Master in Foundations of Data Science — 2020-2021

NUMERICAL LINEAR ALGEBRA

Exercises from previous exams on *algorithms for eigenvalues and eigenproblems* and on *iterative methods for linear equation solving.*

1. Let $A \in \mathbb{C}^{n\times n}$ be a matrix whose eigenvalues satisfy the strict inequalities

$$|\lambda_1| > |\lambda_2| > \cdots > |\lambda_n|.$$

(1) Which iterative algorithm would you apply to compute the eigenvalue λ_1 of largest absolute value? Give an estimate for its rate of convergence.
(2) How would you proceed to compute the eigenvalue λ_n of smallest absolute value? Similarly as before, give an estimate for the rate of convergence of such an iterative method.

2. Consider the matrix

$$A = \begin{pmatrix} 6.45 & -4.17 \\ 1.79 & -0.76 \end{pmatrix}.$$

(1) For an starting vector $x_0 \in \mathbb{R}^2$, compute the corresponding *iterative scheme* for the power method, aimed to approximate the largest eigenvalue and its corresponding eigenvector.
(2) Choose the starting vector $x_0 = (1, 0)^T$. Knowing that the largest eigenvalue of A is about 10 times bigger than the other one, how many iterations of this scheme are necessary to compute 10 decimal digits of the coordinates of the solution? And how many if we want to compute 30 decimal digits?

3. The *Schur decomposition* of a matrix $A \in \mathbb{C}^{n\times n}$ is its factorization as

$$A = U\,T\,U^T$$

where Q is a unitary matrix and R is upper triangular.

(1) Given A, which algorithm would you apply to compute its Schur decomposition?
(2) How can you compute the eigenvalues and the eigenvectors of A from its Schur decomposition?

4. The *real Schur form* of a matrix $A \in \mathbb{R}^{n\times n}$ is its factorization as

$$A = Q\,R\,Q^T$$

where Q is an orthogonal matrix and R is block upper triangular, with diagonal blocks of size of 1×1 and 2×2.

(1) Given A, which algorithm would you apply to compute its real Schur decomposition?
(2) How can you compute the eigenvalues and the eigenvectors of A using its real Schur form?

4. A matrix $H = (h_{i,j})_{i,j} \in \mathbb{R}^{n\times n}$ is *upper Hessenberg* if all its coefficients below the lower secondary diagonal are zero, that is, if $h_{i,j} = 0$ whenever $i \geq j + 2$.

(1) Explain how a matrix $A \in \mathbb{R}^{n\times n}$ can be reduced to upper Hessenberg form via an orthogonal similarity, that is, how to compute an orthogonal matrix $Q \in \mathbb{R}^{n\times n}$ such that $Q\,A\,Q^T$ is upper Hessenberg.
(2) Show that the QR iteration preserves the Hessenberg form.

5. Let

$$A = \begin{pmatrix} 1 & -1 \\ -1 & 1 \end{pmatrix} \quad \text{and} \quad b = \begin{pmatrix} 1 \\ -2 \end{pmatrix}$$

(1) For this matrix and vector, compute the iterative scheme given by (a) the Jacobi method, (b) the Gauss-Seidel method, and (c) the SOR(ω) method for a parameter $\omega \in \mathbb{R}$.
(2) Using the criterium based on the spectral radius, check if you can guarantee if these Jacobi and Gauss-Seidel schemes converge for any choice of initial vector $x_0 \in \mathbb{R}^2$.
(3) Choosing the starting vector $x_0 = (0, 0)^T$, how many iterations of each of these schemes are necessary to compute 30 decimal digits of the solution? And how many if we want to compute 100 decimal digits?