MATRIX THEORY: HOMEWORK 4 (v1), 23 November 2020

In Homework 3, you have seen the importance of the *minimal polynomial* of a matrix $A \in \mathbb{C}^{n\times n}$ for the study of the boundedness of the trajectories of the associated dynamical system:

$$\dot{x}(t) = Ax(t), \quad x(t) \in \mathbb{C}^n, \quad t \geq 0. \tag{1}$$

The minimal polynomial was there defined as $p_A(\lambda) = \prod_{i=1}^{d}(\lambda - \lambda_i)^{m_i}$, where $\lambda_1, \ldots, \lambda_d$ are the distinct eigenvalues of A and m_i is the size of the largest Jordan block associated to λ_i in the Jordan decomposition of A.

In this work, we will see how the minimal polynomial of A can be computed without computing explicitly the Jordan decomposition of A. Therefore, we will rely on the *Smith normal form* of $\lambda I - A$. We remind here the main result related to the Smith normal form: every polynomial matrix $P(\lambda) \in \mathbb{C}^{n\times n}[\lambda]$ can be decomposed as

$$P(\lambda) = M(\lambda)D(\lambda)N(\lambda), \tag{2}$$

where $M(\lambda), N(\lambda) \in \mathbb{C}^{n\times n}[\lambda]$ are unimodular and $D(\lambda) \in \mathbb{C}^{n\times n}[\lambda]$ is diagonal with monic[1] diagonal elements $d_1(\lambda), \ldots, d_n(\lambda)$ satisfying that $d_i(\lambda)$ divides $d_{i+1}(\lambda)$ for all $i \in \{1, \ldots, n-1\}$.

Exercise A: Minimal polynomial and Smith normal form

Let $P \in \mathbb{C}^{n\times n}[\lambda]$ be a polynomial matrix and let $k \in \{1, \ldots, n\}$. A *k-minor* of $P(\lambda)$ is the determinant of a sub-matrix of $P(\lambda)$, obtained from $P(\lambda)$ by removing $n-k$ of its rows and $n-k$ of its columns. We let $\delta_k(P(\lambda))$ be the monic greatest common divisor (gcd) of all k-minors of $P(\lambda)$. Note that $\delta_k(P(\lambda)) \in \mathbb{C}[\lambda]$, since each k-minor belongs to $\mathbb{C}[\lambda]$.

(A1) Let $P(\lambda), Q(\lambda) \in \mathbb{C}^{n\times n}[\lambda]$ and assume that there are unimodular matrices $M(\lambda), N(\lambda) \in \mathbb{C}^{n\times n}[\lambda]$ such that $Q(\lambda) = M(\lambda)P(\lambda)N(\lambda)$. Using Theorem 1.5 from the lecture notes, show that, for all $k \in \{1, \ldots, n\}$, $\delta_k(P(\lambda)) = \delta_k(Q(\lambda))$.

Hint: Show first that $\delta_k(P(\lambda))$ divides $\delta_k(Q(\lambda))$, then swap the roles of $P(\lambda)$ and $Q(\lambda)$, using the invertibility of $M(\lambda)$ and $N(\lambda)$.

(A2) Using A1, show that, for all $k \in \{1, \ldots, n\}$, $\delta_k(P(\lambda)) = \prod_{i=1}^{k} d_i(\lambda)$, where $d_i(\lambda)$ are the diagonal entries of $D(\lambda)$ and $M(\lambda)D(\lambda)N(\lambda)$ is a Smith decomposition (2) of $P(\lambda)$.

Deduce that the diagonal entries $d_1(\lambda), \ldots, d_n(\lambda)$ of the diagonal matrix $D(\lambda)$ appearing in any Smith decomposition (2) of $P(\lambda)$ are unique, and depend only on the sequence $\delta_1(P(\lambda)), \ldots, \delta_n(P(\lambda))$. The diagonal entries $d_1(\lambda), \ldots, d_n(\lambda)$ of $D(\lambda)$ will be called the *elementary polynomials* of $P(\lambda)$.

Finally, deduce that there are unimodular matrices $M(\lambda), N(\lambda) \in \mathbb{C}^{n\times n}[\lambda]$ such that $Q(\lambda) = M(\lambda)P(\lambda)N(\lambda)$ if and only if $\delta_k(P(\lambda)) = \delta_k(Q(\lambda))$ for all $k \in \{1, \ldots, n\}$.

[1] A polynomial is *monic* if it is identically zero or if the coefficient of its monomial with highest degree is equal to one.

(A3) Let $A \in \mathbb{C}^{n\times n}$ be a Jordan block with eigenvalue λ_1. Using A2, show that the elementary polynomials of $\lambda I - A$ are equal to: $d_i(\lambda) = 1$, for $i = 1, \ldots, n-1$, and $d_n(\lambda) = (\lambda - \lambda_1)^n$.

(A4) Let $J_1 \in \mathbb{C}^{n_1\times n_1}$ and $J_2 \in \mathbb{C}^{n_2\times n_2}$ be two Jordan blocks with respective eigenvalue λ_i and size n_i, $i = 1, 2$. Let $A \in \mathbb{C}^{n\times n}$, with $n = n_1 + n_2$, be the Jordan matrix consisting of the two Jordan blocks J_1 and J_2. Using A2 and A3, show that $\lambda I - A$ can be reduced to the form

$$M(\lambda)(\lambda I - A)N(\lambda) = \operatorname{diag}\{\underbrace{1, \ldots, 1}_{n_1-1\text{ times}}, (\lambda - \lambda_1)^{n_1}, \underbrace{1, \ldots, 1}_{n_2-1\text{ times}}, (\lambda - \lambda_2)^{n_2}\},$$

using unimodular matrices $M(\lambda), N(\lambda) \in \mathbb{C}^{n\times n}[\lambda]$.

From the above and A2, deduce the elementary polynomials of $\lambda I - A$. Also, compare with the minimal polynomial of A. Discuss both cases $\lambda_1 \neq \lambda_2$ and $\lambda_1 = \lambda_2$.

(A5) Let $A \in \mathbb{C}^{n\times n}$. Using A2, A3, A4 and the Jordan decomposition of A, show that there are unimodular matrices $M(\lambda), N(\lambda) \in \mathbb{C}^{n\times n}[\lambda]$ such that $\lambda I - A = M(\lambda)E(\lambda)N(\lambda)$, where $E(\lambda)$ is diagonal with diagonal elements being either 1's or polynomials of the form $(\lambda - \lambda_i)^{n_i}$, where λ_i are the eigenvalues of A (possibly appearing multiple times) and n_i is the size of the corresponding Jordan block in the Jordan decomposition of A.

Let $d_1(\lambda), \ldots, d_n(\lambda)$ be the elementary polynomials of $\lambda I - A$. Deduce that $d_n(\lambda)$ is the minimal polynomial of A.

Exercise B: Implementation

(B1) In a *few* sentences, explain why the computation of the minimal polynomial, or the study of the boundedness of trajectories of linear dynamical systems, using the Jordan decomposition is *not well adapted* from the algorithmic point of view. In particular, focus on the numerical accuracy of the obtained result.

As we have seen in the course (Algorithm 6.1 of the lecture notes), the Smith normal form of a polynomial matrix $P(\lambda) \in \mathbb{C}^{n\times n}[\lambda]$ can be computed *exactly* with numerical methods. On the contrary, the roots of a polynomial $p(\lambda) \in \mathbb{C}[\lambda]$ are in general not computable with numerical methods. However, there are ways to decide algorithmically whether a polynomial has all its roots in the left-hand plane and only simple roots on the imaginary axis; we think for instance to techniques inspired from the *Routh–Hurwitz stability criterion.*[2]

(B2) For the following three matrices, compute the Smith normal form of $\lambda I - A_i$. Deduce the minimal polynomial of A_i. From the minimal polynomial, discuss the boundedness of the trajectories of the associated linear dynamical system. For this, you are not required to use the special techniques mentioned above; you can merely provide a factorization of the minimal

[2]This is merely for information, that there are ways to decide *exactly* and algorithmically whether the trajectories of a given linear dynamical system are bounded or not. However, the implementations of such methods is beyond the scope of this homework.

polynomial, which is not to difficult to guess from the eigenvalues of the matrices.

$$A_1 = \begin{bmatrix} 10.5 & -9.5 & -6.5 & 4.5 \\ 13.0 & -12.0 & -7.5 & 5.5 \\ 9.0 & -6.0 & -6.0 & 2.0 \\ 13.0 & -10.0 & -7.5 & 3.5 \end{bmatrix}, \ A_2 = \begin{bmatrix} 14.5 & -13.5 & -8.5 & 6.5 \\ 16.0 & -15.0 & -9.0 & 7.0 \\ 13.0 & -10.0 & -8.0 & 4.0 \\ 16.0 & -13.0 & -9.0 & 5.0 \end{bmatrix},$$

$$A_3 = \begin{bmatrix} 16.0 & -15.0 & -10.0 & 8.0 \\ 18.0 & -17.0 & -11.0 & 9.0 \\ 14.0 & -11.0 & -9.0 & 5.0 \\ 18.0 & -15.0 & -11.0 & 7.0 \end{bmatrix}.$$

For the implementation, you can use any coding language (e.g., Matlab, Julia, Python, etc.) you want. You can use built-in or external functions to compute the Smith normal form.[3]

Practical information

The homework solution should be written in English.
Please, send it by email to zheming.wang@uclouvain.be, julien.calbert@uclouvain.be and guillaume.berger@uclouvain.be with as object: `[LINMA2380] - Homework 4 - Group XX`, and with the same name for the pdf file containing your solution *(failure to respect these guidelines may induce negative points).*

Deadline for turning in the homework: Monday 14 December 2020 (11:59pm).

It is expected that each group makes the homework individually. If your group has problems or questions, you are welcome to contact the teaching assistants (see emails above).

[3] You are thus *not required* to implement by yourself Algorithm 6.1 of the lecture notes; but, of course, you may if you want to.