Numerical Linear Algebra Assignment 9

Exercise 1. (10 points)

Consider the simultaneous iteration:

Algorithm 1: Simultaneous iteration
Pick $\mathbf{Q}_n^{(0)} \in \mathbb{C}^{m \times n}$ with orthonormal columns
for $k = 1, 2, 3, \ldots,$
$\quad \mathbf{Z}^{(k)} = \mathbf{A}\mathbf{Q}_n^{(k-1)}$
$\quad \mathbf{Q}_n^{(k)}\mathbf{R}_n^{(k)} = \mathbf{Z}^{(k)} \qquad$ (QR factorization)
end

Assume $\mathbf{A} = \mathbf{S}\boldsymbol{\Lambda}\mathbf{S}^{-1}$ is diagonalizable with $\boldsymbol{\Lambda} = \text{diag}\{\lambda_1, \lambda_2, \cdots, \lambda_m\}$ and

$$|\lambda_1| \geq \cdots \geq |\lambda_n| > |\lambda_{n+1}| \geq \cdots \geq |\lambda_m|.$$

Assume $\mathbf{X}_n^{(0)} := \begin{bmatrix} \mathbf{I}_n & \mathbf{0} \end{bmatrix} \mathbf{S}^{-1}\mathbf{Q}_n^{(0)}$ has full rank. Prove that

$$\text{span}\{\mathbf{Q}_n^{(k)}\} = \text{span}\{\mathbf{A}^k \mathbf{Q}_n^{(0)}\}.$$

Exercise 2. (Zhihao Cao, 10 points)

Let $\mathbf{A}_1 = \begin{bmatrix} a & b \\ \varepsilon & c \end{bmatrix} \in \mathbb{R}^{2 \times 2}$. Let $\mathbf{A}_2 = \mathbf{R}_1\mathbf{Q}_1 + c\mathbf{I} = \begin{bmatrix} a_{11} & a_{12} \\ a_{21} & a_{22} \end{bmatrix}$, where $\mathbf{Q}_1\mathbf{R}_1 = \mathbf{A}_1 - c\mathbf{I}$ is a QR factorization of $\mathbf{A}_1 - c\mathbf{I}$. Prove:

(a) if a and c are not close (i.e., $\exists \delta > 0$, $|a - c| > \delta$), then $a_{21} = \mathcal{O}(\varepsilon^2)$;

(b) if further $b = \varepsilon$ (i.e., $\mathbf{A}_1$ is symmetric), then $a_{21} = \mathcal{O}(\varepsilon^3)$.

Exercise 3. (10 points)

Assume that $\mathbf{A} \in \mathbb{R}^{n \times n}$ is nonsingular. Let $\mathbf{A} = \mathbf{QR}$ be a QR factorization. Prove that if $\mathbf{A}$ is symmetric and has bandwidth $2p + 1$ (i.e., $a_{ij} = 0$ for $|i - j| > p$), then $\mathbf{B} = \mathbf{RQ}$ is also symmetric and has bandwidth $2p + 1$.

Compulsory requirement for programming: Use Matlab's `publish` to save all your code, comments, and results to a PDF file. You must use the programming format files: example_format.zip.

Programming 1. (10 points)

Write a function `[Q,H] = myhess(A)` that reduces a complex square matrix to upper Hessenberg form by unitary similarity transformations, i.e., $\mathbf{H} = \mathbf{Q}^*\mathbf{A}\mathbf{Q}$. Your program should use only elementary Matlab operations – not the function `hess`, for example. Apply your program to `A = randn(5) + 1i*randn(5)`.

Programming 2. (TreBau Exercise 29.1, 10 points)