QZ(m,AA,BB,ar,autovalores_reales,1)

    A = AA;
    B = BB;
    lambda = autovalores_reales;
    X = V;

```

```

n = ar;

Ax = [];
LBx = [];
for i=1:n
    Ax = [Ax A*X(:,i)];
    LBx = [LBx lambda(i)*B*X(:,i)];
end
fprintf('Comprobar los resultados por columna de autovalores reales\n')
Ax
LBx
end

if ac==0
    fprintf('La pencil (A,B) no tiene autovalores complejos')
else
    autovalores_complejos = vc(:,2)
    fprintf('Autovalores Complejos de la pencil (A,B)')
    autovalores_complejos
    fprintf('Autovectores complejos')
    V = autovectores_QZ(m,AA,BB,ac,autovalores_complejos,2)

    A = AA;
    B = BB;
    lambda = autovalores_complejos;
    X = V;
    n = ac;

```

```

Ax = [];
LBx = [];

for i=1:n
    Ax = [Ax A*X(:,i)];
    LBx = [LBx lambda(i)*B*X(:,i)];
end
fprintf('Comprobar los resultados por columna de autovalores complejos\n')
Ax
LBx
end

```

A.2. Paso1

```

function [Q,R] = paso1(n,B)
%Esta funcion realiza la factorizacion B = QR de una matriz cuadrada B
%B = [1 5 3 5 4;-5 1 3 -4 3 ;2 -7 -6 -5 5 ; 1 2 9 -3 -8;-7 9 6 5 -3]

C = B;

%n,n] = size(B)

Q = eye(n,n);
for i=1:(n-1)
    Qi = eye(n,n);
    for j=(i+1):n
        [c,s,B]=mgivens(n,i,j,B);
        Qi = pgivens(n,c,s,i,j,Qi);
    end
end

```

```

    end
    Qi;
    Q = Q*Qi';
end
R = B;
Q;

```

A.3. Mgivens

```

function [c,s,J]=mgivens(n,i,j,B)
%Obtencií>½n de los parí>½metros de Givens y el producto de J*B
if abs(B(i,i))>=abs(B(j,i))
    t = B(j,i)/B(i,i);
    c = 1/sqrt(1 + t^2);
    s = c*t;
else
    t = B(i,i)/B(j,i);
    s = 1/sqrt(1 + t^2);
    c = s*t;
end

for k=1:n
    a = B(i,k);
    b = B(j,k);
    B(i,k) = c*a + s*b;
    B(j,k) = -s*a + c*b;
end

```

```
J = B;
```

A.4. Pgivens

```
function [J] = pgivens(n,c,s,i,j,B);
```

```
%Halla el producto de J*B
```

```
for k=1:n
```

```
    a = B(i,k);
```

```
    b = B(j,k);
```

```
    B(i,k) = c*a + s*b;
```

```
    B(j,k) = -s*a + c*b;
```

```
end
```

```
J = B;
```

A.5. Paso2

```
function [H,R] = paso2(n,A,B)
```

```
% paso2 del algoritmo QZ
```

```
%A = [1 5 3 5 4;-5 1 3 -4 3 ;2 -7 -6 -5 5 ; 1 2 9 -3 -8;-7 9 6 5 -3]
```

```
%B = [1 -5 3 2 5 ;0 4 6 -4 8;0 0 -7 -3 5;0 0 0 -8 -1;0 0 0 0 -7 ]
```

```
%[n,n] = size(A);
```

```
for i=1:(n-2)
```

```
    %i=1
```

```
    i;
```

```
    p = n;
```

```

for j=n:-1:(i+2)
    p = p -1;
    j;
    [c,s,A] = PQA(n,i,p,j,A);
    B = QA(n,p,j,c,s,B);
    [B,c,s] = PBZ(n,p,j,B);
    A = BZ(n,p,j,c,s,A);
end
end

```

```

%Matriz Hesember Superior H

```

```

H = A;

```

```

%Matriz Triangular Superior R

```

```

R = B;

```

A.6. PQA

```

function [c,s,Q] = PQA(n,i,p,j,A)

```

```

% n: orden de la matriz cuadrada A

```

```

% i: columna del elemento que se desea hacer NULO

```

```

% p: parametros de la matriz de Givens J(p,j,theta)

```

```

% Para hacer la posicion A(j,i) igual a cero

```

```

%if (p<j)

```

```

    if abs(A(p,i))>=abs(A(j,i))

```

```

    t = A(j,i)/A(p,i);
    c = 1/sqrt(1 + t^2);
    s = c*t;
else
    t = A(p,i)/A(j,i);
    s = 1/sqrt(1 + t^2);
    c = s*t;
end

for k=1:n
    a = A(p,k);
    b = A(j,k);
    A(p,k) = c*a + s*b;
    A(j,k) = -s*a + c*b;
end
Q = A;
%end

```

A.7. QA

```

function [Q] = QA(n,p,j,c,s,A)
%Halla el producto de Q*A
for k=1:n
    a = A(p,k);
    b = A(j,k);
    A(p,k) = c*a + s*b;
    A(j,k) = -s*a + c*b;
end

```

```
Q = A;
```

A.8. PBZ

```
function [Z,c,s] = PBZ(n,p,j,B)
```

```
%Para hacer cero la posición  $i > \frac{1}{2}n$  B(j,i) igual a cero
```

```
b1 = B(j,j);
```

```
b2 = B(j,p);
```

```
if abs(b1) >= abs(b2)
```

```
    t = b2/b1;
```

```
    c = 1/sqrt(1 + t^2);
```

```
    s = c*t;
```

```
else
```

```
    t = b1/b2;
```

```
    s = 1/sqrt(1 + t^2);
```

```
    c = s*t;
```

```
end
```

```
for k=1:n
```

```
    a = B(k,p);
```

```
    b = B(k,j);
```

```
    B(k,p) = c*a - s*b;
```

```
    B(k,j) = s*a + c*b;
```

```
end
```

```
Z = B;
```

A.9. BZ

```
function [Z] = BZ(n,i,j,c,s,B)
```

```
%Elabora el producto de una matriz B por una matriz Givens por la derecha
```

```
if i>j
```

```
    aux = j;
```

```
    j = i;
```

```
    i = aux;
```

```
end
```

```
for k=1:n
```

```
    a = B(k,i);
```

```
    b = B(k,j);
```

```
    B(k,i) = c*a - s*b;
```

```
    B(k,j) = s*a + c*b;
```

```
end
```

```
Z = B;
```

A.10. Minversa

```
function B = minversa(n,A)
```

```
%Este programa calcula la inversa de una matriz triangular superior " A "
```

```
%A = [-2  3  4  5  6  2;
```

```

%      0 -2 4  5  3  6;
%      0  0 2 -5 -2  1;
%      0  0 0 -5  6  3;
%      0  0 0  0 -3  2;
%      0  0 0  0  0 -3];
%[n,m] = size(A);

B = eye(n,n);

C = A;

for i=1:n
    B(i,i) = B(i,i)/A(i,i);
    A(i,:) = A(i,+)/A(i,i);
end

for i=1:(n-1)
    for k=(i+1):n
        for j=n:-1:k
            B(i,j) = B(i,j) - A(i,k)*B(k,j);
            A(i,j) = A(i,j) - A(i,k)*A(k,j);
        end
    end
end

C;
B;

```

A.11. Paso3

```
function [u,H] = paso3(n,x);
%Este paso permitira encontrar una matriz Housholder
%tal que Hx sea un multiplo de e_1.
%n : longitud del vector x
if x(1)>0
    signo = 1;
else
    if x(1)<0
        signo =-1;
    else
        signo = 0;
    end
end
norma2 = 0;
for i=1:n
    norma2 = norma2 + x(i)^2;
end
norma2 = sqrt(norma2);

u = x + signo*norma2*eye(n,1);

H = eye(n,n) - 2*u*u'/(u'*u);
```

A.12. Nuevo9

```
function [A1,B1,Q,Z] = nuevo9(n,A,B)
```

```

%Obtiene una matriz Hessemberg A manteniendo la estructura triangular de B
%n = 5;
%A = [1 2 -5 3 6;-6 -2 1 3 5;-7 6 3 -2 1;5 -2 -4 6 2;-4 -6 7 2 -1]
%B = [2 5 3 6 2;0 1 2 -5 6;0 0 -3 5 2;0 0 0 -5 3;0 0 0 0 2]
Q = eye(n,n);
Z = eye(n,n);
e = eye(n,n);
for j=1:(n-2)
    %j = 1
        for i=(n-1):-1:(j+1)
            %i = n-1
                ej = e(:,j); %Vector Columna
                x = A*ej; %Vector Columna
                [c,s,G]= xgivens(n,i,i+1,x);
                [A] = pgivens(n,c,s,i,i+1,A);
                [B] = pgivens(n,c,s,i,i+1,B);
                Q = pgivens(n,c,s,i,i+1,Q');

                ei = e(i+1,:); % Vector Fila
                x = ei*B; % Vector Fila
                [c,s,G]= ygivens(n,i+1,i,x);
                [A] = BZ(n,i+1,i,c,s,A);
                [B] = BZ(n,i+1,i,c,s,B);
                [Z] = BZ(n,i+1,i,c,s,Z');
            end
        end
    end
end

```

```
A1 = A;
```

```
B1 = B;
```

A.13. Xgivens

```
function [c,s,Jx]=xgivens(n,i,j,x)
%x   vector columna
%n   Numero de elementos del vector
%i,j: filas del vector (i: menor valor, j: mayor valor)
%Esta funcion permite hallar una matriz J(i,j) tal que la
%fila "j" del vector columna J*x = 0
%SALIDA: parametros c y s de la matriz de givens J(i,j)
if i>j
    aux = j;
    j = i;
    i = aux;
end
if abs(x(i))>=abs(x(j))
    t = x(j)/x(i);
    c = 1/sqrt(1+t^2);
    s = c*t;
else
    t = x(i)/x(j);
    s = 1/sqrt(1+t^2);
    c = s*t;
end
Jx = x;
a = Jx(i);
```

```

b = Jx(j);
Jx(i) = c*a + s*b;
Jx(j) = -s*a + c*b;

```

A.14. Ygivens

```

function [c,s,Jy]=ygivens(n,i,j,y)
%y : vector fila
%n : Numero de elementos del vector
%i,j: Columnas del vector (i: menor valor, j: mayor valor)
%Esta funcion permite hallar una matriz J(i,j) tal que la
%columna " i " del vector fila x*J = 0
%SALIDA: parametros c y s de la matriz de givens J(i,j)
if i>j
    aux = j;
    j= i;
    i = aux;
end

if abs(y(i))>=abs(y(j))
    t = y(j)/y(i);
    s = 1/sqrt(1 + t^2);
    c = s*t;
else
    t = y(i)/y(j);
    c = 1/sqrt(1 + t^2);
    s = c*t;
end

```

```

Jy = y;
a = Jy(i);
b = Jy(j);
Jy(i) = c*a - s*b;
Jy(j) = s*a + c*b;

```

A.15. ST

```

function lambda = ST(A,B)
%A,B: Matrices cuadradas de orden 2x2
%Resuelve  $\det(A - \lambda*B) = 0$  donde:
%A = [a11 a12;
%     a21 a22];
%B = [b11 b12;
%     0   b22];
lambda = zeros(2,1);
a = B(1,1)*B(2,2);
b = A(2,1)*B(1,2) - A(1,1)*B(2,2) - A(2,2)*B(1,1);
c = A(1,1)*A(2,2) - A(2,1)*A(1,2);

D = b^2-4*a*c;

lambda = [(-b+sqrt(D))/(2*a); (-b-sqrt(D))/(2*a)]

```

A.16. Vectores Propios-QZ

```
function V = autovectores_QZ(n,A,B,m,lambda,opcion)
%n : orden de las matrices A,B
%m : Numero de autovalores reales de la pencil (A,B)
%opcion = 1 para autovalores reales
%opcion = 2 para autovalores complejos
V = [];
tol = 1e-3;
Nmax = 100;

if opcion==1
    v0 = ones(n,1);
else
    v0 = complex(ones(n,1),ones(n,1));
end

for i=1:m
    L = lambda(i);
    exito = 0;
    k = 0;
    %vk = ones(n,1);
    vk = v0;
    Ai = A - L*B;
    while (exito==0 & k<=Nmax)
        k = k + 1;
        %vkr = resolver(Ak,B*vk)';
        vkr = gausjordan(n,[Ai B*vk]); %Vector Columna
```

```

        vk = vkr/norm(vkr,2);
        if norm(Ai*vkr,inf)<norm(A,inf)
            exito = 1;
        end
    end
    V = [V vk];
end

```

A.17. Gauss-Jordan

```

function x = gausjordan(n,a)
%Orden de la matriz A del sistema Ax=b
%a : matriz aumentada del sistema Ax=b, es decir: a = [A b]
tam = size(a);
fil=tam(1,1);
col=tam(1,2);
x = zeros(n,1); %x: vector columna como respuesta
%y=a
%Matriz diagonal aumentada
for k = 1:fil
    %pivoteo
    may = abs(a(k,k));
    pos = k;
    y=a;
    for i=k+1:fil
        if may < abs(a(i,k))
            may=abs(a(i,k));
            aux=a(i,:);

```

```

    a(i,:)=a(k,:);
    a(k,:)=aux;
    end
end
y=a;
%fin de pivoteo
for i = 1:fil
    if i ~= k
        pivote=a(i,k)/a(k,k);
        for j = k:col
            a(i,j) = a(i,j) - pivote*a(k,j);
        end
    end
    y=a;
end
end
%Matriz identidad aumentada
for i = 1:fil
    a(i,col)= a(i,col)/a(i,i);
    a(i,i)=1;
end
y=a;
%Valores de x(i)
for i=1:fil
    x(i)=a(i,col);
end

```

A.18. CholeskyQR

```
%Se implementa el metodo de Cholesky para el calculo de  
%autovalores y autovectores de la pencil simetrica (A,B)
```

```
fprintf('Matrices Ingresadas\n')
```

```
B = [ 36   6  -54   24   -6  -42  -48;  
      6   5  -25   -2    7   11  -18;  
     -54 -25  181    0   13    3  130;  
      24  -2    0   30   -9  -52  -13;  
      -6   7   13   -9  142   28   56;  
     -42  11    3  -52   28  187  -37;  
     -48 -18  130  -13   56  -37  164]
```

```
A = [ 4   -2  -10   -8   -2    2    2;  
      -2    5   17   -2    5    3    9;  
     -10   17   70   -4   -1   16   52;  
      -8   -2   -4   78   38   12   12;  
      -2    5   -1   38  121  -23   15;  
       2    3   16   12  -23   43   66;  
       2    9   52   12   15   66  217]
```

```
n = 7;
```

```
L = cholesky(n,B); % Matriz triangular inferior
```

```
%minversa(n,U): funcion que permite calcular la inversa de una matriz  
% triangular superior " U " de orden " n ".
```

```
inversa_L = minversa(n,L')'; % Inversa de L (L' es triangular superior)
```

```

inversa_Lt = minversa(n,L'); % Inversa de L' (Matriz Transpuesta de L)

C = inversa_L*A*inversa_Lt;

%[P,T] = algo544(n,C): Halla la matriz Hesemberg de una matriz " C "
%      de orden " n ". Como " C " es simetrica entonces T es TRIDIAGONAL.
[P,T] = algo544(n,C);

%Calcula los autovalores de la matriz cuadrada " T " de orden " n "
%y los almacena en " lambda ".
fprintf('Autovalores obtenidos\n')
lambda = QR_simetrico(n,T)

%Ahora calculamos los autovectores
%de la matriz Tridiagonal T obtenida al transformar la matriz simetrica
%inicial C a su forma Hessemberg.
fprintf('Autovectores respectivos\n')
[Y] = autovectores(n,C,P,T,lambda);

% procedemos a calcular
%los autovectores de la pencil (A,B) simetrica.

X = [];
for i=1:n
    y = Y(:,i);
    %resolver: funcion implementada que permite resolver un sistema de
    %ecuaciones lineales Ax = b donde se factoriza la matriz en A = LU para

```

```

    %resolver los sistemas " Ly = b " por " sustitucion hacia adelante "
    %y luego el sistema " Ux = y " por " sustitucion hacia atras "
    x = resolver(L',y);
    X = [X x'];
end
X .

%C O M P R O B A C I O N
%Lo que sigue es tan solo para comprobar los resultados
%debe de cumplirse Ax = lambdaBx
Ax = [];
LBx = [];
for i=1:n
    Ax = [Ax A*X(:,i)];
    LBx = [LBx lambda(i)*B*X(:,i)];
end
fprintf('Comprobar los resultados por columna\n')
Ax
LBx

```

A.19. Cholesky

```

function [H] = cholesky(n,A)
% n : Orden de la matriz definida positiva cuadrada A
H = zeros(n,n);
for k=1:n
    for i=1:(k-1)
        suma = 0;

```

```

    for j=1:(i-1)
        suma = suma + H(i,j)*H(k,j);
    end
    H(k,i) = (A(k,i)-suma)/H(i,i);
end
suma = 0;
for j=1:(k-1)
    suma = suma + H(k,j)^2;
end
H(k,k) = sqrt(A(k,k) - suma);
end

```

A.20. Hessenberg

```

function [P,H] = algo544(n,A)
% Esta funcion permite transformar la matriz simetrica A en una matriz
% tridiagonal.
% A: es una matriz simetrica
% n: orden de la matriz simetrica cuadrada A
%k = 1
P = eye(n,n);
for k=1:(n-2)
    k;
    x = A(k+1:n,k);
    [P1,u,sigma] = algo541(n-k,x);
    P1 = [eye(k,k) zeros(k,n-k);zeros(n-k,k) P1];

```

```

P = P*P1;
A = P1*A*P1';
end
H = A;

```

A.21. QRsimetrico

```

function [lambda] = QR_simetrico(n,A);
%Este programa permite calcular los autovalores de la matriz
%cuadrada " A " de orden " n ".
%A= [ 1 2;3 4]
%A = [ 4   -2  -10   -8   -2   2   2;
%     -2   5   17   -2   5   3   9;
%     -10  17   70   -4   -1  16  52;
%     -8   -2   -4   78   38  12  12;
%     -2   5   -1   38  121 -23  15;
%     2   3   16   12 -23  43  66;
%     2   9   52   12  15  66  217]

B = A;
m = 7;
n = m;
lambda = [];
subm = 1;
while subm<(m)
    k = 0;
    Tk = A;
    r = (Tk(n-1,n-1) - Tk(n,n))/2;

```

```

w = Tk(n,n) + r - signo(r)*sqrt(r^2 + Tk(n,n-1)^2);
tol = 1e-5;
Nmax = 100; %Numero maximo de iteraciones
%Este es el algoritmo Iteracion QR con desplazamiento simple " w "

while (abs(Tk(n,n-1))>=tol & k<=Nmax)
    k = k + 1;
    Tk = Tk - w*eye(n,n);
    [Qk,Rk] = FQR(n,Tk); %Calculamos la factorizacion Tk = Qk*Rk
    Tk1 = Rk*Qk + w*eye(n,n);
    Tk = Tk1;
    r = (Tk(n-1,n-1) - Tk(n,n))/2;
    w = Tk(n,n) + r - signo(r)*sqrt(r^2 + Tk(n,n-1)^2);
end

lambda = [lambda;Tk1(n,n)];

n = m - subm;
A = Tk1(1:n,1:n);
subm = subm+1;
end

lambda = [lambda;A]; %Aqui se almacenan los autovalores de la matriz
%original A.

```

A.22. Householder

```

function [H,u,sigma] = algo541(n,x)
% n : longitud del vector " x " (como vector columna)
% Esta funcion crea ceros en el vector x mediante el metodo de Householder

```

```

% de manera que Hx = (sigma,0,0,0,...,0)
m = max(abs(x));
u = x/m;
signou1 = signo(u(1));
norma2u = 0;
for k=1:n
    norma2u = norma2u + u(k)^2;
end
norma2u = sqrt(norma2u);

sigma = signou1*norma2u;

u(1) = u(1) + sigma;
sigma = -m*sigma;

H = eye(n,n) - 2*u*u'/(u'*u);

```

A.23. Signo

```

function s = signo(num)
%Calcula el signo del vector Householder
if num > 0
    s = 1;
else
    if num<0
        s = -1;
    else
        s = 0;
    end
end

```

end

end

A.24. Vectores Propios

```
function [V] = autovectores(n,C,P,T,lambda)
%Iteracion Inversa de Hessemberg halla los autovectores

V = [];
for i=1:n
    exito = 0;
    k = 0;
    yk = ones(n,1);
    Nmax = 1000;
    H = T;
    %tol = 1e-5;
    while (exito == 0 & k<Nmax)
        k = k + 1;
        Hk = H - lambda(i)*eye(n,n);
        %zk = resolver(Hk,yk);
        zk = tridiagonal(n,Hk,yk); %Resolvemos un sistema tridiagonal
        yk = zk'/max(zk);
        if norm((H - lambda(i)*eye(n,n))*yk)<=norm(C,inf)
            exito = 1;
            V = [V P*yk];
        end
    end
end
end
end
```

A.25. Tridiagonal

```
function x = tridiagonal(n,A,b)
%resuelve el sistema Ax=b donde A es una matriz tridiagonal
%b : vector columna
A = [A b] %Construimos la matriz aumentada del sistema Ax = b
L(1,1) = A(1,1);
U(1,2) = A(1,2)/L(1,1);
z(1) = A(1,n+1)/L(1,1);

for i=2:(n-1)
    L(i,i-1) = A(i,i-1);
    L(i,i) = A(i,i) - L(i,i-1)*U(i-1,i);
    U(i,i+1) = A(i,i+1)/L(i,i);
    z(i) = (A(i,n+1) - L(i,i-1)*z(i-1))/L(i,i);
end

L(n,n-1) = A(n,n-1);
L(n,n) = A(n,n) - L(n,n-1)*U(n-1,n);
z(n) = (A(n,n+1) - L(n,n-1)*z(n-1))/L(n,n);

x(n) = z(n);
for i=(n-1):-1:1
    x(i) = z(i) - U(i,i+1)*x(i+1);
end
```

Bibliografía

- [1] R. L. Burden, Análisis Numérico, Grupo Editorial Iberoamericana , 1985
- [2] B. N. Datta, Numerical Linear Algebra and Applications, Books Publishing Company. 1995.
- [3] R. A. DeCarlo, Linear Systems: A State Variable Approach with Numerical Implementation, Printice Hall, 1989.
- [4] P. Deift, "The Bidiagonal Singular Value Decomposition and Hamiltonian Mechanics," SIAM J. Numer. Anal. v 1463-1516,1991.
- [5] B. P. Deminovich y I. A. Maron, Cálculo Numérico Fundamental, Paraninfo- Madrid, 1985.
- [6] G. E. Forsythe, M. A. Malcolm and C. B. Moler, Computer Methods for Mathematical Computations, Englewood Cliffs. N. J. Printice Hall.
- [7] N. Gastinel, Análisis Numérico Lineal, Editorial Reverté. S.A.
- [8] G. H. Golub and Van Loan, Matrix Computations, Johns Hopkins Univ. Press, Baltimore Maryland, New York, 1989
- [9] R. Horn and C. R. Johnson, Matrix Analysis. Cambridge, University Press, 2005.
- [10] D. Kincaid y W. Cheney, Análisis Numérico, Addison-Wesley Iberoamericana
- [11] M. J. Maron, Análisis Numérico, CECSA México, 1995.

- [12] Daniel Kressner, Numerical Methods for General and Structural Eigenvalue Problems. Springer 2005
- [13] E. D. Nering, Álgebra Lineal y Teoría de Matrices, Editorial Limusa, México.
- [14] B. Noble y J. W. Daniel, Álgebra Lineal Aplicada, Prentice-Hall Hispanoamericana. S.A.
- [15] A. Ralston, Análisis Numérico, un primer curso, McGraw-Hill Company.
- [16] Y. Saad, Numerical Methods for Large Eigenvalue Problems: Theory and Algorithms. New York: John Wiley, 1992.
- [17] G. W. Stewart, Matrix Algorithms. Volume I: Basic Decompositions, Siam 1998.
- [18] J. Stoer and R. Bulirsch, Introduction to Numerical Analysis, Springer Verlag. 1992
- [19] J. H. Wilkinson, The Algebraic Eigenvalue Problem. Clarendon Press, Oxford-1965.