TUTORIAL 8

1 Wiedemann's algorithm

Let K be a field and $M \in M_n(K)$ be an invertible matrix, with $\omega(M)$ non zero coefficients.

1. Recall the main steps of Wiedemann's algorithm to compute a solution of $Mx = b$ for some vector b. What is its complexity ?

2. Assume now that M is non invertible. Can you modify Wiedemann's algorithm to find a non zero element in the kernel of M ? What complexity do you obtain ?

2 Iterative methods for solving linear systems

In this exercise, we let that $K = \mathbb{R}$ or $K = \mathbb{C}$. We will consider iterative methods to compute an approximation of the solution of a system

$$Ax = b \tag{1}$$

with A an invertible matrix of size n.

1. Let $A = M - N$ with $M, N \in M_n(K)$ and M invertible. Show that solving (1) is equivalent to find a fixed point of the function $f : K^n \to K^n$ defined by $f(x) = M^{-1}Nx + M^{-1}b$.

   In the following two questions, we prove Banach fixed point theorem (ou théorème du point fixe de Picard). This theorem states that under some conditions on a function f, this function has a unique fixed point in K^n. Let $g : K^n \to K^n$ be a contraction mapping, that is for all $x, y \in K^n$, we have $\|g(x) - g(y)\| \leq k\|x - y\|$ for some $k < 1$.

2. Prove that g has at most one fixed point ℓ in K^n.

3. Let $x_0 \in K^n$ be any vector and define $x_{n+1} = g(x_n)$. Prove that this sequence converges. What is its limit ? What is the speed of convergence of this sequence ? (Hint: you may want to use a compacity argument: recall that in $\mathbb{C}^n$ or in $\mathbb{R}^n$, from any bounded sequence you can extract a sub-sequence that converges).

   Let $M \in M_n(K)$ be a matrix (with $K = \mathbb{C}$ or $K = \mathbb{R}$). Let $\|.\|$ be a norm over K^n (for instance $\|.\|_2$ or $\|.\|_\infty$). We define the matrix norm of M associated to $\|.\|$ by

$$|||M||| = \sup_{x \in K^n \setminus \{0\}} \left(\frac{\|Mx\|}{\|x\|} \right).$$

4. Prove that $|||M||| = \max_{x \in K^n, \|x\|=1} (\|Mx\|)$ (beware, there is now a max and not a sup). (Hint: use the fact that the unit ball is compact in K^n).

5. Let f be as in question 1, give a condition on M and N such that we can can apply Banach fixed point theorem to it.

   In the following, we write $A = D - E - F$ with D the diagonal part of A (D is a diagonal matrix with the same coefficients as on the diagonal of A), $-E$ is the lower triangular part of A with zeros on the diagonal and $-F$ is the upper triangular part of A with zeros on the diagonal.

6. **Jacobi's method.** Assume A has non zero diagonal elements. Let $M = D - E$ and $N = F$ and assume that the condition of question 5 is satisfied. Give an algorithm to compute an approximation of x such that $Ax = b$ with at least r bits of precision for each coordinate. What is its complexity in terms of operations in K (assume we already know a ball of radius 10 containing x)?

7. Let A be a strictly row diagonally dominant matrix, that is $|a_{i,i}| > \sum_{j \neq i} |a_{i,j}|$ for all $1 \leq i \leq n$. Prove that the Jacobi's method converges for A (Hint : use the $\| \cdot \|_\infty$ norm to prove that the condition of question 5 is satisfied).

3 Hensel-type strategy for solving linear system

In this exercise, we study algorithms to solve $Mx = b$, $M \in \mathcal{M}_n(K[X]), b \in K[X]^n$. We shall assume that the degree of all coordinates of M, b is $\leqslant d$.

Cramer's formulas show that if x is a solution of $Mx = b$, $(\det M) \cdot x \in K[X]^n$, and the coefficients of $(\det M) \cdot x$ have degree $\leqslant nd$. We'll also assume that $\det M(u) \neq 0$ for all $u \in K$.

1. What is the complexity of computing $B := (M \bmod X)^{-1}$?

   Let $y_i \in K[X]^n$ be a solution of $My_i = b \bmod X^i$, and define $r_i = b - My_i$.

2. Prove that $r_i = \lambda_i X^i$ for some $\lambda_i \in K[X]^n$. If $z_i = B\lambda_i \bmod X$, prove that $y_{i+1} = y_i + X^i z_i$ and $r_{i+1} = r_i - X^i M z_i$.

3. What is the complexity of computing y_{nd+1} using this method? Assuming that $\det M$ is given as input or precomputed, deduce an algorithm for solving $Mx = b$.

4. If we need to compute $\det M$ beforehand, then this computation is going to dominate the complexity of linear system solving. Can we avoid computing the determinant? (Hint: use rational reconstruction.)