TER - Calcul rapide des polynômes cyclotomiques

Jean-Philippe Merx

M1 Mathématiques - Sorbonne Université

Mai 2025

Contenu du TER

- Analyser l'article d'Andrew Arnold et Michael Monagan sur le calcul rapide des polynômes cyclotomiques
- Expliquer les méthodes et algorithmes utilisés
- Éventuellement réaliser une mise en œuvre et utiliser des logiciels de calcul formel

Définition des polynômes cyclotomiques

- Si U_n est le groupe cyclique des racines n-ième complexes de l'unité, le polynôme cyclotomique Φ_n est :

$$\Phi_n(X) = \prod_{\zeta \in U_n^*} (X - \zeta) = \prod_{\substack{j=1 \\ \mathrm{pgcd}(j,n)=1}}^{n} (X - e^{\frac{2\pi i}{n} j})$$

où $U_n^* \subseteq U_n$ est l'ensemble des générateurs de U_n.

- Polynôme cyclotomique inverse :

$$\Psi_n(x) = \prod_{\zeta \in U_n \setminus U_n^*} (X - \zeta) = \prod_{\substack{j=1 \\ \mathrm{pgcd}(j,n)>1}}^{n} (X - e^{\frac{2\pi i}{n} j}) = \frac{X^n - 1}{\Phi_n(X)}.$$

De l'importance des diviseurs de n

- $\Phi_n \in \mathbb{Z}[X]$ de degré $\varphi(n)$ où φ est l'indicatrice d'Euler et :

$$P_n(X) = X^n - 1 = \prod_{d|n} \Phi_d(X)$$

- Pour p premier ne divisant pas n :

$$\Phi_{np}(X) = \frac{\Phi_n(X^p)}{\Phi_n(X)} \text{ et } \Psi_{np}(X) = \Psi_n(X^p)\Phi_n(X)$$

et pour q premier divisant n

$$\Phi_{nq}(X) = \Phi_n(X^q) \text{ et } \Psi_{nq}(X) = \Psi_n(X^q)$$

Restriction à n produit de premiers impairs distincts

- Comme on a aussi :

$$\Phi_{2n}(X) = \Phi_n(X^2) \text{ si } 2 \mid n \text{ et } \Phi_{2n}(X) = \Phi_n(-X) \text{ sinon}$$

$\Longrightarrow$ il suffit de considérer n produit de premiers impairs distincts.

- D'où un premier algorithme possible pour $n = p_1^{e_1} \cdots p_k^{e_k}$:
  - Itérations pour calculer $\Phi_m(X) = \Phi_{p_1 \cdots p_k}(X)$
  - Calcul de $\Phi_{n/m}(X)$
- Les divisions de polynômes sont la base de l'algorithme
- Utilisation d'une division rapide avec Newton sur séries formelles et FFT

Formules closes

Si $\mu : \mathbb{N}^* \to \{-1, 0, 1\}\}$ est la fonction de Möbius :

$$\mu(n) =: \begin{cases} 1 & \text{si } n = 1 \\ 0 & \text{si } n \text{ a un facteur premier carré} \\ (-1)^r & \text{où r est le nombre de facteurs premiers de n} \end{cases}$$

on a :

$$\Phi_n(X) = \prod_{d|n} (1 - X^{\frac{n}{d}})^{\mu(d)} \tag{2.4}$$

$$\Psi_n(X) = - \prod_{d|n, d<n} (1 - X^d)^{-\mu(\frac{n}{d})} \tag{2.5}$$

Points critiques pour l'implémentation

- **RAPIDITÉ** $\Longrightarrow$ type d'algorithme
- **Taille mémoire** : pour $n = 3 \cdot 5 \cdot 7 \cdot 11 \cdot 13 \cdot 17 \cdot 19 \cdot 23 \cdot 29$, $\deg \Phi_n = 1,021,870,080$ et on obtient un polynôme nécessitant au moins 8 Go de mémoire.
- **Hauteur** $A(n)$: quel type de données manipuler ?
  Un sujet en soi sur lequel on revient plus loin

Algorithmes SPS et SPS-Psi

- Application des formules closes :
  - Multiplication par $1 - X^d$: $\varphi(n)$ soustractions
  - Division par $1 - X^d$... ATTENTION ! ! !
    Seules $\varphi(n)$ additions sont nécessaires et non $(\varphi(n))^2$
- Algorithme simple avec une allocation mémoire unique pour Φ_n.
- ... Mais tous les calculs sont effectués sur un tableau de taille $\varphi(n)$

Algorithme SPS4

Utilisation de la localité des formules initiales et de l'efficacité des formules closes
Formules récursives de calcul des polynômes cyclotomiques :

$$\Phi_n(X) = \prod_{j=2}^{k} -\Psi_{m_j}(X^{e_j}) \prod_{j=1}^{k} (1 - X^{n/p_j})^{-1} (1 - X^n) \tag{3.17}$$

$$\Psi_n(X) = \prod_{j=1}^{k} \Phi_{m_j}(X^{e_j}) \tag{3.25}$$

où $n = p_1 \cdots p_k$ et pour $1 \leq j \leq k$, $e_j = p_{j+1} \cdots p_k$ et $m_j = p_1 \cdots p_{j-1}$

Calcul sur des polynômes de degrés inférieurs au degré total

Démonstration

Notebooks Python SAGE & module Python