Algorithme de Berlekamp

Théorème. *Soit K un corps fini de cardinal q, $P \in K[X]$ un polynôme unitaire non constant sans facteur carré. Alors on peut construire $V \in K[X]$ unitaire et non constant modulo P tel que*

$$P = \prod_{a \in K} P \wedge (V - a)$$

Démonstration. Soit $n = \deg(P)$, $\frac{K[X]}{(P)}$ est une K-algèbre de dimension n. Notons $P_1, .., P_r$ les facteurs irréductibles de P, on a un isomorphisme :

$$\psi : \begin{cases} \frac{K[X]}{(P)} \to \bigoplus_{i=1}^{r} \frac{K[X]}{(P_i)} \\ Q\ [P] \mapsto (Q\ [P_1], Q\ [P_2], .., Q\ [P_r]) \end{cases}$$

ψ est bien définie car pour tout $i = 1..r$, $(P) \subset (P_i)$ donc $Q\ [P] \mapsto Q\ [P_i]$ est bien défini. ψ respecte bien la loi de K-algèbre et est en particulier une application K-linéaire entre espaces de même dimension (car $\deg(P) = \deg(P_1) + .. + \deg(P_r)$) ; il suffit de montrer que ψ est injective.
Or, si pour tout i, $P_i | Q$, comme les P_i sont premiers entre eux, on a alors $P = P_1..P_r | Q$, donc $Q\ [P] = 0$, d'où le résultat.

Soit f l'endomorphisme de $\frac{K[X]}{(P)}$ défini par

$$f(Q\ [P]) = Q^q\ [P]$$

Vérifions que f est bien un endomorphisme ; $Car(K) | q$, donc $Q \mapsto Q^q$ est une application K-linéaire (en fait, $Q(X)^q = Q(X^q)$). Cette application passe bien au quotient par (P) car si $P | (Q_1 - Q_2)$, alors il faut $P | (Q_1^q - Q_2^q)$, ce qui est vrai grâce à l'identité suivante : $(Q_1^q - Q_2^q) = (Q_1 - Q_2)(Q_1^{q-1} - Q_1^{q-2}Q_2 + .. + (-1)^{q-1}Q_2^{q-1})$.
Soit $g = \psi \circ f \circ \psi^{-1}$, comme ψ préserve la structure d'algèbre, on en déduit que :

$$g(Q_1\ [P_1], .., Q_r\ [P_r]) = (Q_1^q\ [P_1], .., Q_r^q\ [P_r])$$

On note $Fix(f) = Ker(f - I)$, $Fix(g) = Ker(g - I)$, on a $Fix(g) = \psi(Fix(f))$. Étudions la dimension de $Fix(g)$:
Comme les P_i sont irréductibles, les $\frac{K[X]}{(P_i)}$ sont des corps. Le polynôme $T^q - T$ admet donc au plus q racines dans $\frac{K[X]}{(P_i)}$, et pour tout $x \in K$, $x^q = x$, donc $T^q - T$ admet exactement K comme racines. On en déduit la description de $Fix(g)$ suivante :

$$Fix(g) = \{(\lambda_1\ [P_1], .., \lambda_r\ [P_r]), (\lambda_1, .., \lambda_r) \in K^r\}$$

En particulier, $dim(Fix(f)) = r$. Si $r = 1$, f est irréductible et c'est fini. Si $r \geq 2$, on peut trouver $V \in K_n[x]$ un polynôme unitaire non constant modulo P (car l'espace des constantes modulo P est de dimension 1), tel que $P | (V^q - V)$, donc $P = P \wedge (V^q - V)$. On peut écrire $V^q - V = \prod_{x \in K}(V - x)$, et les $(V - x)_{x \in K}$ sont premiers entre eux, d'où le résultat. Regardons maintenant comment faire le calcul explicite.

Pour la suite, on fixe $(1, X, .., X^{n-1})$ une K-base de $\frac{K[X]}{(P)}$. Pour tout $j \in [0, n-1]$, la division euclidienne de X^{jq} par P donne un polynôme $\sum_{i=0}^{n-1} M_{i,j} X^j$ congrus à X^{jq} modulo P. $M := (M_{i,j})_{0 \leq i,j \leq n-1}$ est la matrice de f dans la base canonique.
On fait un pivot de Gauss (opérations sur les **lignes**) sur $M - I$ pour la ramener à une forme échelonnée, qui donne une base de $Fix(f)$. Il suffit en fait de deux vecteurs linéairement indépendants de cette base pour être sûr d'avoir un polynôme non constant.

□

En pratique, si q est très grand, la deuxième étape de l'algorithme nécessite de calculer q pgcd de degré n, ce qui peut devenir long.
Une autre approche est d'écrire

$$P = (P \wedge V) \times \left(P \wedge (V^{\frac{q-1}{2}} - 1)\right) \times \left(P \wedge (V^{\frac{q-1}{2}} + 1)\right)$$

Avec V choisi au hasard (uniformément) dans $Fix(f)$. En effet, si on note $v_i \in K$ tel que $P_i | (V - v_i)$ (image de V par ψ), alors il y a trois possibilités : $v_i = 0$ ou v_i est un carré et donc $v_i^{\frac{p-1}{2}} = 1$, ou v_i n'est pas un carré et $v_i^{\frac{p-1}{2}} = -1$. Alors la décomposition ci-dessus est triviale si et seulement si tous les v_i sont du même 'type' ; on peut majorer la probabilité de cet événement par $\frac{1}{2^{r-1}}$. En répétant k fois l'opération, on trouve un facteur non trivial avec probabilité $\frac{1}{2^{(r-1)k}}$. Référence : Demazure