1. Tutorial

Exercise 1 (Square root)**.** The goal of this exercise is to calculate the square root of a real number.

1. Give an algorithm for computing the square root.
2. Show that the algorithm converges to the square root.
3. Show that the convergence is quadratic.

Exercise 2 (Newton's p-adic method)**.** Let f be a polynomial with integer coefficients. We will see in this exercise how to reassemble the solutions of f modulo a prime p into solutions modulo p^k.

1. Let f be a polynomial with coefficients in $\mathbb{Z}$. Let us note f' the formal derivative of f. Show that f' has coefficients in $\mathbb{Z}$ and that there exists a polynomial r in two variables and with integer coefficients such that:
$$f(x+h) = f(x) + f'(x)h + r(x,h)h^2.$$
2. Let f be a polynomial with coefficients in $\mathbb{Z}$ and p be a prime not dividing the leading coefficient of f. Show that if there exists x_1 satisfying
$$f(x_1) = 0 \mod p \text{ and } f'(x_1) \neq 0 \mod p$$
then there exists for all $k > 1$ an integer x_k such that
$$f(x_k) = 0 \mod p^k \text{ and } x_k = x_{k-1} \mod p^{k-1}.$$
3. Deduce from the previous question an algorithm allowing to construct a root of a polynomial modulo p^k ($k > 1$ an integer) from its image modulo p. Study its complexity.
4. Let p be an odd prime number. Show that if a is a *quadratic residue* modulo p then it is a quadratic residue modulo p^k ($k > 1$ an integer).
5. Let $x_1 = 3$. Show that x_1 is a *simple root* of $f = x^2 - 2$ seen in $\mathbb{Z}/7\mathbb{Z}[x]$. Compute a square root of 2 modulo 7^k for $k = 2, 3, 4$.

2. Practical

Exercise 3 (Calculation of roots of nonlinear functions)**.**

1. Implement Newton's method seen in class for non-linear functions.
2. Deduce an approximate solution of the positive root of the equation $x^3 = \cos(x)$. Study the speed of convergence.

Exercise 4 (Newton's p-adic method)**.**

1. Implement in MATLAB Newton's p-adic method from exercise 2.
2. Test your code on the example of the question 5 of the exercise 2.

Exercise 5 (Inversion of a matrix)**.** Consider $A \in \mathbb{R}^{n\times n}$, invertible. We look for its inverse by the Newton's method applied to the function $f(X) = A - X^{-1}$.

1. Show that Newton's iteration verifies $X_{n+1} = 2X_n - X_nAX_n$.
2. Program the method starting from $X_0 := A^T/\mathrm{Trace}(A^TA)$. We define the error by $e_n = \|I - X_nA\|_2$.