Polynomial matrices
introduction, motivations, basic algorithms

Exercises for 22 October 2025

Exercise 0. Gaussian elimination and degree growth. Read and understand the provided code about degree growth when applying Gaussian elimination on a "random" polynomial matrix. Observe that this degree growth has exponential behaviour in the naive version (direct Gaussian elimination), and polynomial behaviour in the version exploiting specificities of polynomials (XGCD computations).

Exercise 1. Matrix equation $\mathbf{AU} = \mathbf{V}$. Let $\mathbf{A} \in \mathbb{K}[X]^{m \times m}$ be nonsingular with all entries of degree $\leqslant d_1$, let $\mathbf{V} \in \mathbb{K}[X]^{m \times k}$ with all entries of degree $\leqslant d_2$.

Two typical cases of interest for the equation $\mathbf{AU} = \mathbf{V}$ are $k = 1$ (linear system solving over $\mathbb{K}[X]$), and $k = m$ with $\mathbf{V} = \mathbf{I}_m$ (inversion of $\mathbf{A}$).

1. Show that $\mathbf{A}^{-1}\mathbf{V}$ can be represented as a fraction with numerator a matrix $\mathbf{U}$ in $\mathbb{K}[X]^{m \times k}$ and denominator a polynomial Δ in $\mathbb{K}[X]$.
2. Give an upper bound on $\deg \det(\mathbf{A})$.
3. Give upper bounds that you can require on $\deg(\Delta)$ and on the degrees of entries of $\mathbf{U}$ (i.e. there exists a couple $(\mathbf{U}, \Delta)$ for question 1 which satisfies these bounds).
4. Prove that $\mathbf{A}^{-1} \in \mathbb{K}[X]^{m \times m} \Leftrightarrow \det(\mathbf{A}) \in \mathbb{K} \setminus \{0\}$.

Remark: matrices with determinant in $\mathbb{K} \setminus \{0\}$ are called *unimodular*.

Exercise 2. Using the **evaluation-interpolation paradigm**,

1. Give a multiplication algorithm for matrices in $\mathbb{K}[X]^{m \times m}$.
2. Give a determinant algorithm.
3. Give an inversion algorithm (finding the inverse over the fractions $\mathbb{K}(X)$).

Hints: exploit known degree bounds on the output to determine the number of points to use; for inversion, you can assume that you have at your disposal a quasi-linear algorithm for Cauchy interpolation (see the slides for references).

For each of these algorithms,

1. Give the lower bound it requires on the cardinality of $\mathbb{K}$.
2. State and prove an upper bound on its complexity.

Further perspective: could your complexity bounds take into account degree measures that refine the matrix degree such as the average row degree?