



FACULTADE DE MATEMÁTICAS

Traballo Fin de Grao

MÉTODO DE GIVENS-HYMAN PARA CÁLCULO DE AUTOVALORES DE GRANDES MATRICES

ROSA MARÍA TÚÑEZ ALCALDE

XUÑO, 2022

UNIVERSIDADE DE SANTIAGO DE COMPOSTELA

GRAO DE MATEMÁTICAS

Traballo Fin de Grao

MÉTODO DE GIVENS-HYMAN
PARA CÁLCULO DE
AUTOVALORES DE GRANDES
MATRICES

ROSA MARÍA TÚÑEZ ALCALDE

XUÑO, 2022

UNIVERSIDADE DE SANTIAGO DE COMPOSTELA

Traballo proposto

Área de Coñecemento: Matemática Aplicada
Título: Método de Givens-Hyman para cálculo de autovalores de grandes matrices
Breve descripción do contido
A primeira parte do traballo estaría dedicada aos conceptos e resultados fundamentais relacionados co método de Givens de redución a matrices Hessenberg superiores e co método de Hyman para o cálculo e resolución do polinomio característico destas matrices. A segunda parte estaría dedicada á implementación e posta a punto dos métodos.
Recomendacións
Outras observacións

Índice

Resumo	VII
Introdución	1
1. Métodos de cálculo de autovalores	5
1.1. Métodos “directos”	5
1.2. Métodos iterativos	6
2. Método de Givens	9
2.1. Rotacións de Givens	9
2.2. Método de Givens para redución dunha matriz a unha semellante de tipo Hessenberg superior	14
2.3. Programación en Matlab	18
3. O método de Hyman para unha matriz Hessenberg superior	25
3.1. Programación en Matlab	28
4. Método de bisección de Givens para matrices tridiagonais simétricas	33
4.1. Algoritmo de bisección de Givens	39
4.2. Programación en Matlab	41
Bibliografía	47

Resumo

O traballo comeza cunha breve introdución acerca da importancia dos autovalores de grandes matrices na nosa vida diaria para resolver diferentes tipos de problemas. Móstrase, ademais, unha clasificación dos diferentes métodos de cálculo de valores propios para grandes matrices e para centrarse despois no método de cálculo de autovalores baseado en reducir unha matriz calquera a unha semellante (e polo tanto cos mesmos autovalores) de tipo Hessenberg superior mediante matrices de rotación de Givens e utilizar o método de Hyman para avaliar e resolver o polinomio característico desta matriz. Ademais, estudárase a adaptación do método de Givens para matrices simétricas que, por tanto, se reducen a matrices tridiagonais simétricas, o que permite utilizar a estratexia de Sturm para o cálculo de autovalores deste tipo de matrices. Os métodos son descritos tanto dende o punto de vista matemático como computacional detallando a súa codificación e programación en Matlab. Os códigos engádense ao final de cada tema e o texto complétase cos resultados obtidos no cálculo de autovalores de matrices tomadas como exemplo, comparando os resultados cos proporcionados directamente polo código Matlab (comando *eig*).

Abstract

The project begins with a brief introduction about the importance of the eigenvalues of large matrices in our daily life to solve different types of problems. We also show a classification of the different methods of computing the eigenvalues for large matrices and we focus later on the method based on reducing any matrix to a similar one (and therefore with the same eigenvalues) of upper Hessenberg type using Givens rotation matrices and finally to apply the Hyman's method to evaluate and solve its characteristic polynomial. In addition, we study the adaptation of the Givens method for symmetric matrices, which, therefore, are reduced to symmetric tridiagonal matrices, which allow us to use the Sturm's strategy combined with the bisection method for approximate the eigenvalues of this type of matrices. The methods are described both from

a mathematical and computational point of view, detailing their coding and programming in Matlab. The codes are added at the end of each topic and the text is completed with the results obtained in the calculation of eigenvalues of matrices taken as an example, comparing the results with those provided directly by the Matlab code (*eig* command).

Introdución

Denotamos por $\mathcal{M}_{n \times n}(\mathbb{R})$ ao espazo de matrices reais cadradas de dimensión n (é dicir, de n filas e n columnas). O caso de matrices de números complexos pode ser tratado do mesmo xeito pero non se considerará neste traballo.

Os autovalores dunha matriz cadrada $A \in \mathcal{M}_{n \times n}(\mathbb{R})$ son as raíces da ecuación característica da matriz, isto é, da ecuación polinómica:

$$p(\lambda) = \det(A - \lambda I) = (-1)^n [\lambda^n + a_1 \lambda^{n-1} + \dots + a_{n-1} \lambda + a_n] = 0,$$

sendo $p(\lambda)$ o polinomio característico da matriz A .

Polo tanto unha matriz de orde n posúe n autovalores, non necesariamente distintos, contando cada un tantas veces como a súa multiplicidade como raíz do polinomio $p(\lambda)$. Así, calcular os autovalores dunha matriz non é nada máis que calcular as raíces do polinomio característico.

Unha simple verificación mostra que un polinomio arbitrario de grao n da forma

$$p : \lambda \in \mathbb{C} \longrightarrow p(\lambda) = (-1)^n [\lambda^n + a_1 \lambda^{n-1} + \dots + a_{n-1} \lambda + a_n], \text{ con } a_i \in \mathbb{R}$$

é o polinomio característico da matriz chamada *matriz de compañía*:

$$\begin{pmatrix} -a_1 & -a_2 & -a_3 & \dots & -a_{n-1} & -a_n \\ 1 & 0 & 0 & \dots & 0 & 0 \\ & 1 & 0 & \dots & 0 & 0 \\ & & \ddots & & & \vdots \\ & & & \ddots & & \vdots \\ & & & & 1 & 0 \end{pmatrix} \quad (1)$$

Isto vén a dicir que todo polinomio é polinomio característico dalgunha matriz e polo tanto os métodos de cálculo de valores propios dunha matriz son en esencia métodos para calcular as raíces de calquer polinomio. Dado que o célebre *Teorema de Abel* (teorema fundamental

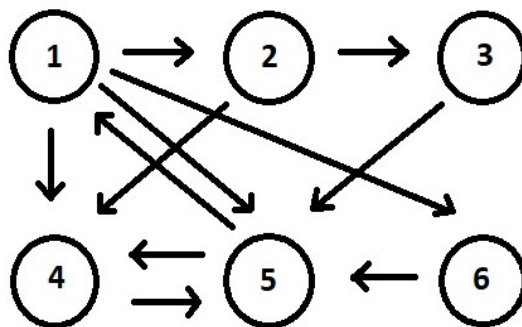
da álgebra) asegura que é imposible resolver cun número finito de operacións elementais un polinomio xeral de grao maior ou igual que 5, concluímos que os métodos de cálculo de autovalores son necesariamente iterativos (ver Capítulo 1).

Estamos diante dun problema de grande dificultade teórica ao que se tratou de dar diversas solucións ao longo da historia dada a súa importancia nas aplicacións da vida real. No Capítulo 1 damos un breve resumo sobre os métodos máis importantes e no resto do traballo profundízase no método de Givens-Hyman.

O cálculo dos autovalores (e nalgúns casos de autovectores) de grandes matrices está moi presente na nosa vida para poder resolver importantes problemas na ciencia e na enxeñería: sistemas de comunicacións e ondas electromagnéticas, circuitos eléctricos, análise de vibracións de estruturas, acústica, deseño de pontes e estruturas civís, terremotos, exploracións subterráneas en minería e petróleo e ata nos buscadores de páxinas web, son algúns dos numerosos exemplos nos que de maneira directa, é necesario o cálculo de autovalores de grandes matrices. Destacan os seguintes:

1. *Deseño de pontes e viaductos.* A frecuencia natural dunha ponte é o autovalor de menor magnitude dun sistema que modela a ponte. Os enxeñeiros aproveitan este coñecemento para asegurar a estabilidade das súas construcións. O caso de Tacoma Narrow Bridge inaugurado en 1940 é un exemplo da importancia dos valores propios xa que debido a que non se tivo en conta no deseño da ponte, fixo que ésta caera aos poucos meses da inauguración con vento de tan só 68 quilómetros por hora que impactaba de forma transversal na estrutura da ponte. A razón foi unha resonancia provocada pola coincidencia da frecuencia do vento coa da propia estrutura, que non fora debidamente avaliada.
2. *Localización reservas de petróleo.* As compañías petroleiras utilizan con frecuencia o estudo de valores propios para explorar terreos en busca de petróleo. A análise de ondas de choque con bolsas de petróleo e doutras substancias dá lugar a sistemas lineais que posúen distintos valores propios. A análise destes autovalores pode dar unha boa indicación de onde se atopan as reservas de petróleo e a composición das capas de terreo que as cubren, importantes para a toma de decisións nas perforacións.
3. *Búsqueda de páxinas web.* O extraordinario éxito de Google como motor de búsqueda debeuse ao uso de valores propios e vectores propios. Os métodos de Google para atopar o resultado máis relevante nas consultas de búsqueda evolucionou de moitas formas. PageRank, marca rexistrada, creada e desenvolva por Google contén unha familia de algoritmos utilizados para asignar de forma numérica a relevancia dos documentos ou páxinas web indexados por un motor de búsqueda, publicouse en 1998 e sufriu moitas renovacións. Para ver a idea orixinal de PageRank asumamos que unha web contén só 6 páxinas. O autor da

páxina 1 pensa que as páxinas 2, 4, 5 e 6 teñen un bo contido e entra nelas. O autor da páxina 2 só lle gusta os enlaces 3 e 4 así que só entra nesas. As preferencias do resto dos autores está resumida no seguinte diagrama:



Pois ben, a súa tarefa era atopar a páxina máis importante para unha consulta de búsqueda en particular, como o indicaron os autores das 6 páxinas. Por exemplo, se todos se vincularan coa páxina 1 e fose a única con 5 ingresos, entón sería lóxico que a páxina 1 aparecese na parte superior do resultado de búsqueda. Sen embargo, podemos ver que algunhas páxinas desta web, como a páxina 3 que ten un só ingreso, non se consideran moi importantes.

A base de PageRank era que consideraba as páxinas con moitas entradas (especialmente doutras páxinas populares) como as máis importantes e dáballes máis peso. Vexamos agora como se aplican os autovalores e os autovectores neste exemplo de 6 páxinas formando a matriz de enlaces que representa as entradas e saídas de cada páxina.

A páxina 1 ten 4 enlaces saíntes (ás páxinas 2, 4, 5 e 6). Entón na primeira columna da nosa “matriz enlace” poñemos o valor $1/4$ nas filas 2, 4, 5 e 6. No resto de filas da columna 1 poñemos o valor 0. Na páxina 2 temos 2 enlaces saíntes (ás páxinas 3 e 4) polo que poñemos $1/2$ nas filas 3 e 4 e 0 no resto. Así sucesivamente ata completar o proceso coas 6 páxinas. Finalmente quedaríanos a matriz:

$$A = \begin{bmatrix} 0 & 0 & 0 & 0 & \frac{1}{2} & 0 \\ \frac{1}{4} & 0 & 0 & 0 & 0 & 0 \\ 0 & \frac{1}{2} & 0 & 0 & 0 & 0 \\ \frac{1}{4} & \frac{1}{2} & 0 & 0 & \frac{1}{2} & 0 \\ \frac{1}{4} & 0 & 1 & 1 & 0 & 1 \\ \frac{1}{4} & 0 & 0 & 0 & 0 & 0 \end{bmatrix}$$

E, para conseguir os valores propios temos:

$$\det(A - \lambda I) = \det \begin{bmatrix} -\lambda & 0 & 0 & 0 & \frac{1}{2} & 0 \\ \frac{1}{4} & -\lambda & 0 & 0 & 0 & 0 \\ 0 & \frac{1}{2} & -\lambda & 0 & 0 & 0 \\ \frac{1}{4} & \frac{1}{2} & 0 & -\lambda & \frac{1}{2} & 0 \\ \frac{1}{4} & 0 & 1 & 1 & -\lambda & 1 \\ \frac{1}{4} & 0 & 0 & 0 & 0 & -\lambda \end{bmatrix} = \lambda^6 - \frac{5\lambda^4}{8} - \frac{\lambda^3}{4} - \frac{\lambda^2}{8}.$$

Esta expresión é cero para $\lambda = -0,72031, -0,13985 \pm 0,39240i, 0, 1$. Escollamos o valor de λ real e non negativo $\lambda = 1$, e calculemos un autovector asociado $v \in \mathbb{R}^6$ tal que $Av = v$.

$$v = [4 \ 1 \ 0,5 \ 5,5 \ 8 \ 1]^T.$$

Como a páxina 5 ten o PageRank máis alto (pois o valor máis alto no vector v é 8), concluímos que ésta é a máis importante e aparecerá de primeira na búsqueda de resultados.

Capítulo 1

Métodos de cálculo de autovalores

1.1. Métodos “directos”

Na introdución xa se advertiu que calquera polinomio é o polinomio característico dunha matriz (matriz de compañía) e polo tanto, calcular os valores propios dunha matriz arbitraria equivale a resolver calquera polinomio. Por iso os métodos só poden ser iterativos (cun número infinito de operacións). Sen embargo, na literatura soen chamarse *métodos “directos”* aos métodos centrados no cálculo efectivo do polinomio característico e da súa posterior resolución por métodos de cálculo de raíces de polinomios. Os métodos directos máis coñecidos son os métodos clásicos que permiten, nun número relativamente pequeno de operacións, o cálculo efectivo dos coeficientes do polinomio característico. O cálculo de autovalores, raíces do polinomio característico, realizárase por métodos apropiados de cálculo de raíces de polinomios (como poden ser o do punto fixo, Newton-Raphson, Bairstow...). Sen embargo, estes métodos non se soen utilizar debido ao mal condicionamento do problema do cálculo de raíces dun polinomio con respecto aos coeficientes do mesmo. Vexamos un exemplo para mostrar este feito (ver Wilkinson [1956]).

Sexa A unha matriz cuxo polinomio característico é o seguinte

$$p(\lambda) = (\lambda - 1)(\lambda - 2) \cdots (\lambda - 20) = \lambda^{20} - 210\lambda^{19} + a_2\lambda^{18} + \cdots + a_{20}$$

Supóñase que no cálculo dos coeficientes do polinomio anterior cometemos un erro simple no coeficiente correspondente a λ^{19} de xeito que o polinomio efectivamente calculado é:

$$q(\lambda) = \lambda^{20} - [210 + 2^{-23}]\lambda^{19} + a_2\lambda^{18} + \cdots + a_{20} = p(\lambda) - 2^{-23}\lambda^{19}$$

Esta simple perturbación ten efectos catastróficos sobre as raíces. En efecto, $q(\lambda)$ ten autovalores que pasarían a ser complexos con parte imaxinaria grande e polo tanto nada despreziable: $\pm 2,81i$,

$\pm 2,51i, \pm 1,94i, \dots$ É dicir, unha pequena variación nos datos orixina unha gran variación nos autovalores e aparece entón a necesidade de medir o condicionamento do problema de autovalores.

Isto explica a razón de que os métodos clásicos non deben utilizarse para matrices con orde $n > 15$. Moi ao contrario, os métodos de cálculo de valores propios de matrices aplícanse á matriz de compañía dun polinomio para aproximar as súas raíces. Entre os métodos máis coñecidos dos tempos anteriores ao manexo de ordenadores destacan os métodos de Leverrier, Souriau, Krylov, Danilevsky, Lanczos, entre outros, que teñen un fundamento teórico sorprendentemente simple.

1.2. Métodos iterativos

En canto aos métodos iterativos, os únicos realmente utilizados para grandes matrices danse a continuación nunha clasificación xeral.

Os métodos de cálculo de autovalores que observaremos a continuación dunha matriz A , en xeral, aplícanse a unha matriz semellante á matriz orixinal, que polas súas características resulta de maior simplicidade e/ou eficacia (matriz tridiagonal, Hessenberg superior,...).

Dúas matrices $A, B \in \mathcal{M}_{n \times n}(\mathbb{R})$ dinse que son *semellantes* se existe unha matriz invertible $P \in \mathcal{M}_{n \times n}(\mathbb{C})$ tal que

$$B = P^{-1}AP.$$

Como dicimos, a importancia da semellanza queda escrita na seguinte propiedade, *dúas matrices semellantes teñen o mesmo polinomio característico e, polo tanto, os mesmos autovalores*. En efecto:

$$\det(B - \lambda I) = \det(P^{-1}AP - \lambda P^{-1}P) = \det(P^{-1})\det(A - \lambda I)\det(P) = \det(A - \lambda I).$$

As transformacións de semellanza $A \rightarrow B = P^{-1}AP$ chámanse tamén “redución”, termo que aparece na seguinte clasificación.

Clasificación dos métodos iterativos.

1. *Métodos de resolución do polinomio característico*. Son métodos que non calculan o polinomio característico pero que sí o avalían nos puntos necesarios e, incluso, as súas derivadas, o que facilita a implementación dun método iterativo como dicotomía, Newton-Raphson, Regula-Falsi, etc., para o cálculo dunha ou máis raíces (autovalores). Os máis coñecidos son os seguintes:

- *Método de bisección de Givens para matrices tridiagonais simétricas.* ⊗
 - *Método de Hyman para matrices tipo Hessenberg superior utilizando Newton-Raphson, bisección ou outro método.* ⊗
2. *Métodos de reducción a forma condensada por transformaci3ns de semellanza.* Baséanse en transformar a matriz orixinal noutra semellante, cuxos autovalores (que son os mesmos que os da matriz dada) sexan máis simples de calcular.
- *Métodos para matrices simétricas:*
 - *Método de Jacobi de reducción a forma diagonal.* Proceso infinito que nos dá os autovalores da matriz na diagonal da matriz de chegada.
 - *Método de Givens de reducción a forma tridiagonal simétrica.* Proceso finito utilizando matrices de rotación plana. Os autovalores da matriz reducida calcúlanse polo método de bisección de Givens. ⊗
 - *Método de Householder de reducción a forma tridiagonal simétrica.* Proceso finito que utiliza matrices elementais de Householder. Os autovalores da matriz reducida calcúlanse polo método de bisección de Givens. O proceso completo chámase método de Givens-Householder.
 - *Métodos para matrices xerais:*
 - *Método de Givens de reducción a forma Hessenberg superior.* Proceso finito utilizando matrices de rotación plana. Os autovalores da matriz reducida calcúlanse polo método de Hyman con Newton-Raphson, dicotomía,..., ou o método QR. ⊗
 - *Método de Householder de reducción a forma Hessenberg superior.* Proceso finito utilizando matrices elementais de Householder. A matriz reducida trátase co método de Hymann máis Newton-Raphson ou outro método.
 - *Método de Jacobi de reducción dunha matriz normal a forma diagonal* [Gourlay-Watson, 1973, Cap.11].
 - *Método de Jacobi-Eberlein para matrices xerais:* aproxímase á matriz dada por outra normal semellante e aplícase o algoritmo de Jacobi para matrices normais [Gourlay-Watson, 1973].
- Observaci3n 1.1.* Os métodos de Givens e Householder de reduci3n dunha matriz simétrica a unha tridiagonal simétrica coinciden cos métodos con mesmo nome de reduci3n dunha matriz calquera a forma Hessenberg superior cando se aplica a unha matriz simétrica que loxicamente nos leva a unha matriz tridiagonal (a cal é Hessenberg superior). Polo tanto é suficiente estudar o caso xeral.
3. *Métodos de factorizaci3n.* Utilizando as factorizaci3ns LU e QR dunha matriz, constrúen unha sucesi3n de matrices semellantes a A que “converxe”, nun sentido que debe precisarse, a unha matriz triangular cuxos elementos diagonais son os autovalores da matriz orixinal.

- *Método LR de Rutishauser* (1952). Baseado na factorización LU.
 - *Método QR* de J.C.F.Francis (1960) e V.N.Kublanovskaya (1961). Baseado na factorización QR.
4. *Métodos iterativos para o autovalor dominante*. Constrúen unha sucesión de números complexos que converxe ao autovalor dominante ou a outro distinto. Son métodos que permiten o cálculo dun só valor propio en cada utilización. Entre os métodos que non precisan o cálculo do polinomio característico destaca pola súa sencillez o método da potencia iterada e todas as súas variantes. Existen varias modificacións:
- *Método da potencia iterada e variantes con cociente de Rayleigh*.
 - *Método da potencia inversa*. [Wienlandt, 1944].

Todos os métodos admiten estratexias máis ou menos válidas para aproximar un autovector asociado ao autovalor calculado. O resto da memoria estará dedicada á descrición, análise e implementación dos métodos sinalados cun \otimes na clasificación anterior.

Capítulo 2

Método de Givens

2.1. Rotacións de Givens

Unha rotación de Givens é unha rotación no plano abarcada por dous eixos de coordenadas. Estas rotacións levan o nome de Wallace Givens, xa que foi quen as presentou aos analistas numéricos na década de 1950 mentras traballaba no Laboratorio Nacional de Argonne (USA).

Unha matriz de rotación de Givens, $G(i, j, \theta) = (g_{ij})$, é unha matriz que coincide coa matriz identidade, excepto en catro entradas:

$$G(i, j, \theta) = \begin{pmatrix} 1 & 0 & \dots & 0 & \dots & 0 & \dots & 0 \\ 0 & 1 & \dots & 0 & \dots & 0 & \dots & 0 \\ \vdots & \vdots & \ddots & \vdots & \vdots & \vdots & \vdots & \vdots \\ 0 & 0 & \dots & c & \dots & s & \dots & 0 \\ \vdots & \vdots & \vdots & \vdots & \ddots & \vdots & \vdots & \vdots \\ 0 & 0 & \dots & -s & \dots & c & \dots & 0 \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \dots & 0 & \dots & 0 & \dots & 1 \end{pmatrix} \quad (2.1)$$

onde $c = \cos \theta$ e $s = \sin \theta$ aparecen nas interseccións i -ésimas e j -ésimas de filas e columnas. É dicir, para $i < j$ fixo, os elementos distintos de cero da matriz de Givens están dados por:

$$\begin{aligned} g_{kk} &= 1, \text{ para } k \neq i, j. \\ g_{kk} &= c = \cos \theta, \text{ para } k = i, j. \\ g_{ij} &= s = \sin \theta. \\ g_{ji} &= -s = -\sin \theta. \end{aligned}$$

Demostración. 1. Como A é simétrica,

$$B^T = (\Omega^T A \Omega)^T = \Omega^T A^T (\Omega^T)^T = \Omega^T A \Omega = B,$$

así que B é simétrica tamén.

Por outra parte é doado comprobar que a matriz Ω é ortogonal. A norma $\|\cdot\|_E$ é invariante por transformacións ortogonais. En efecto, se $\Omega \Omega^T = \Omega^T \Omega = I$, entón:

$$\begin{aligned} \|A\|_E^2 &= \text{Tr}(A^T A) = \text{Tr}(A^T \Omega^T \Omega A) = \text{Tr}((\Omega A)^T \Omega A) = \|\Omega A\|_E^2, \\ \|A\|_E^2 &= \text{Tr}(A A^T) = \text{Tr}(A \Omega \Omega^T A^T) = \text{Tr}(A \Omega (A \Omega)^T) = \|A \Omega\|_E^2. \end{aligned}$$

De onde se deduce tamén:

$$\|A\|_E^2 = \|\Omega A\|_E^2 = \|A \Omega\|_E^2 = \|\Omega^T A \Omega\|_E^2.$$

Polo tanto,

$$\sum_{i,j} a_{ij}^2 = \|A\|_E^2 = \|\Omega^T A \Omega\|_E^2 = \sum_{i,j} b_{ij}^2.$$

2. A continuación, observemos que a transformación que afecta aos elementos de índices (p, p) , (p, q) , (q, p) , (q, q) escríbese como

$$\begin{pmatrix} b_{pp} & b_{pq} \\ b_{qp} & b_{qq} \end{pmatrix} = \begin{pmatrix} \cos \theta & -\text{sen } \theta \\ \text{sen } \theta & \cos \theta \end{pmatrix} \begin{pmatrix} a_{pp} & a_{pq} \\ a_{qp} & a_{qq} \end{pmatrix} \begin{pmatrix} \cos \theta & \text{sen } \theta \\ -\text{sen } \theta & \cos \theta \end{pmatrix},$$

e o argumento utilizado en (1) mostra que

$$a_{pp}^2 + a_{qq}^2 + 2a_{pq}^2 = b_{pp}^2 + b_{qq}^2 + 2b_{pq}^2,$$

para calquer valor do número θ . Facendo operacións e utilizando as coñecidas fórmulas trigonométricas $\cos 2\theta = \cos^2 \theta - \text{sen}^2 \theta$ e $\text{sen } 2\theta = 2 \text{sen } \theta \cos \theta$, obtemos

$$b_{pq} = b_{qp} = a_{pq} \cos 2\theta + \frac{a_{pp} - a_{qq}}{2} \text{sen } 2\theta,$$

a elección do número θ indicado no enunciado implica que

$$b_{pq} = b_{qp} = 0,$$

e así

$$a_{pp}^2 + a_{qq}^2 + 2a_{pq}^2 = b_{pp}^2 + b_{qq}^2.$$

Como os coeficientes das diagonais principais son da seguinte forma $a_{ii} = b_{ii}$ para $i \neq p$ e $i \neq q$, obtemos

$$\sum_{i=1}^n b_{ii}^2 = \sum_{i=1}^n a_{ii}^2 + 2a_{pq}^2.$$

□

Observación 2.2. 1. A matriz Ω representa unha rotación de ángulo θ no plano p -ésimo e do q -ésimo vectores da base, o que proporciona outra maneira de ver que é ortogonal.

2. Así mesmo, observemos que na transformación $A \rightarrow B = \Omega^T A \Omega$ soamente se modifican as filas e columnas p -ésima e q -ésima da matriz A . De forma máis precisa, para todo valor do ángulo θ ,

$$\left\{ \begin{array}{ll} b_{ij} = a_{ij} & \text{se } i \neq p, q, j \neq p, q \\ b_{pi} = a_{pi} \cos \theta - a_{qi} \sin \theta & \text{se } i \neq p, q \\ b_{qi} = a_{pi} \sin \theta + a_{qi} \cos \theta & \text{se } i \neq p, q \\ b_{pp} = a_{pp} \cos^2 \theta + a_{qq} \sin^2 \theta - a_{pq} \sin 2\theta \\ b_{qq} = a_{pp} \sin^2 \theta + a_{qq} \cos^2 \theta + a_{pq} \sin 2\theta \\ b_{pq} = b_{qp} = a_{pq} \cos 2\theta + \frac{a_{pp} - a_{qq}}{2} \sin 2\theta \end{array} \right. \quad (2.3)$$

3. Grazas ás relacións existentes entre as funcións trigonométricas, os elementos da matriz B están, a pesar das apariencias, determinados por *relacións alxébricas* a partir dos elementos da matriz A .

Das fórmulas obtidas en (2.3) temos para o ángulo θ a seguinte ecuación

$$\tan 2\theta = \frac{2a_{pq}}{a_{qq} - a_{pp}}, \quad |\theta| \leq \frac{\pi}{4}.$$

A través das identidades trigonométricas, podemos calcular a partir destas ecuacións os valores c , s e, por (2.3), b_{ip} e b_{iq} .

Sen embargo é recomendable que se utilicen as seguintes fórmulas, as cales son máis estables. Primeiro calculamos o valor $\kappa = \cot 2\theta$ a partir de

$$\kappa := \frac{a_{qq} - a_{pp}}{2a_{pq}},$$

e o valor $t = \tan \theta$ como a raíz do módulo máis pequeno da ecuación cadrática

$$t^2 + 2\kappa t - 1 = 0.$$

Despexando t en función de κ obtemos $t = -\kappa \pm \sqrt{\kappa^2 + 1}$, e co dato adicional de que $|t| \leq 1$ (pois $|\theta| \leq \frac{\pi}{4}$) podemos precisar máis:

$$t = \begin{cases} -\kappa + \sqrt{\kappa^2 + 1}, & \text{se } \kappa \geq 0 \\ -\kappa - \sqrt{\kappa^2 + 1}, & \text{se } \kappa < 0 \end{cases}$$

Recordando que $|\theta| \leq \frac{\pi}{4}$ (polo que $\cos\theta > 0$) e usando $\tan^2\theta + 1 = \frac{1}{\cos^2\theta}$, obtemos os valores c e s de

$$c := \frac{1}{\sqrt{1+t^2}} (= \cos\theta),$$

$$s := tc = \frac{t}{\sqrt{1+t^2}} (= \sin\theta).$$

Agora xa podemos atopar expresións máis convenientes para b_{pp} e b_{qq} :

$$\begin{aligned} b_{pp} &= a_{pp} \cos^2\theta + a_{qq} \sin^2\theta - a_{pq} \sin 2\theta = \\ &= a_{pp}c^2 + a_{qq}s^2 - 2a_{pq}sc = \\ &= a_{pp} + s^2(a_{qq} - a_{pp}) - 2a_{pq}sc = \\ &= a_{pp} + \frac{t^2}{t^2+1}\kappa 2a_{pq} - \frac{2t}{t^2+1}a_{pq} = \\ &= a_{pp} + \frac{t(1-t)^2}{t^2+1}a_{pq} - \frac{2t}{t^2+1}a_{pq} = \\ &= a_{pp} - ta_{pq}. \end{aligned}$$

Análogamente, $b_{qq} = a_{qq} + ta_{pq}$. Entón, coa axuda de s , c e t , as fórmulas de (2.3) quedarían do seguinte xeito:

$$\begin{cases} b_{pi} = b_{ip} = ca_{pi} - sa_{qi}, & \text{se } i \neq p, q \\ b_{qi} = b_{iq} = sa_{pi} + ca_{qi}, & \text{se } i \neq p, q \\ b_{pp} = a_{pp} - ta_{pq} \\ b_{qq} = a_{qq} + ta_{pq} \\ b_{pq} = b_{qp} = 0 \\ b_{ij} = a_{ij}, & \text{no resto de casos} \end{cases}$$

Observación 2.3. Utilizando o número $\tau := \tan \frac{\theta}{2}$ que se calcula do xeito seguinte

$$\tau := \frac{s}{1+c}$$

podemos escribir as fórmulas anteriores na forma equivalente seguinte que resulta máis estable numericamente:

$$\begin{cases} b_{pi} = b_{ip} = a_{pi} - s(a_{qi} - \tau a_{pi}), & i \neq p, q \\ b_{qi} = b_{iq} = a_{qi} + s(a_{pi} + \tau a_{iq}), & i \neq p, q \\ b_{pp} = a_{pp} - ta_{pq} \\ b_{qq} = a_{qq} + ta_{pq} \\ b_{pq} = b_{qp} = 0 \\ b_{ij} = a_{ij}, & \text{no resto de casos.} \end{cases}$$

2.2. Método de Givens para redución dunha matriz a unha semellante de tipo Hessenberg superior

Unha matriz $B = (b_{ij}) \in \mathcal{M}_{n \times n}(\mathbb{R})$ dise que é de tipo Hessenberg superior se $b_{pq} = 0$, $3 \leq p \leq n$, $1 \leq q \leq p - 2$.

O método de Givens para transformar unha matriz A calquera noutra matriz de tipo Hessenberg superior utiliza matrices ortogonais elementais da forma

$$G(p, q, \theta) = \begin{pmatrix} 1 & 0 & \cdots & 0 & \cdots & 0 & \cdots & 0 \\ 0 & 1 & \cdots & 0 & \cdots & 0 & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots & \vdots & \vdots & \vdots & \vdots \\ 0 & 0 & \cdots & c & \cdots & s & \cdots & 0 \\ \vdots & \vdots & \vdots & \vdots & \ddots & \vdots & \vdots & \vdots \\ 0 & 0 & \cdots & -s & \cdots & c & \cdots & 0 \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & 0 & \cdots & 0 & \cdots & 1 \end{pmatrix} \quad (2.4)$$

O método de Givens para transformar unha matriz real $A \in \mathcal{M}_{n \times n}(\mathbb{R})$ nunha matriz semellante con forma Hessenberg superior utiliza matrices elementais ortogonais $\Omega(p, q)$ de rotacións de Givens para crear en cada paso unha matriz semellante a A . Con cada paso crearase un cero na posición (p, q) para $3 \leq p \leq n$, $1 \leq q \leq p - 2$. Polo tanto o número de pasos é $m = (n - 2) + (n - 1) + \cdots + 2 + 1 = \frac{1}{2}(n - 1)(n - 2)$ e o método resúmese nas m transformacións de semellanza seguintes:

$$A_1 = A; \quad A_{k+1} = \Omega_k^T A_k \Omega_k, \text{ con } k = 1, 2, \dots, m,$$

onde $\Omega_k = \Omega_k(p, q)$ é unha matriz de rotación plana escollida convenientemente para crear cero nunha posición concreta (p, q) , $3 \leq p \leq n$, $1 \leq q \leq p - 2$.

Despois de m pasos, a matriz de chegada é da forma Hessenberg superior e semellante a A :

$$A_{m+1} = \Omega_m^T A_m \Omega_m = \Omega_m^T \Omega_{m-1}^T A_{m-1} \Omega_{m-1} \Omega_m = \cdots = \Omega_m^T \Omega_{m-1}^T \cdots \Omega_1 A \Omega_1 \cdots \Omega_{m-1} \Omega_m.$$

No caso de que $\Omega = \Omega(p, q)$, a transformación

$$A \rightarrow B = \Omega^T A \Omega$$

só afecta ás filas e columnas p -ésima e q -ésima. Ademais, pódese observar cun exemplo ($n = 5$, $p = 2$, $q = 4$) os elementos afectados (neste caso son os marcados coas letras a , b ou c):

$$B = \begin{pmatrix} \times & a & \times & b & \times \\ a & c & a & c & a \\ \times & b & \times & b & \times \\ a & c & a & c & a \\ \times & b & \times & b & \times \end{pmatrix}$$

A estratexia de Givens consiste en escoller θ para que un elemento dos sinalados coas letras a ou b se anulen. Veremos a continuación como escoller os pares (p, q) e como calcular θ .

As fórmulas de cálculo de $B = \Omega^T A \Omega$ sendo $\Omega = \Omega(p, q)$ son as seguintes (se A é simétrica, os elementos simétricos contaránse unha soa vez):

$$\left\{ \begin{array}{ll} b_{ij} = a_{ij}, & \text{se } i \neq p, q, j \neq p, q \\ b_{pj} = a_{pj} \cos \theta - a_{qj} \sin \theta, & \text{se } j \neq p, q \\ b_{qj} = a_{pj} \sin \theta + a_{qj} \cos \theta, & \text{se } j \neq p, q \\ b_{ip} = a_{ip} \cos \theta - a_{iq} \sin \theta, & \text{se } i \neq p, q \\ b_{iq} = a_{ip} \sin \theta + a_{iq} \cos \theta, & \text{se } i \neq p, q \\ b_{pp} = a_{pp} \cos^2 \theta + a_{qq} \sin^2 \theta - (a_{pq} + a_{qp}) \sin \theta \cos \theta \\ b_{qq} = a_{pp} \sin^2 \theta + a_{qq} \cos^2 \theta + (a_{pq} + a_{qp}) \sin \theta \cos \theta \\ b_{pq} = a_{pq} \cos^2 \theta - a_{qp} \sin^2 \theta + (a_{pp} - a_{qq}) \sin \theta \cos \theta \\ b_{qp} = a_{qp} \cos^2 \theta - a_{pq} \sin^2 \theta + (a_{pp} - a_{qq}) \sin \theta \cos \theta \end{array} \right.$$

Observación 2.4 (Caso A simétrica.). Se A é simétrica, obter a matriz B é suficiente con reducir as fórmulas anteriores ás seguintes:

$$\left\{ \begin{array}{ll} b_{ij} = a_{ij}, & \text{se } i \neq p, q, j \neq p, q \\ b_{pj} = b_{jp} = a_{pj} \cos \theta - a_{qj} \sin \theta, & \text{se } j \neq p, q \\ b_{qj} = b_{jq} = a_{qj} \cos \theta + a_{pj} \sin \theta, & \text{se } j \neq p, q \\ b_{pp} = a_{pp} \cos^2 \theta + a_{qq} \sin^2 \theta - a_{pq} \sin 2\theta \\ b_{qq} = a_{pp} \sin^2 \theta + a_{qq} \cos^2 \theta + a_{pq} \sin 2\theta \\ b_{pq} = b_{qp} = a_{pq} \cos^2 \theta - a_{pq} \sin^2 \theta + (a_{pp} - a_{qq}) \sin \theta \cos \theta = \\ \quad = a_{pq} \cos 2\theta + \frac{a_{pp} - a_{qq}}{2} \sin 2\theta \end{array} \right.$$

Con estas fórmulas na mente, supoñamos 3 índices p, q, r e intentemos escoller θ para que

$$b_{qr} = a_{pr} \sin \theta + a_{qr} \cos \theta = 0; |\theta| \leq \frac{\pi}{4}, \theta \neq 0.$$

Entón,

- Se $a_{pr} = a_{qr} = 0$, para calquera valor $\theta \in (-\frac{\pi}{4}, \frac{\pi}{4})$ tense $b_{qr} = 0$.

- No caso contrario, bastará escoller θ tal que:

$$\theta \in \left(-\frac{\pi}{4}, \frac{\pi}{4}\right), \cos \theta = \frac{a_{pr}}{\xi}, \sin \theta = -\frac{a_{qr}}{\xi}, \text{ onde } \xi = [a_{pr}^2 + a_{qr}^2]^{\frac{1}{2}}.$$

A transformación ortogonal así definida denotarémola por $G(p, q, r)$. Givens probou que a orde óptima para “facen zeros” que leve a unha matriz A_{m+1} Hessenberg superior (tridiagonal simétrica se A é simétrica) é a seguinte:

1. *Etapa 1: Creación de zeros na primeira columna.* Aplícanse á matriz A de forma sucesiva as $n - 2$ transformacións $G(2, i, 1)$ con $i = 3, 4, \dots, n$ para crear zeros nas posicións $(i, 1)$ con $i = 2, 3, \dots, n$. Entón, as transformacións son do tipo:

$$A = A_1 \rightarrow A_2 = \Omega(2, 3)^T A_1 \Omega(2, 3) \rightarrow A_3 = \Omega(2, 4)^T A_2 \Omega(2, 4) \rightarrow \dots \rightarrow \\ \rightarrow A_{i+1} = \Omega(2, i+2)^T A_i \Omega(2, i+2) \rightarrow \dots \rightarrow A_{n-1},$$

onde a matriz A_{n-1} é da seguinte forma:

$$A_{n-1} = \begin{pmatrix} \times & \times & \dots & \dots & \times \\ \times & \times & \dots & \dots & \times \\ 0 & \times & \dots & \dots & \times \\ \vdots & \vdots & \ddots & \ddots & \vdots \\ 0 & \times & \dots & \times & \times \end{pmatrix}$$

2. *Etapa 2: Creación de zeros na segunda columna.* Aplícanse a A_{n-1} as $n - 3$ primeiras transformacións $G(3, i, 2)$, $i = 4, 5, \dots, n$, que anulan todos os elementos $(i, 2)$, con $i = 4, 5, \dots, n$ da segunda columna, deixando sen variación os termos xa anulados (o que se verifica atendendo ás fórmulas).
3. *Etapa k: Creación de zeros na columna k-ésima, $k = 3, 4, \dots, n$.* Aplícanse sucesivamente as $n - k - 1$ transformacións do tipo $G(k+1, i, k)$, $i = k+2, \dots, n$ que anulan (conservando os elementos anulados anteriormente) os elementos da columna k nas posicións (i, k) , $i = k+2, \dots, n$.

É evidente que ao cabo de $(n - 2) + (n - 3) + \dots + 2 + 1 = \frac{1}{2}(n - 1)(n - 2)$ transformacións chegarase a unha matriz Hessenberg superior (tridiagonal simétrica se A é unha matriz simétrica) ao ter en conta o dito anteriormente: a transformación do tipo $G(k+1, i, k)$ utiliza as matrices $\Omega(k+1, i)$ e polo tanto só interveñen as filas e columnas $k+1$ e i , é dicir, que en toda a etapa k interveñen as filas e columnas $k+1, k+2, \dots, n$. Ademais, posto que as operacións que se realizan son combinacións lineais, compróbase facilmente que os zeros xa conseguidos nunca son substituídos por elementos non nulos e se conservan así ata o remate do proceso.

Unha concisa interpretación matricial do método de Givens resúmese no seguinte teorema:

Teorema 2.5 (Givens, 1954). *Dada unha matriz real $A \in \mathcal{M}_{n \times n}(\mathbb{R})$ (respectivamente simétrica) existe unha matriz ortogonal Ω , produto de $m = \frac{1}{2}(n-1)(n-2)$ matrices ortogonais elementais de rotación plana tal que a matriz $\Omega^T A \Omega$, semellante a A , é de tipo Hessenberg superior (respectivamente tridiagonal simétrica).*

Observación importante. O método de Givens utilízase case exclusivamente no caso no que a matriz A é simétrica e, aínda neste caso, este método foi superado polo método de Householder debido a que require un número de operacións menor.

Exemplo 2.6. Ilustramos o método vendo un exemplo dunha matriz 4×4 xenérica. Sexa A unha matriz 4×4 :

$$A = \begin{pmatrix} a_{11} & a_{12} & a_{13} & a_{14} \\ a_{21} & a_{22} & a_{23} & a_{24} \\ a_{31} & a_{32} & a_{33} & a_{34} \\ a_{41} & a_{42} & a_{43} & a_{44} \end{pmatrix}$$

Utilizando o índice p para as filas e o índice q para as columnas, para a fila $p = 1$ emprégase a matriz $\Omega(2, 3)$ para anular o elemento $(p, q) = (3, 1)$ e realízase

$$\Omega(2, 3)^T A \Omega(2, 3)$$

Recórdese que a matriz $\Omega(2, 3)$ é do seguinte xeito:

$$\Omega(2, 3) = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & c & s & 0 \\ 0 & -s & c & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}$$

Seguindo con $p = 1$, queda por anular o elemento $(4, 1)$ e para iso utilízase a matriz de rotación $\Omega(2, 4)$ que é da seguinte forma:

$$\Omega(2, 4) = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & c & 0 & s \\ 0 & 0 & 1 & 0 \\ 0 & -s & 0 & c \end{pmatrix}$$

Agora teranse ceros na primeira columna, filas 3 e 4.

$$\begin{pmatrix} a_{11}^{(1)} & a_{12}^{(1)} & a_{13}^{(1)} & a_{14}^{(1)} \\ a_{21}^{(1)} & a_{22}^{(1)} & a_{23}^{(1)} & a_{24}^{(1)} \\ 0 & a_{32}^{(1)} & a_{33}^{(1)} & a_{34}^{(1)} \\ 0 & a_{42}^{(1)} & a_{43}^{(1)} & a_{44}^{(1)} \end{pmatrix}$$

Pásase entón á segunda columna, é dicir, $q = 2$ e anúlase o elemento $(4, 2)$ coa matriz $\Omega(3, 4)$,

$$\Omega(3, 4) = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & c & s \\ 0 & 0 & -s & c \end{pmatrix}$$

co que chegamos á matriz Hessenberg superior

$$\begin{pmatrix} a_{11}^{(1)} & a_{12}^{(1)} & a_{13}^{(1)} & a_{14}^{(1)} \\ a_{21}^{(1)} & a_{22}^{(1)} & a_{23}^{(1)} & a_{24}^{(1)} \\ 0 & a_{32}^{(2)} & a_{33}^{(2)} & a_{34}^{(2)} \\ 0 & 0 & a_{43}^{(2)} & a_{44}^{(2)} \end{pmatrix}$$

2.3. Programación en Matlab

A continuación móstrase a realización informática do algoritmo de Givens no caso simétrico, que aínda sendo o mesmo que no caso non simétrico permite obter vantaxes da simetría no que á realización de operacións se refire. En ambos casos realízase o código cos seus respectivos exemplos cuxas solucións serán utilizadas nos capítulos posteriores para calcular os seus autovalores.

En primeiro lugar tomamos como matriz non simétrica a seguinte

$$A = \begin{pmatrix} 30 & 2 & 3 & 13 \\ 5 & 11 & 10 & 8 \\ 9 & 7 & 6 & 12 \\ 4 & 14 & 15 & 1 \end{pmatrix}$$

cuxos autovalores, calculados co comando $eig(A)$ en Matlab son: $\lambda_1 = 39,3960$, $\lambda_2 = 17,8208$, $\lambda_3 = -9,5022$ e $\lambda_4 = 0,2854$.

Como pode verse a continuación a matriz Hessenberg superior calculada co algoritmo de Givens, cuxo código se adxunta é a seguinte:

$$B = \begin{pmatrix} 30 & 8,0577 & 8,8958 & -6,1595 \\ 11,0454 & 24,2131 & 8,1984 & 2,1241 \\ 0 & 13,5058 & -4,3894 & 7,8918 \\ 0 & 0 & 3,2744 & -1,8237 \end{pmatrix}.$$

Neste caso tamén se calculan os autovalores tanto da matriz orixinal como da matriz resultante co comando *eig* de Matlab, comprobando así que ambas matrices son semellantes.

En segundo lugar realízase o código para o caso simétrico coa seguinte matriz simétrica, chamada matriz de Wilson (Ciarlet, [1999]):

$$A = \begin{pmatrix} 10 & 7 & 8 & 7 \\ 7 & 5 & 6 & 5 \\ 8 & 6 & 10 & 9 \\ 7 & 5 & 9 & 10 \end{pmatrix}$$

cuxos autovalores calculados co comando *eig* son: $\lambda_1 = 0,0102$, $\lambda_2 = 0,8431$, $\lambda_3 = 3,8581$ e $\lambda_4 = 30,2887$.

Do algoritmo de Givens para o caso simétrico, cuxo código adxuntamos a continuación, danos a seguinte matriz:

$$B = \begin{pmatrix} 10 & 12,7279 & 0 & 0 \\ 12,7279 & 21,8827 & 3,4281 & 0 \\ 0 & 3,4281 & 2,3825 & 0,4621 \\ 0 & 0 & 0,4621 & 0,7348 \end{pmatrix}.$$

Para asegurar que a matriz orixinal e a resultante son semellantes calcúlanse os autovalores a través do comando de matlab *eig* e compróbase que, en efecto, teñen os mesmos autovalores.

Código Matlab do algoritmo de Givens - Caso non simétrico.

```
a = [30 2 3 13;5 11 10 8;9 7 6 12;4 14 15 1]
Espectro = eig(a) % Autovalores da matriz a.
[n,n] = size(a);
for k = 1:n-2 % Bucle de columnas.
    p = k+1;
    r = k;
    for q = k+2:n
        x = sqrt((a(p,r))^2 +(a(q,r))^2);
        if x > 1.e-15
```

```

c = a(p,r)/x;
s = -a(q,r)/x;
app = a(p,p);
aqq = a(q,q);
apq = a(p,q);
aqp = a(q,p);
% Fórmulas para o caso non simétrico.
for j = 1:n
    ajp = a(j,p);
    ajq = a(j,q);
    apj = a(p,j);
    aqj = a(q,j);
    if (j ~= p)&&(j ~= q)
        a(j,p) = ajp*c - ajq*s;
        a(j,q) = ajq*c + ajp*s;
        a(p,j) = apj*c - aqj*s;
        a(q,j) = aqj*c + apj*s;
    end
end
a(p,p) = app*c^2 + aqq*s^2 - (apq + aqp)*c*s;
a(q,q) = app*s^2 + aqq*c^2 + (apq + aqp)*c*s;
a(p,q) = apq*c^2 - aqp*s^2 + (app - aqq)*s*c;
a(q,p) = aqp*c^2 - apq*s^2 + (app - aqq)*s*c;
end
end
a % Matriz Hessenberg calculada.
Espectro = eig(a) % Autovalores da nova matriz.
% Compróbase así que os autovalores da matriz orixinal e os da nova matriz
% son os mesmos.

```

A solución é a seguinte:

```

a =
30     2     3    13
 5    11    10     8
 9     7     6    12
 4    14    15     1

```

```
Espectro =
```

```
39.3960
```

```
17.8208
```

```
-9.5022
```

```
0.2854
```

```
a =
```

```
30.0000    8.0577    8.8958   -6.1595
```

```
11.0454   24.2131    8.1984    2.1241
```

```
0         13.5058   -4.3894    7.8918
```

```
0         0         3.2744   -1.8237
```

```
Espectro =
```

```
39.3960
```

```
17.8208
```

```
-9.5022
```

```
0.2854
```

Código Matlab do algoritmo de Givens - Caso simétrico.

```
a = [10 7 8 7;7 5 6 5;8 6 10 9;7 5 9 10]
Espectro = eig(a) % Autovalores da matriz a.
[n,n] = size(a);
for k = 1:n-2 % Bucle de columnas.
    p = k+1;
    r = k;
    for q = k+2:n
        x = sqrt((a(p,r))^2 + (a(q,r))^2);
        if x > 1.e-15
            c = a(p,r)/x;
            s = -a(q,r)/x;
            app = a(p,p);
            aqq = a(q,q);
            apq = a(p,q);
            % Fórmulas para o caso simétrico.
            for j = 1:n
                ajp = a(j,p);
```

```

    ajq = a(j,q);
    if (j ~= p)&&(j ~= q)
        a(j,p) = ajp*c - ajq*s;
        a(j,q) = ajq*c + ajp*s;
        a(p,j) = a(j,p);
        a(q,j) = a(j,q);
    end
end
a(p,p) = app*c^2 + aqq*s^2 - 2*apq*c*s;
a(q,q) = aqq*c^2 + app*s^2 + 2*apq*c*s;
a(p,q) = (app - aqq)*c*s + apq*(c^2-s^2);
a(q,p) = a(p,q);
end
end
a % Matriz tridiagonal calculada.
Espectro = eig(a) % Autovalores da nova matriz.
% Compróbase así que os autovalores da matriz orixinal e os da nova matriz
% son os mesmos.

```

A solución é a seguinte:

```

a =
10    7    8    7
 7    5    6    5
 8    6   10    9
 7    5    9   10

```

```

Espectro =
0.0102
0.8431
3.8581
30.2887

```

```

a =
10.0000  12.7279  0.0000  0.0000
12.7279  21.8827  3.4281  0.0000
0.0000  3.4281  2.3825  0.4621

```


0.0000 0.0000 0.4621 0.7348

Espectro =

0.0102

0.8431

3.8581

30.2887

Capítulo 3

O método de Hyman para unha matriz Hessenberg superior

Considérese $B = [b_{ij}]$ unha matriz de tipo Hessenberg superior e sexa $p(\mu) = \det(B - \mu I)$ o seu polinomio característico. Para resolver dito polinomio característico pódense utilizar diversos métodos de cálculo de raíces de polinomios, podendo aplicar por exemplo o método de Newton-Raphson xa que a avaliación dos polinomios $p(\mu)$ e $p'(\mu)$ é relativamente sinxela. O método de Hyman é un método simple para avaliar estas cantidades. Trátase dun método alternativo ao método QR (máis utilizado actualmente).

Así pois, como B é unha matriz Hessenberg superior, temos: $b_{ij} = 0, 3 \leq i \leq n, 1 \leq j \leq i-2$.

$$B = \begin{pmatrix} b_{11} & b_{12} & b_{13} & \cdots & \cdots & b_{1n} \\ b_{21} & b_{22} & b_{23} & \cdots & \cdots & b_{2n} \\ & b_{32} & b_{33} & \cdots & \cdots & b_{3n} \\ & & \ddots & \ddots & & \\ & & & \ddots & \ddots & \\ & & & & b_{n,n-1} & b_{nn} \end{pmatrix} \quad (3.1)$$

Supoñáse ademáis que a matriz B é irreducible, é dicir, que $b_{i,i-1} \neq 0$ para $i \in 2, 3, \dots, n$. No caso de que algún destes elementos sexan nulos, entón podemos considerar que a matriz B é triangular superior por bloques. Cada un destes bloques diagonais é unha matriz Hessenberg superior que pode ser tratada independentemente por este método e polo tanto resolve o problema xa que o espectro da matriz B é a unión dos espectros dos bloques diagonais.

Fixado μ pódense determinar os valores $\alpha(\mu), x_1(\mu), \dots, x_{n-1}(\mu)$ tales que $x(\mu) = [x_1(\mu), \dots, x_n(\mu)]^T \in \mathbb{R}^n$ con $x_n(\mu) = 1$ é unha solución do sistema lineal

$$(B - \mu I)x(\mu) = \alpha(\mu)e_1$$

ou, escrito doutro xeito,

$$\begin{cases} (b_{11} - \mu)x_1(\mu) + b_{12}x_2(\mu) + \cdots + b_{1n}x_n(\mu) = \alpha(\mu), \\ b_{21}x_1(\mu) + (b_{22} - \mu)x_2(\mu) + \cdots + b_{2n}x_n(\mu) = 0, \\ \vdots \\ b_{n,n-1}x_{n-1}(\mu) + (b_{nn} - \mu)x_n(\mu) = 0. \end{cases} \quad (3.2)$$

Para calcular o valor de $\alpha(\mu)$ o método máis sinxelo é resolver por recursión o sistema lineal (3.2). Comezando con $x_n(\mu) = 1$, un pode determinar o valor de $x_{n-1}(\mu)$ da última ecuación, $x_{n-2}(\mu)$ da penúltima ecuación e así sucesivamente ata atopar $x_1(\mu)$ da segunda ecuación e, finalmente, $\alpha(\mu)$ da primeira ecuación.

Pois ben, tomando $x_n(\mu) = 1$ como se mencionou anteriormente é relativamente sinxelo calcular $\det(B - \mu I)$ en función de $\alpha(\mu)$. En efecto, utilizando a regra de Cramer no sistema (3.2) de incógnitas $x_1(\mu), x_2(\mu), \dots, x_n(\mu)$ e tendo termo independente $(\alpha(\mu), 0, \dots, 0)^T \in \mathbb{R}^n$, tense

$$x_n(\mu) = \frac{\begin{vmatrix} b_{11} - \mu & b_{12} & \cdots & b_{1,n-1} & \alpha(\mu) \\ b_{21} & b_{22} - \mu & \cdots & b_{2,n-1} & 0 \\ & & \ddots & & \\ & & & b_{n-1,n-1} - \mu & 0 \\ & & & b_{n,n-1} & 0 \end{vmatrix}}{\begin{vmatrix} b_{11} - \mu & b_{12} & \cdots & b_{1,n-1} & b_{1n} \\ b_{21} & b_{22} - \mu & \cdots & b_{2,n-1} & b_{2n} \\ & & \ddots & & \\ & & & b_{n-1,n-1} - \mu & b_{n-1,n} \\ & & & b_{n,n-1} & b_{nn} - \mu \end{vmatrix}}$$

Desenvolvendo o determinante do numerador recursivamente pola última fila, temos o seguinte:

$$x_n(\mu) = \frac{(-1)^{n+1} \alpha(\mu) b_{21} b_{32} \cdots b_{n,n-1}}{\det(B - \mu I)}.$$

Tomando $x_n(\mu) = 1$ teríamos:

$$\det(B - \mu I) = (-1)^{n+1} \alpha(\mu) b_{21} b_{32} \cdots b_{n,n-1}.$$

Polo tanto, $\alpha(\mu)$, salvo unha constante, é o polinomio característico. Se derivamos con respecto a μ e observando $x_n(\mu) = 1$, $x'_n(\mu) = 0$, un pode obter de (3.2) o seguinte sistema,

$$\begin{cases} (b_{11} - \mu)x'_1(\mu) - x_1(\mu) + b_{12}x'_2(\mu) + \cdots + b_{1,n-1}x'_{n-1}(\mu) = \alpha'(\mu), \\ b_{21}x'_1(\mu) + (b_{22} - \mu)x'_2(\mu) - x_2(\mu) + \cdots + b_{2,n-1}x'_{n-1}(\mu) = 0, \\ \vdots \\ b_{n,n-1}x'_{n-1}(\mu) - x_n(\mu) = 0. \end{cases} \quad (3.3)$$

Así, a incógnita $x'_{n-1}(\mu)$ pode determinarse na última ecuación, $x'_{n-2}(\mu)$ na penúltima ecuación e así sucesivamente ata chegar a $x'_1(\mu)$ que se saca da segunda ecuación e, finalmente, $\alpha'(\mu)$ que sae da primeira ecuación. Para concluir, a derivada do polinomio p en μ obtense da igualdade:

$$p'(\mu) = (-1)^{n+1}\alpha'(\mu)b_{21}b_{32} \cdots b_{n,n-1}.$$

O método de Hyman queda pois resumido no seguinte teorema no que se dán as fórmulas recurrentes para a solución dos sistemas anteriores.

Teorema 3.1. (Hyman) *Sexa B unha matriz de tipo Hessenberg superior $B = (b_{ij})$ con elementos subdiagonais $b_{i,i-1} \neq 0$ para todo $i = 2, 3, \dots, n$. Para un valor arbitrario μ , constrúense as seguintes sucesións finitas:*

$$\begin{aligned} x_1(\mu), x_2(\mu), \dots, x_n(\mu) \\ y_1(\mu), y_2(\mu), \dots, y_n(\mu) \end{aligned}$$

mediante as relacións recurrentes que se mostran a continuación:

$$\begin{cases} x_n(\mu) = 1, \\ x_i(\mu) = -\frac{1}{b_{i+1,i}}[(b_{i+1,i+1} - \mu)x_{i+1}(\mu) + b_{i+1,i+2}x_{i+2}(\mu) + \cdots + b_{i+1,n}x_n(\mu)], i = n-1, \dots, 1, \\ y_n(\mu) = 0, \\ y_i(\mu) = -\frac{1}{b_{i+1,i}}[(b_{i+1,i+1} - \mu)y_{i+1}(\mu) + b_{i+1,i+2}y_{i+2}(\mu) + \cdots + b_{i+1,n}y_n(\mu) - x_{i+1}(\mu)], i = n-1, \dots, 1. \end{cases}$$

Entón:

$$p(\mu) = \det(B - \mu I) = (-1)^{n+1}\alpha(\mu)b_{21}b_{32} \cdots b_{n,n-1},$$

$$p'(\mu) = (-1)^{n+1}\beta(\mu)b_{21}b_{32} \cdots b_{n,n-1},$$

onde:

$$\begin{aligned}\alpha(\mu) &= (b_{11} - \mu)x_1(\mu) + b_{12}x_2(\mu) + \cdots + b_{1n}x_n(\mu), \\ \beta(\mu) &= (b_{11} - \mu)y_1(\mu) + b_{12}y_2(\mu) + \cdots + b_{1n}y_n(\mu) - x_1(\mu).\end{aligned}$$

Non se ten, entón, unha expresión analítica explícita de $p(\mu)$, mais pódese avaliar $p(\mu)$ e $p'(\mu)$ para calquera valor de μ . Pódense obter os autovalores de B utilizando o método iterativo de Newton-Raphson para resolver a ecuación $p(\mu) = 0$:

$$\mu^{(0)} \text{ dado, } \mu^{(k+1)} = \mu^{(k)} - \frac{p(\mu^{(k)})}{p'(\mu^{(k)})}, k \geq 0.$$

Como é natural, isto precisa ter localizados os autovalores que se queren calcular e proporcionar un valor inicial $\mu^{(0)}$. Se os autovalores que se pretenden calcular son de maultiplicidade maior que un, é dicir, por exemplo de multiplicidade m , terase en conta a variante de Schröder que é a seguinte:

$$\mu^{(0)} \text{ dado, } \mu^{(k+1)} = \mu^{(k)} - m \frac{p(\mu^{(k)})}{p'(\mu^{(k)})}.$$

Despois de ter calculados $i - 1$ valores propios $\mu_1, \mu_2, \dots, \mu_{i-1}$, para calcular outro dos autovalores, por exemplo μ_i , utilizarase o método de Newton-Raphson coa función:

$$F(\mu) = \frac{p(\mu)}{\prod_{j=1}^{i-1} (\mu_j - \mu)}$$

O método escríbese neste caso:

$$\mu^{(k+1)} = \mu^{(k)} - \left[\frac{p'(\mu^{(k)})}{p(\mu^{(k)})} + \sum_{j=1}^{i-1} \frac{1}{\mu_j - \mu^{(k)}} \right]^{-1}$$

3.1. Programación en Matlab

A continuación mostramos o código en Matlab do método e os resultados que se obteñen para a matriz de tipo Hessenberg superior obtida no caso non simétrico do capítulo 2. Recordamos que é

$$B = \begin{pmatrix} 30 & 8,0577 & 8,8958 & -6,1595 \\ 11,0454 & 24,2131 & 8,1984 & 2,1241 \\ 0 & 13,5058 & -4,3894 & 7,8918 \\ 0 & 0 & 3,2744 & -1,8237 \end{pmatrix}$$

cuxos autovalores son xa coñecidos, $\lambda_1 = 39,3960$, $\lambda_2 = 17,8208$, $\lambda_3 = -9,5022$ e $\lambda_4 = 0,2854$.

Código Matlab do Método de Hyman con Newton-Raphson.

```

b = [30 8.0577 8.8958 -6.1596;11.0454 24.2131 8.1984 2.1241;
     0 13.5058 -4.3894 7.8918;0 0 3.2744 -1.8237];
[n,n] = size(b);
nmaxit = 100;
delta = 1.e-15;
eps = delta;
mu = 0.;
sp = zeros(n,1); % Variable para almacenar os autovalores.
for nautov=1:n
    disp(['BUSCANDO O AUTOVALOR NÚMERO: ', num2str(nautov)])
    ind = 0; mu = 0;
    k = 1;
    while k <= nmaxit
        % Cálculo do polinomio p.
        x(n) = 1;
        alfa = b(1,n)*x(n);
        for i=n-1:-1:1
            z=0.;
            for j = i+2:n
                z = z + b(i+1,j)*x(j);
            end
            x(i) = -(1/b(i+1,i))*((b(i+1,i+1)-mu)*x(i+1) + z);
            alfa = alfa + b(1,i)*x(i);
        end
        alfa = alfa - x(1)*mu;
        p = ((-1)^(n+1))*alfa;
        for i = 2:n
            p = p*b(i,i-1);
        end
        if abs(p) < delta
            disp(['Iteración: ', num2str(k), ' p(mu)=0. Último iterante,mu: ',
                 num2str(mu)])
            sp(nautov) = mu;
            k = nmaxit+1;
            ind = 1;
        end
    end
end

```

```

else
    % Cálculo da derivada do polinomio p (pp).
    y(n) = 0;
    beta = 0.;
    for i = n-1:-1:1
        z=0.;
        for j = i+2:n
            z = z + b(i+1,j)*y(j);
        end
        y(i) = -(1/b(i+1,i))*((b(i+1,i+1)-mu)*y(i+1) + z - x(i+1));
        beta = beta + b(1,i)*y(i);
    end
    beta = beta - y(1)*mu - x(1);
    pp = ((-1)^(n+1))*beta;
    for i = 2:n
        pp = pp*b(i,i-1);
    end
    % Aplicación do método de Newton-Raphson.
    if nautov == 1
        if abs(pp) < 1.e-25
            disp('Problema: valor da derivada casi nulo, Non podemos seguir')
            return
        end
        s = p/pp;
    else
        s = pp/p;
        for j=1:nautov-1
            s = s+1./(sp(j)-mu);
        end
        s = 1./s;
    end
    err = abs(s)/(abs(mu)+1);
    mu = mu-s;
    disp(['Iteración: k= ', num2str(k), ' err: ', num2str(err), ' mu_k+1: ',
        num2str(mu)])
    if err < eps
        disp(['Iteración ', num2str(k), ' Alcanzado test de parada.
            Último iterante, mu: ', num2str(mu)])
    end

```



```

        disp(['-----'])
        sp(nautov) = mu;
        ind = 1;
        k = nmaxit+1;
    end
end
k = k+1;
end
if ind == 0
    disp(['Alcanzado máximo número de iteraciones. Último iterante, mu: ',
        num2str(mu)])
    sp(nautov) = mu;
end
end
sp % Dá todos os autovalores.

```

A solución é a seguinte:

```

BUSCANDO 0 AUTOVALOR NÚMERO: 1
Iteración: k= 1  err: 0.28734  mu_k+1: 0.28734
Iteración: k= 2  err: 0.0015123  mu_k+1: 0.28539
Iteración: k= 3  err: 5.7636e-08  mu_k+1: 0.28539
Iteración: k= 4  err: 1.3906e-16  mu_k+1: 0.28539
Iteración 4 Alcanzado test de parada. Último iterante, mu: 0.28539
-----
BUSCANDO 0 AUTOVALOR NÚMERO: 2
Iteración: k= 1  err: 42.121  mu_k+1: -42.121
Iteración: k= 2  err: 0.38905  mu_k+1: -25.3446
Iteración: k= 3  err: 0.37311  mu_k+1: -15.5151
Iteración: k= 4  err: 0.28226  mu_k+1: -10.8535
Iteración: k= 5  err: 0.10614  mu_k+1: -9.5953
Iteración: k= 6  err: 0.0087429  mu_k+1: -9.5027
Iteración: k= 7  err: 4.6723e-05  mu_k+1: -9.5022
Iteración: k= 8  err: 1.3081e-09  mu_k+1: -9.5022
Iteración: k= 9  err: 3.159e-18  mu_k+1: -9.5022
Iteración 9 Alcanzado test de parada. Último iterante, mu: -9.5022
-----
BUSCANDO 0 AUTOVALOR NÚMERO: 3

```

```
Iteración: k= 1  err: 12.2703  mu_k+1: 12.2703
Iteración: k= 2  err: 0.34721  mu_k+1: 16.8779
Iteración: k= 3  err: 0.050617  mu_k+1: 17.7829
Iteración: k= 4  err: 0.0020137  mu_k+1: 17.8207
Iteración: k= 5  err: 3.5229e-06  mu_k+1: 17.8208
Iteración: k= 6  err: 1.0826e-11  mu_k+1: 17.8208
Iteración: k= 7  err: 7.1359e-17  mu_k+1: 17.8208
Iteración 7 Alcanzado test de parada. Último iterante, mu: 17.8208
```

```
-----
BUSCANDO 0 AUTOVALOR NÚMERO: 4
```

```
Iteración: k= 1  err: 39.3961  mu_k+1: 39.3961
Iteración: k= 2  err: 2.9652e-14  mu_k+1: 39.3961
Iteración: k= 3  err: 6.6633e-17  mu_k+1: 39.3961
Iteración 3 Alcanzado test de parada. Último iterante, mu: 39.3961
```

```
-----
sp =
0.2854
-9.5022
17.8208
39.3961
```

Capítulo 4

Método de bisección de Givens para matrices tridiagonais simétricas

Sexa A unha matriz tridiagonal simétrica (real) da seguinte forma:

$$A = \begin{pmatrix} a_1 & b_1 & & & \\ b_1 & a_2 & b_2 & & \\ & \ddots & \ddots & \ddots & \\ & & b_{n-2} & a_{n-1} & b_{n-1} \\ & & & b_{n-1} & a_n \end{pmatrix}. \quad (4.1)$$

O seu polinomio característico será denotado por $p_n(\lambda)$:

$$p_n(\lambda) = \det(A - \lambda I) = \det \begin{pmatrix} a_1 - \lambda & b_1 & & & \\ b_1 & a_2 - \lambda & b_2 & & \\ & \ddots & \ddots & \ddots & \\ & & b_{n-2} & a_{n-1} - \lambda & b_{n-1} \\ & & & b_{n-1} & a_n - \lambda \end{pmatrix}.$$

O noso obxectivo é calcular as raíces de $p_n(\lambda) = 0$, autovalores da matriz A , que por ser simétrica son todos reais. Supóñase que $b_i \neq 0$, con $i = 1, 2, \dots, n-1$, posto que se algún $b_i = 0$, a matriz é tridiagonal por bloques e poderíase traballar en cada bloque por separado.

Tal e como se mencionou anteriormente, cando n é grande ($n > 20$) non é aconsellable calcular os coeficientes do polinomio característico $p_n(\lambda)$ como un medio para aproximar as raíces (autovalores de A) pero o que sí é aceptable é avaliar o polinomio característico directamente (sen coñecer os coeficientes). Este é o aspecto básico do algoritmo de Givens para matrices tridiagonais simétricas e do algoritmo de Hyman para matrices Hessenberg superior.

Givens (1954) propón o seguinte proceso de recurrencia para expresar o polinomio característico de A . Sexa $p_i(\lambda)$ o determinante da submatriz principal de orde i da matriz $A - \lambda I$:

$$p_i(\lambda) = \det \begin{pmatrix} a_1 - \lambda & b_1 & & & & \\ b_1 & a_2 - \lambda & b_2 & & & \\ & & \ddots & \ddots & \ddots & \\ & & & b_{i-2} & a_{i-1} - \lambda & b_{i-1} \\ & & & & b_{i-1} & a_i - \lambda \end{pmatrix} = \det(A_i - \lambda I)$$

Entón para $i = 1, 2, 3$ obtemos:

$$\begin{aligned} p_1(\lambda) &= a_1 - \lambda, \\ p_2(\lambda) &= (a_2 - \lambda)p_1(\lambda) - b_1^2, \\ p_3(\lambda) &= (a_3 - \lambda)p_2(\lambda) - b_2^2 p_1(\lambda). \end{aligned}$$

Isto suxire a seguinte forma de recurrencia xeral:

Teorema 4.1. *Para a matriz tridiagonal simétrica A verificase a seguinte relación de recurrencia:*

$$\begin{cases} p_0(\lambda) = 1, \\ p_1(\lambda) = a_1 - \lambda, \\ p_i(\lambda) = (a_i - \lambda)p_{i-1}(\lambda) - b_{i-1}^2 p_{i-2}(\lambda), \quad \text{con } i = 2, 3, \dots, n. \end{cases} \quad (4.2)$$

Demostración. Basta desenvolver o determinante que define $p_i(\lambda)$ polos elemento da última fila. \square

Dado que cada unha das submatrices principais Δ_i da matriz A é tamén simétrica e $p_i(\lambda)$ é o polinomio característico de Δ_i , dedúcese que cada un dos polinomios $p_i(\lambda)$, $i = 1, 2, \dots, n$ ten i raíces reais.

O noso obxectivo é calcular as n raíces de $p_n(\lambda) = 0$, autovalores de A , pero sen calcular os seus coeficientes. En primeiro lugar temos o seguinte resultado de importancia:

Teorema 4.2. *Supoñamos que a matriz tridiagonal simétrica A con b_1, b_2, \dots, b_{n-1} distintos de cero e sexan $p_0(\lambda), p_1(\lambda), \dots, p_n(\lambda)$ os polinomios definidos para a seguinte relación de recurrencia:*

$$\begin{cases} p_0(\lambda) = 1, \\ p_1(\lambda) = a_1 - \lambda, \\ p_i(\lambda) = (a_i - \lambda)p_{i-1}(\lambda) - b_{i-1}^2 p_{i-2}(\lambda), \quad \text{con } i = 2, 3, \dots, n. \end{cases}$$

Entón, todas as raíces $\lambda_k^{(i)}$ con $k = 1, \dots, i$, dos polinomios p_i , $i = 1, \dots, n$ son reais e simples:

$$\lambda_1^{(i)} > \lambda_2^{(i)} > \dots > \lambda_i^{(i)},$$

e todas as raíces de p_{i-1} e p_i , respectivamente, sepáranse estrictamente da seguinte forma:

$$\lambda_1^{(i)} > \lambda_1^{(i-1)} > \lambda_2^{(i)} > \lambda_2^{(i-1)} > \dots > \lambda_{i-1}^{(i-1)} > \lambda_i^{(i)}.$$

Demostración. Procedemos por indución con respecto a i . O teorema é certo para $i = 1$. Asumamos que é certo para algún $i \geq 1$, isto é, que as raíces $\lambda_k^{(i)}$ e $\lambda_k^{(i-1)}$ de p_i e p_{i-1} , respectivamente, satisfán

$$\lambda_1^{(i)} > \lambda_1^{(i-1)} > \lambda_2^{(i)} > \lambda_2^{(i-1)} > \dots > \lambda_{i-1}^{(i-1)} > \lambda_i^{(i)}.$$

Por (4.2), p_k é da forma $p_k(\lambda) = (-1)^k \lambda^k + \dots$. En particular, o grao de p_k é igual a k . Entón, $p_{i-1}(\lambda)$ non cambia de signo para $\lambda > \lambda_1^{(i-1)}$ e como as raíces $\lambda_k^{(i-1)}$ son todas simples, temos inmediatamente que

$$\text{sign } p_{i-1}(\lambda_k^{(i)}) = (-1)^{i+k} \text{ para } k = 1, 2, \dots, i. \quad (4.3)$$

Ademáis por (4.2),

$$p_{i+1}(\lambda_k^{(i)}) = -b_i^2 p_{i-1}(\lambda_k^{(i)}) \text{ para } k = 1, 2, \dots, i.$$

Xa que $b_i^2 > 0$,

$$\begin{aligned} \text{sign } p_{i+1}(\lambda_k^{(i)}) &= (-1)^{i+k+1}, \quad k = 1, 2, \dots, i, \\ \text{sign } p_{i+1}(+\infty) &= (-1)^{i+1}, \\ \text{sign } p_{i+1}(-\infty) &= 1, \end{aligned}$$

e $p_{i+1}(\lambda)$ cambia de signo en cada un dos intervalos $[\lambda_1^{(i)}, \infty)$, $(-\infty, \lambda_i^{(i)}]$, $[\lambda_{k+1}^{(i)}, \lambda_k^{(i)}]$, $k = 1, \dots, n-1$. As raíces $\lambda_k^{(i+1)}$ de p_{i+1} son tamén reais e simples e separan as raíces $\lambda_k^{(i)}$ de p_i :

$$\lambda_1^{(i+1)} > \lambda_1^{(i)} > \lambda_2^{(i+1)} > \lambda_2^{(i)} > \dots > \lambda_i^{(i)} > \lambda_{i+1}^{(i+1)}.$$

□

A sucesión de polinomios construídos no teorema anterior ten unha serie de propiedades que a convirten nunha sucesión de Sturm, conceptos que estudiaremos a continuación.

Definición 4.3. Un conxunto ordenado de polinomios reais

$$p_0(\lambda), p_1(\lambda), p_2(\lambda), \dots, p_n(\lambda) = p(\lambda)$$

constitúe unha sucesión de Sturm do polinomio $p(\lambda)$ se verifica as seguintes propiedades:

- (i) Todas as raíces reais de $p(\lambda) = p_n(\lambda)$ son simples.
- (ii) $\text{sign } p_{n-1}(\xi) = -\text{sign } p'_n(\xi)$, se ξ é unha raíz real de $p_n(\lambda)$.
- (iii) Para $i = 1, 2, \dots, n - 1$,

$$p_{i+1}(\xi)p_{i-1}(\xi) < 0$$

se ξ é unha raíz real de $p_i(\lambda)$.

- (iv) O primeiro polinomio $p_0(\lambda)$ non ten raíces reais.

Véxase agora un teorema para as sucesións de Sturm que se necesitará posteriormente.

Teorema 4.4. *O número de raíces reais do polinomio $p(\lambda) = p_n(\lambda)$ no intervalo $a \leq \lambda < b$ é igual a $\omega(b) - \omega(a)$, onde $\omega(\lambda)$ denótase como o número de cambios de signo dunha sucesión de Sturm*

$$p_0(\lambda), p_1(\lambda), \dots, p_n(\lambda)$$

no punto λ e ignorando os valores nulos:

$$\omega(\lambda) = V(p_0(\lambda), p_1(\lambda), \dots, p_n(\lambda)).$$

Demostración. Examinemos como a perturbación do valor a afecta ao número de cambios de signo $\omega(a)$ da secuencia

$$p_0(a), p_1(a), \dots, p_n(a).$$

Se o valor a non é unha raíz do polinomio $p_i(\lambda)$, con $i = 0, 1, \dots, n$, entón non se producirá ningún cambio. No caso de que a sexa unha raíz de $p_i(\lambda)$, consideraremos os dous casos $i < n$ e $i = n$.

No primeiro caso, $i > 0$ por (iv) da definición 4.3 e $p_{i+1}(a) \neq 0$, $p_{i-1}(a) \neq 0$ por (iii) da definición 4.3. Se $p_i(\lambda)$ cambia de signo en $\lambda = a$, entón para unha perturbación suficientemente pequena, h , o signo do polinomio $p_j(a)$, para $j = i - 1, i, i + 1$, mostra o seu comportamento do xeito ilustrado nunha das seguintes catro táboas:

	$a-h$	a	$a+h$
$i-1$	-	-	-
i	-	0	+
$i+1$	+	+	+

	$a-h$	a	$a+h$
$i-1$	+	+	+
i	-	0	+
$i+1$	-	-	-

	$a-h$	a	$a+h$
$i-1$	-	-	-
i	+	0	-
$i+1$	+	+	+

	$a-h$	a	$a+h$
$i-1$	+	+	+
i	+	0	-
$i+1$	-	-	-

En cada un dos casos tense $\omega(a-h) = \omega(a) = \omega(a+h)$, é dicir, o número de cambios de signo segue sendo o mesmo. Isto tamén é certo se $p_i(\lambda)$ non cambia de signo en $\lambda = a$.

No segundo caso, podemos concluir por (ii) da definición (4.3) os seguintes patróns de signos:

i	$a-h$	a	$a+h$
0	-	0	+
1	-	-	-

i	$a-h$	a	$a+h$
0	+	0	-
1	+	+	+

En cada un deles tense, $\omega(a-h) = \omega(a) = \omega(a+h) - 1$, é dicir, cámbiase de signo ao pasar a través dunha raíz de $p_n(\lambda) \equiv p(\lambda)$.

Para $a < b$ e $h > 0$ suficientemente pequena,

$$\omega(b) - \omega(a) = \omega(b-h) - \omega(a-h)$$

indica o número de raíces de $p(\lambda)$ no intervalo $a-h < \lambda < b-h$. Xa que $h > 0$ pode ser escollido arbitrariamente pequeno, a diferenza anterior indica tamén o número de raíces no intervalo $a \leq \lambda < b$. □

Como consecuencia do teorema podemos probar o seguinte resultado fundamental de Givens.

Teorema 4.5 (Givens). *Sexa A unha matriz tridiagonal simétrica da forma (4.1) con b_1, b_2, \dots, b_{n-1} da forma (4.1) distintos de cero. Entón a sucesión $p_0(\lambda), p_1(\lambda), \dots, p_n(\lambda)$ do teorema 4.1 é unha sucesión de Sturm relativa ao polinomio característico de A , $p_n(\lambda) = \det(A - \lambda I)$.*

Demostración. En efecto, comprobemos as condicións da sucesión de Sturm.

1. Xa probamos que $p_n(\lambda)$ ten todas as súas raíces reais (é o polinomio característico dunha matriz simétrica, polo tanto todos os seus autovalores son reais) e no teorema 4.2 probamos que todas son simples.

2. Se $\lambda_1 > \lambda_2 > \dots > \lambda_n$ son as raíces de $p_n(\lambda)$ tense de (4.3), para $k = 1, 2, \dots, n$,

$$\begin{aligned} \text{sign } p_{n-1}(\lambda_k) &= (-1)^{n+k}, \\ \text{sign } p'_n(\lambda_k) &= (-1)^{n+k+1} = -\text{sign } p_{n-1}(\lambda_k). \end{aligned}$$

3. Sexa α unha raíz de $p_i(\lambda)$, $i = 1, 2, \dots, n-1$. Entón pola relación de recurrencia:

$$p_{i+1}(\alpha) = (a_{i+1} - \alpha)p_i(\alpha) - b_i^2 p_{i-1}(\alpha) = -b_i^2 p_{i-1}(\alpha).$$

Polas propiedades de separación das raíces dos polinomios $p_i(\lambda)$ tense que α non pode ser raíz de p_{i-1} e p_{i+1} polo que se conclúe

$$p_{i+1}(\alpha)p_{i-1}(\alpha) < 0.$$

4. $p_0 \equiv 1$ non ten raíces reais.

□

Como consecuencia das propiedades especiais da sucesión de polinomios $p_0(\lambda), p_1(\lambda), \dots, p_n(\lambda)$ tense o seguinte corolario.

Corolario 4.6. *O número $\omega(\mu) = V(p_0(\mu), p_1(\mu), \dots, p_n(\mu))$ coincide co número de raíces de $p_n(\lambda)$, autovalores de A , estrictamente menores que μ .*

Demostración. Tendo en conta que

$$p_i(\lambda) = (-1)^i \lambda^i + \dots$$

tense que

$$\begin{aligned} \lim_{\lambda \rightarrow -\infty} p_i(\lambda) &= +\infty \\ \lim_{\lambda \rightarrow +\infty} p_i(\lambda) &= (-1)^i \infty. \end{aligned}$$

Polo tanto,

$$\begin{aligned} \omega(\mu) &= V(+1, +1, \dots, +1) = 0, \text{ para } \mu \leq \lambda_n \\ \omega(\mu) &= V(+1, -1, \dots, (-1)^n) = n, \text{ para } \mu > \lambda_1 \end{aligned}$$

e o resultado dedúcese do teorema 4.4.

□

Corolario 4.7. *Para $i = 1, 2, \dots, n$,*

$$\lambda_i < \mu \text{ se e só se } \omega(\mu) \geq n - i + 1$$

ou ben,

$$\lambda_i \geq \mu \text{ se e só se } \omega(\mu) < n - i + 1.$$

Demostración. Basta ter en conta que $\lambda_n < \lambda_{n-1} < \dots < \lambda_2 < \lambda_1$. \square

Observación 4.8. Dedúcese en particular que a matriz A é definida positiva se, e só se, $\omega(0) = 0$ e $p_n(0) \neq 0$.

A continuación móstrase un teorema que será importante para o algoritmo de bisección de Givens.

Teorema 4.9. *Os autovalores λ_i da matriz A en (4.1) satisfán a seguinte desigualdade*

$$|\lambda_i| \leq \max_{1 \leq j \leq n} |a_j| + |b_j| + |b_{j+1}|, \text{ con } b_1 = b_{n+1} = 0.$$

Demostración. Para a norma máxima $\|x\|_\infty = \max |x_j|$ un ten

$$\|A\|_\infty = \max_{1 \leq j \leq n} |a_j| + |b_j| + |b_{j+1}|, \text{ e de}$$

$Ax = \lambda_i x$, con $x \neq 0$ séguese que

$$|\lambda_i| \|x\|_\infty = \|Ax\|_\infty \leq \|A\|_\infty \|x\|_\infty, \quad \|x\|_\infty \neq 0,$$

e entón $\lambda_i \leq \|A\|_\infty$. \square

4.1. Algoritmo de bisección de Givens

Utilizando adecuadamente os corolarios anteriores é posible atopar os autovalores de A cunha precisión teoricamente arbitraria.

En primeiro lugar, dado que $\rho(A) \leq \|A\|_\infty$ terase que todos os autovalores de A , raíces de $p_n(\lambda) = 0$, pertencen ao intervalo pechado

$$I = [-\|A\|_\infty, \|A\|_\infty]$$

con

$$\|A\|_\infty = \max_{2 \leq i \leq n} \{|a_1| + |b_1|, |a_i| + |b_i| + |b_{i-1}|, |a_n| + |b_{n-1}|\}$$

Outra posibilidade para a acotación é utilizar o teorema de Gerschgorin que nos dí que todos os autovalores de A están nos intervalos pechados da recta real:

$$I_1 = [a_1 - |b_1|, a_1 + |b_1|]$$

$$I_i = [a_i - |b_{i-1}| - |b_i|, a_i + |b_{i-1}| + |b_i|], \quad i = 2, 3, \dots, n-1$$

$$I_n = [a_n - |b_{n-1}|, a_n + |b_{n-1}|]$$

Podemos asegurar entón que os n autovalores de A están no intervalo $[\alpha, \beta]$ en:

$$\alpha = \min_{2 \leq i \leq n-1} \{a_1 - |b_1|, a_i - |b_{i-1}| - |b_i|, a_n - |b_{n-1}|\}$$

$$\beta = \max_{2 \leq i \leq n-1} \{a_1 + |b_1|, a_i + |b_{i-1}| + |b_i|, a_n + |b_{n-1}|\}.$$

O algoritmo de Givens para aproximar o autovalor λ_i , $i = 1, 2, \dots, n$ baséase en aplicar o algoritmo de dicotomía ou de bisección á ecuación $p_n(\lambda) = 0$, coa vantaxe de que a información proporcionada pola sucesión de Sturm permítenos seleccionar os intervalos que conteñen a raíz λ_i . O algoritmo descríbese como sigue, supoñendo

$$\lambda_n < \lambda_{n-1} < \dots < \lambda_i < \dots < \lambda_2 < \lambda_1.$$

Comezáse escollendo un intervalo

$$[a_0, b_0],$$

no cal está contida a raíz λ_i , por exemplo, escollemos $b_0 > \lambda_1$, $a_0 < \lambda_n$. A continuación dividimos este intervalo en dous subintervalos iguais e comprobamos por medio da secuencia de Sturm en cal dos dous subintervalos está λ_i . O subintervalo que conteña a λ_i dividirase outra vez e así sucesivamente. Entón, para $j = 0, 1, 2, \dots$, tense o seguinte

$$\mu_j = (a_j + b_j)/2,$$

$$a_{j+1} := \begin{cases} a_j & \text{se } \omega(\mu_j) \geq n + 1 - i \\ \mu_j & \text{se } \omega(\mu_j) < n + 1 - i \end{cases}$$

$$b_{j+1} := \begin{cases} \mu_j & \text{se } \omega(\mu_j) \geq n + 1 - i \\ b_j & \text{se } \omega(\mu_j) < n + 1 - i \end{cases}$$

Entón, tense que

$$\begin{aligned} [a_{j+1}, b_{j+1}] &\subseteq [a_j, b_j], \\ |a_{j+1} - b_{j+1}| &= |a_j - b_j|/2, \\ \lambda_i &\in [a_{j+1}, b_{j+1}]. \end{aligned}$$

Os valores a_j aumentan mentres que os valores b_j decrecen e así aproxímanse á raíz desexada λ_i . O proceso de converxencia é lineal con taxa de converxencia 0,5. En efecto, para $j = 0, 1, 2, \dots$ tense que

$$|\lambda_i - \mu_j| \leq \frac{1}{2} |b_j - a_j| = \frac{1}{2^{j+1}} |b_0 - a_0|.$$

Se queremos un test de parada que asegure que $|\lambda_i - \mu_j| < \epsilon$ bastará facer j tal que

$$\frac{1}{2^{j+1}} |b_0 - a_0| < \epsilon.$$

Este método para determinar raíces dun polinomio real, as cales son todas reais, é relativamente lento pero moi preciso. Ademais ten a vantaxe de que cada raíz pode ser determinada independentemente das outras.

Observación 4.10. A técnica anterior pode particularizarse para aproximar os autovalores pertencentes a un intervalo dado, para aproximar o autovalor máis pequeno, o autovalor máis grande, etc.

Por outra parte, a fórmula de recurrencia de Givens permite avaliar o polinomio característico da matriz tridiagonal simétrica A en todos os puntos que se desexe. Isto permite utilizar outro método iterativo calquera diferente do de dicotomía como poden ser os métodos de regula falsi, iteración funcional, e incluso Newton-Raphson posto que podemos avaliar por recurrencia $p'_n(\lambda)$:

$$\begin{cases} p'(\lambda) = 0, \\ p'_1(\lambda) = -1, \\ p'_i(\lambda) = -p_{i-1}(\lambda) + (a_i - \lambda)p'_{i-1}(\lambda) - b_{i-1}^2 p'_{i-2}(\lambda), \quad \text{con } i = 2, 3, \dots, n. \end{cases}$$

Un manexo práctico moi común do método de bisección de Givens é para proporcionar unha aproximación do autovalor ou autovalores e, posteriormente, afinar esa aproximación con métodos máis potentes como pode ser, por exemplo, o método da potencia inversa.

4.2. Programación en Matlab

Mostramos a continuación o código en Matlab do algoritmo de bisección de Givens e cos resultados aplicados á matriz tridiagonal calculada no capítulo 2:

$$\begin{pmatrix} 10 & 12,7279 & 0 & 0 \\ 12,7279 & 21,8827 & 3,4281 & 0 \\ 0 & 3,4281 & 2,3825 & 0,4621 \\ 0 & 0 & 0,4621 & 0,7348 \end{pmatrix}.$$

cuxos autovalores son $\lambda_1 = 0,0102$, $\lambda_2 = 0,8431$, $\lambda_3 = 3,8581$ e $\lambda_4 = 30,2887$.

Primeiro calcúlase o intervalo que contén todos os autovalores desta matriz e procédese ao seu cálculo. O código principal fai chamada a unha función $w(p)$ que calcula os cambios de signo nunha sucesión de números $\{p_1, \dots, p_n\}$. Para rematar podemos observar que os autovalores calculados son correctos xa que son os mesmos que os calculados co comando *eig* no apéndice A.

Código Matlab do Método de bisección de Givens.

```
n = 4;
% Dáselle nome aos valores da matriz.
a = zeros(n,1);
b = zeros(n-1,1);
a = [10 21.8827 2.3825 0.7348];
b = [12.7279 3.4281 0.4621];
% Cálculo do intervalo que contén aos autovalores.
c = [a(1)-abs(b(1))];
d = [a(1)+abs(b(1))];
for i = 2:n-1
c = [c a(i)-abs(b(i-1))-abs(b(i))];
d = [d a(i)+abs(b(i-1))+abs(b(i))];
end
c = [c a(n)-abs(b(n-1))];
d = [d a(n)+abs(b(n-1))];
alfa0 = min(c);
beta0 = max(d);
disp(['O intervalo inicial é: ', num2str(alfa0), ', ', num2str(beta0) ''])
vp = zeros(n,1); % Vector para almacenar os autovalores.
for i = 1:n % Número do autovalor que se calcula.
disp(['-----'])
disp(['CÁLCULO DO AUTOVALOR NÚMERO: ', num2str(i)])
disp(['-----'])
alfa = alfa0;
beta = beta0;
% [alpha,beta] é o intervalo inicial no que están todos os autovalores.
```

```

eps = 1.e-10;
while abs(alfa-beta)> eps
    % Cálculo de mu (punto medio de alfa+beta).
    mu = (alfa+beta)/2;
    disp(['0 punto medio é mu = ',num2str(mu)])
    % Cálculo do polinomio p_n.
    p = zeros(n+1,1); % A sucesión comeza en 1 xa que en matlab non se admite p(0).
    p(1) = 1.;
    p(2) = a(1) - mu;
    for k = 2:n
        p(k+1) = (a(k) - mu)*p(k) - p(k-1)*b(k-1)^2 ;
    end
    if w(p) >= n + 1 - i
        beta = mu;
    else
        alfa = mu;
    end
end
disp(['Autovalor número ', num2str(i), ': ',num2str(mu)])
vp(n-i+1) = mu;
end
vp % Dá todos os autovalores ordenados de menor a maior.

```

Este algoritmo precisa tamén o seguinte código que constitúe o cálculo de cambios de signo da sucesión.

```

function var = w(p)
% Cambio de signo da sucesion p_1, ..., p_n.
n = length(p);
s = sign(p(1:n-1).*p(2:n));
vn = 0;
v0 = 0;
for i = 1:n-1
    if s(i) == 0
        v0 = v0 + 1;
    end
    if s(i) < 0
        vn = vn + 1;
    end
end

```

```

    end
  end
  var = vn + v0/2;
end

```

A solución é a seguinte:

O intervalo inicial é: [-2.7279,38.0387]

 CÁLCULO DO AUTOVALOR NÚMERO: 1

 O punto medio é mu = 17.6554
 O punto medio é mu = 27.847
 O punto medio é mu = 32.9429
 O punto medio é mu = 30.395
 O punto medio é mu = 29.121
 O punto medio é mu = 29.758
 O punto medio é mu = 30.0765
 O punto medio é mu = 30.2357
 O punto medio é mu = 30.3153
 O punto medio é mu = 30.2755

...

Autovalor número 1: 30.2887

 CÁLCULO DO AUTOVALOR NÚMERO: 2

 O punto medio é mu = 17.6554
 O punto medio é mu = 7.4638
 O punto medio é mu = 2.3679
 O punto medio é mu = 4.9158
 O punto medio é mu = 3.6419
 O punto medio é mu = 4.2789
 O punto medio é mu = 3.9604
 O punto medio é mu = 3.8011
 O punto medio é mu = 3.8807
 O punto medio é mu = 3.8409

...

Autovalor número 2: 3.8581

CÁLCULO DO AUTOVALOR NÚMERO: 3

0 punto medio é mu = 17.6554
0 punto medio é mu = 7.4638
0 punto medio é mu = 2.3679
0 punto medio é mu = -0.17999
0 punto medio é mu = 1.094
0 punto medio é mu = 0.45699
0 punto medio é mu = 0.77548
0 punto medio é mu = 0.93472
0 punto medio é mu = 0.8551
0 punto medio é mu = 0.81529

...

Autovalor número 3: 0.84315

CÁLCULO DO AUTOVALOR NÚMERO: 4

0 punto medio é mu = 17.6554
0 punto medio é mu = 7.4638
0 punto medio é mu = 2.3679
0 punto medio é mu = -0.17999
0 punto medio é mu = 1.094
0 punto medio é mu = 0.45699
0 punto medio é mu = 0.1385
0 punto medio é mu = -0.020743
0 punto medio é mu = 0.058879
0 punto medio é mu = 0.019068

...

Autovalor número 4: 0.010132

vp =
0.0101
0.8431
3.8581
30.2887

Bibliografía

- [1] Álvarez, L. e Martínez A. (2004). *Métodos Numéricos*, Universidade de Vigo.
- [2] Arbenz, P. (2016). *Lecture Notes on Solving Large Scale Eigenvalue Problems*, ETH Zürich.
- [3] Ciarlet, Philippe G. (1999). *Introducción á análise numérica matricial e á optimización*, tradución de Ballesteros P. e Rubal X. A., Servicio de Publicacións da Universidade de Santiago de Compostela.
- [4] Galántai, A. e Hegedüs, C. J. (2009). *Hyman's method revisited*, Journal of Computational and Applied Mathematics, Vol. 226, pp. 246-258.
- [5] García Argos, C. (2002/2003) *Apuntes de Métodos Numéricos 2º E.T.S.I. Telecomunicación*, Universidad de Málaga.
- [6] Jiménez López V. e Pallarés Ruiz, A. (2007/2008). *Métodos Numéricos*, Universidad de Murcia.
- [7] O'Connor, J. L. (2017). *Ingeniería de los Algoritmos y Métodos Numéricos. Un acercamiento práctico avanzado a la computación científica e ingenieril con MATLAB*, Editorial Círculo Rojo, España.
- [8] Rannacher, R. (2014). *Numerical Linear Algebra*, Institute of Applied Mathematics Heidelberg University.
- [9] Stoer, J. e Burlirsch, R. (1993). *Introduction to Numerical Analysis*, Springer-Verlag, New York.
- [10] Viaño, J. M. (1997). *Lecciones de métodos numéricos 2.: Resolución de ecuaciones numéricas*, Tórculo Edicións, Santiago de Compostela.
- [11] Wilkinson, J. H. e Reinsch, C. (1971). *Linear Algebra*, Springer-Verlag, New York.
- [12] <https://www.cpp.edu/manasab/eigenvalue.pdf>

- [13] <https://www.intmath.com/matrices-determinants/8-applications-eigenvalues-eigenvectors.php>
- [14] https://hmong.es/wiki/Givens_rotation