

4

O algoritmo tipo Jacobi quaterniônico para Matrizes Normais

Diferentemente com o que acontece com a maioria dos algoritmos existentes para o cálculo de autovalores de uma matriz em específico, o método tipo Jacobi quaterniônico foi proposto inicialmente para matrizes anti-simétricas. Em 1995, (14) propôs uma extensão do algoritmo a matrizes simétricas.

Com o caso simétrico existente, tornou-se evidente a possibilidade de generalização para matrizes reais normais, baseando-se em uma série de propriedades, dentre as quais podemos citar: matrizes normais são unitariamente diagonalizáveis, além de possuírem uma série de propriedades ligadas ao seu espectro.

Devido ao fato de que a normalidade de matrizes aparece nas mais diferentes formas, o algoritmo torna-se aplicável a vasta quantidade de casos em que a mesma se apresenta.

4.1

Quatérnios para a determinação do espectro de Matrizes Simétricas

Uma importante propriedade das matrizes simétricas está descrita a seguir:

Proposição 4.1.1 *As seguintes afirmações são equivalentes:*

- 1) S é simétrica.
- 2) $S \in (\mathbb{R} \otimes \mathbb{R}) \oplus (\mathcal{P} \otimes \mathcal{P})$.
- 3) Podemos decompor a matriz S na forma $S = c1 \otimes 1 + p \otimes i + q \otimes j + r \otimes k$, onde $c \in \mathbb{R}$ e p, q , e r são quatérnios puros.

Considerando a matriz $\widehat{S} = S - cI + p \otimes i + q \otimes j + r \otimes k$, devemos obter uma matriz de rotação $R = x \otimes y \in \mathcal{SO}_4$ que a diagonalize, ou seja:

$$R\widehat{S}R^T = xp\bar{x} \otimes yi\bar{y} + xq\bar{x} \otimes yj\bar{y} + xr\bar{x} \otimes yk\bar{y} \quad (4-1)$$

Para que a matriz R diagonalize \widehat{S} , então esta deve diagonalizar a tupla de vetores $\{p, q, r\}$ ao longo dos eixos ortogonais $\{i, j, \pm k\}$. O problema está no fato de que tal tupla $\{p, q, r\}$ é geralmente oblíqua, o que torna essa tarefa impossível. Uma solução consiste em decompor essa rotação em duas rotações independentes, ao invés de apenas uma.

Considerando a decomposição de \widehat{S} da forma

$$a_1 \otimes b_1 + a_2 \otimes b_2 + a_3 \otimes b_3 \in \mathcal{P} \times \mathcal{P} \quad (4-2)$$

com a propriedade de que as tuplas $\{a_1, a_2, a_3\}$ e $\{b_1, b_2, b_3\}$ tem seus elementos ortogonais entre si. Utilizando o isomorfismo definido em (3-18) temos

$$\begin{aligned} \psi(\widehat{S}) &= \psi(p \otimes i + q \otimes j + r \otimes k) = pe_1^T + qe_2^T + re_3^T \\ &= \begin{bmatrix} p_1 & q_1 & r_1 \\ p_2 & q_2 & r_2 \\ p_3 & q_3 & r_3 \end{bmatrix} = \sigma_1 u_1 v_1 + \sigma_2 u_2 v_2 + \sigma_3 u_3 v_3 = \\ &= \psi(\sigma_1 u_1 \otimes v_1 + \sigma_2 u_2 \otimes v_2 + \sigma_3 u_3 \otimes v_3) \end{aligned} \quad (4-3)$$

Obtemos, então, a matriz $R = x \otimes y$ de forma que um os vetores singulares à direita e outro à esquerda sejam rotacionados em direção ao eixo dos i . Com isso, os demais vetores serão movidos para o plano- jk . Agora basta um vetor de cada tupla na direção do eixo j ou do eixo k . Com isso, os dois vetores restantes, um de cada tupla, já estão alinhados na outra direção diferente da escolhida. Considerando a matriz Q como a responsável pelo alinhamento de um dos dois vetores escolhidos de cada tupla no segundo processo, podemos dizer que a matriz $QR\widehat{S}R^TQ^T$ está na forma diagonal.

Resumindo o parágrafo anterior, uma alternativa para obter os autovalores da matriz simétrica dar-se-á de acordo com a seguinte alternativa:

R rotaciona u_1, v_1 no eixo i

Q rotaciona u_1, v_1 no eixo i e u_2, v_2 no eixo j

Em face do exposto, dados $R, S \in \mathcal{SO}_4$ podemos escrever

$$QR(p \otimes i + q \otimes j + r \otimes k)R^TQ^T = \begin{cases} \sigma_1 i \otimes i + \sigma_2 j \otimes j + \sigma_3 k \otimes k \\ \sigma_1 i \otimes i + \sigma_2 j \otimes j - \sigma_3 k \otimes k \end{cases}$$

O primeiro caso é válido quando os vetores que compõem as duas tuplas envolvidas tenham sido projetadas nos eixos $\{i, j, k\}$. O segundo caso é aplicado aos eixos $\{i, j, -k\}$. Dessa forma, devemos considerar dois casos:

Caso 1: Os vetores foram alinhados na direção dos eixos $\{i, j, k\}$:

Então escreveremos

$$\sigma_1 i \otimes i + \sigma_2 j \otimes j + \sigma_3 k \otimes k = \begin{bmatrix} \sigma_1 + \sigma_2 + \sigma_3 & 0 & 0 & 0 \\ 0 & \sigma_1 - \sigma_2 - \sigma_3 & 0 & 0 \\ 0 & 0 & -\sigma_1 + \sigma_2 - \sigma_3 & 0 \\ 0 & 0 & 0 & -\sigma_1 - \sigma_2 + \sigma_3 \end{bmatrix}$$

Caso 2: Os vetores que compõem as duas tuplas foram alinhados na direção dos eixos $\{i, j, -k\}$:

Donde se obtém:

$$\sigma_1 i \otimes i + \sigma_2 j \otimes j + \sigma_3 k \otimes k = \begin{bmatrix} \sigma_1 + \sigma_2 - \sigma_3 & 0 & 0 & 0 \\ 0 & \sigma_1 - \sigma_2 + \sigma_3 & 0 & 0 \\ 0 & 0 & -\sigma_1 + \sigma_2 + \sigma_3 & 0 \\ 0 & 0 & 0 & -\sigma_1 - \sigma_2 - \sigma_3 \end{bmatrix}$$

4.2

Aplicação do método tipo Jacobi quaterniônico a uma matriz simétrica

A implementação do algoritmo tipo Jacobi quaterniônico para o caso simétrico é mais complexo que o do caso anti-simétrico; a decomposição da matriz em seus valores singulares, aliado a obtenção das tuplas de vetores ortogonais contribuem para um aumento na complexidade da codificação e execução do algoritmo computacional.

Abaixo, uma tabela de execução do método qJ para a matriz simétrica de ordem 10, cujos elementos estão localizados no intervalo $[-10, +10]$:

| | Matriz Simétrica E | | |
|-----------------|----------------------|-------|--------------|
| | Iterações | Tempo | Norma Obtida |
| $\leq 10^{-1}$ | 4 | 0.00" | 7.58e-1 |
| $\leq 10^{-7}$ | 6 | 0.02" | 3.245e-9 |
| $\leq 10^{-8}$ | 6 | 0.02" | 3.245e-9 |
| $\leq 10^{-9}$ | 6 | 0.02" | 3.245e-9 |
| $\leq 10^{-10}$ | 7 | 0.04" | 7.849e-11 |
| $\leq 10^{-11}$ | 7 | 0.04" | 7.849e-11 |

Tabela 4.1: Execução do algoritmo tipo Jacobi quaterniônico para uma matriz de ordem 10 cujos elementos encontram-se no intervalo $[-10, +10]$

A tabela abaixo faz uma comparação entre os valores encontrados pelos métodos comumente utilizados para a obtenção de autovalores e o software Maple 9.5 ¹; tal comparação tem como objetivo analisar os resultados obtidos através de um software comercial e os algoritmos em suas versões clássicas.

Eis a tabela comparativa abaixo:

¹desenvolvido pela Waterloo Software Incorporation©

| Maple | qJ | QR Householder |
|-----------|-----------|----------------|
| -15.75787 | -15.75778 | -15.68972 |
| -14.60071 | -14.55803 | -14.55298 |
| -10.64974 | -10.64971 | -10.57320 |
| -9.02126 | -9.10231 | -9.020995 |
| -8.59356 | -8.65724 | -8.90497 |
| 0.54855 | 0.55347 | 0.67236 |
| 3.65580 | 3.81749 | 3.95822 |
| 5.98235 | 5.99649 | 5.74588 |
| 8.01080 | 8.64839 | 8.84774 |
| 40.42564 | 40.69230 | 40.94872 |

Tabela 4.2: Tabela dos valores obtidos no cálculo dos autovalores da matriz simétrica de ordem 10

4.3

O caso da Matrizes Normais...

Uma matriz real A é dita *normal* se satisfaz a condição

$$AA^T = A^T A \quad (4-4)$$

onde A^T é a transposta da matriz.

A seguir algumas considerações sobre matrizes normais:

Definição 4.3.1 Dizemos que a matriz A é equivalente a uma matriz B (notação $A \approx B$) se existe uma matriz ortogonal P tal que $B = P^T A P$.

Lema 4.3.2 Sejam A uma matriz normal e uma matriz B tal que $A \approx B$. Então B também é normal.

Teorema 4.3.3 (Teorema Espectral para Matrizes Normais) Toda matriz normal é equivalente a uma matriz da forma $A_1 \oplus A_2 \oplus A_3 \oplus \dots \oplus A_k$, onde cada A_j é de ordem 1×1 ou possui a forma

$$A_i = \begin{bmatrix} a & b \\ -b & a \end{bmatrix} \quad (4-5)$$

com $b > 0$. A menos de reordenação dos A_j tal forma é única.

O caso 2×2 do teorema é facilmente demonstrado. No caso geral, toda matriz é equivalente a uma matriz da forma

$$A_i = \begin{bmatrix} A & X \\ \circ & N \end{bmatrix},$$

com A uma matriz de ordem 1×1 ou 2×2 , donde segue a redução da forma $A_1 \oplus A_2 \oplus \dots \oplus A_k$ acima.

Matrizes reais normais são generalizações de matrizes simétricas. São exemplos importantes de matrizes normais: matrizes simétricas e anti-simétricas, matrizes ortogonais, etc.

Para aplicar o algoritmo tipo Jacobi quaterniônico para matrizes normais devemos, primeiramente, escrevê-la como uma soma de uma matriz simétrica e uma matriz anti-simétrica. Sabemos que, em uma matriz normal, a parte simétrica e anti-simétrica comutam entre si. Matematicamente, decompondo a matriz normal A em sua forma real simétrica, definida por $S = \frac{A+A^T}{2}$ e anti-simétrica, $K = \frac{A-A^T}{2}$, podemos escrever:

$$A \text{ é normal} \Leftrightarrow A = K \cdot S = S \cdot K.$$

Diagonalizando a matriz simétrica S através do algoritmo tipo Jacobi quaterniônico nos permite escrever:

$$UAU^T = UKU^T + D_S, \text{ onde } D_S \text{ está na forma diagonal.}$$

Denotando $B = UKU^T$, temos que tal matriz já é bloco-diagonal.

Devido a comutatividade entre as matrizes simétrica e anti-simétrica no caso das matrizes normais, os únicos elementos não nulos fora da diagonal de B correspondem aos autovalores correspondentes em D_S .

Dessa forma, o primeiro passo do algoritmo qJ aplicado a matrizes reais normais consiste em diagonalizar S , ou seja, $USU^T = aI \oplus bI \oplus \dots \oplus kI$, onde a, b, \dots, k são reais. De $S \cdot K = K \cdot S$, segue-se que, se $B_a \oplus B_b \oplus \dots \oplus B_k$ já são blocos-diagonais da forma (4.5).

Em nossos experimentos, obtemos a priori a decomposição da matriz da normal em sua forma simétrica e anti-simétrica, em seguida, executamos o algoritmo tipo Jacobi quaterniônico para o caso simétrico e para o caso anti-simétrico, obtendo a norma ao final de cada um dos processos.

Foram utilizados três tipos de matrizes normais de ordem 40×40 : a primeira matriz de teste possui todos os seus autovalores reais; o segundo tipo de matriz possui metade dos autovalores reais e metade dos autovalores complexos; no terceiro caso, a matriz possui todos os seus autovalores complexos.

Eis a tabela de execução do algoritmo para os três casos observados:

| Norma | Caso 1 | | Caso 2 | | Caso 3 | |
|-----------------|-----------|--------|-----------|--------|-----------|--------|
| | Iterações | Tempo | Iterações | Tempo | Iterações | Tempo |
| $\leq 10^{-1}$ | 9 | 9.00" | 10 | 9.15" | 10 | 9.52" |
| $\leq 10^{-7}$ | 14 | 11.23" | 15 | 12.74" | 16 | 14.21" |
| $\leq 10^{-8}$ | 14 | 11.23" | 15 | 12.74" | 16 | 14.21" |
| $\leq 10^{-9}$ | 15 | 12.31" | 16 | 13.95" | 16 | 14.21" |
| $\leq 10^{-10}$ | 16 | 13.92" | 16 | 13.95" | 17 | 15.02" |

Nas figuras abaixo, temos a evolução da norma de cada elemento de uma matriz 4×4 normal ao longo do processo:

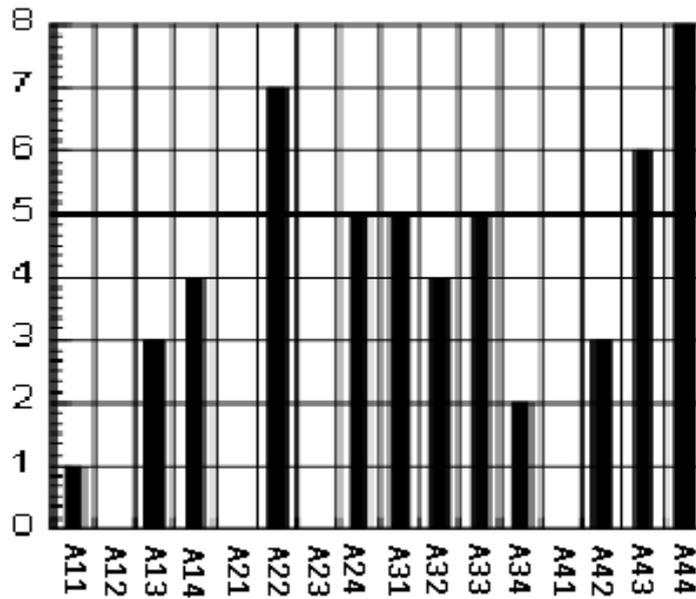


Figura 4.1: Norma dos elementos de uma matriz normal antes de primeira iteração do algoritmo

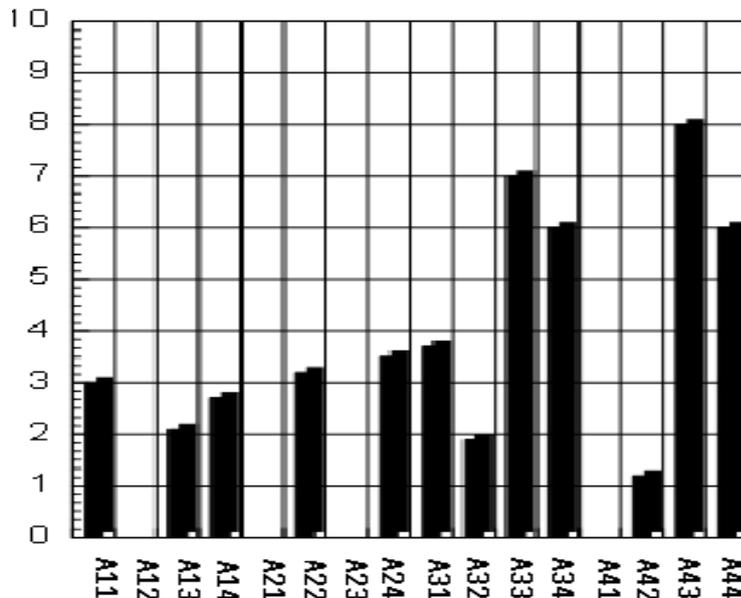


Figura 4.2: Norma dos elementos de uma matriz normal após a terceira iteração do algoritmo

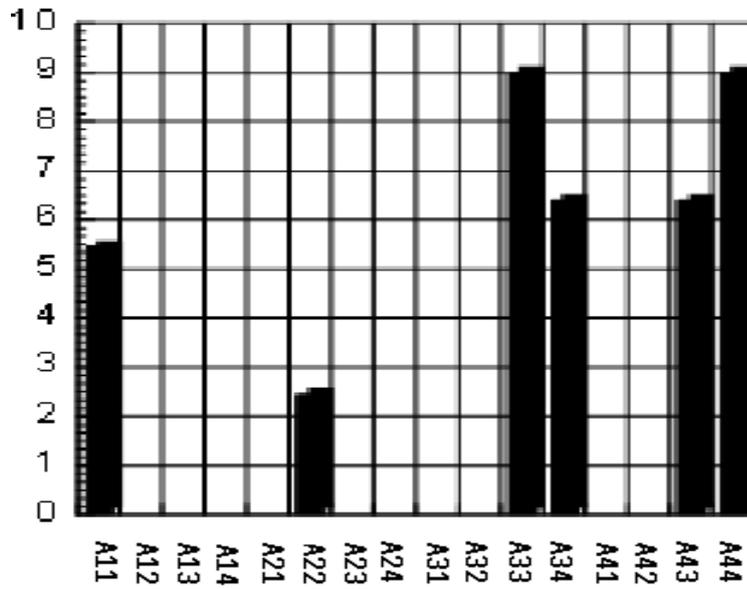


Figura 4.3: Norma dos elementos de uma matriz normal após dez iterações do algoritmo