Travail d'étude et de recherche (TER)
M1 de Mathématiques - Sorbonne Université
Calcul des polynômes cyclotomiques

Jean-Philippe MERX

Mai 2025

Résumé

Ce document constitue le mémoire de mon TER effectué dans le cadre du M1 de Mathématiques de Sorbonne Université. Le TER a consisté en l'étude de l'article [2] d'Arnold et Monagan consacré au calcul rapide des polynômes cyclotomiques. Le travail principal s'est concentré sur d'une part la compréhension des résultats mathématiques nécessaires à justifier le fonctionnement des algorithmes proposés par Arnold et Monagan et en particulier l'algorithme récursif le plus rapide, et d'autre part à analyser des éléments importants pour la mise en œuvre de cet algorithme comme la magnitude des coefficients des polynômes à calculer. Une implémentation en Python de l'algorithme a été réalisée dans le cadre du TER.

Remerciements

Lorsque j'ai opté pour la réalisation d'un TER, j'avais à l'esprit l'étude de l'article de John Milnor sur l'existence sur la sphère de dimension 7 de structures différentielles distinctes. J'ai toujours été attiré par *les bizarreries mathématiques* comme l'atteste mon site www.mathcounterexamples.net dédié aux contre-exemples. J'ai cependant consulté avec attention la liste des TERs proposés par les enseignants du M1... et le sujet consacré au calcul des polynômes cyclotomiques a retenu mon attention. Ayant passé plusieurs années à développer des logiciels de CAO, m'arrachant les cheveux à mettre à plat des surfaces non développables et au total quarante années dans l'univers des éditeurs de logiciels, j'ai retenu cet attrayant sujet qui allie théorie et implémentation.

Je tiens à remercier Pierre-Vincent Koseleff d'avoir proposé ce sujet et d'avoir encadré mes activités de TER. Les échanges que nous avons eus m'ont permis d'avoir une vue plus directe du comportement des polynômes cyclotomiques, et une ouverture sur les algorithmes utilisés en calcul formel.

Introduction

Le sujet du TER est à la jonction de l'arithmétique et du calcul formel. Pour mémoire, Pierre-Vincent Koseleff l'avait formulé ainsi :

« Dans [AM10], Arnold et Monagan proposent une méthode pour calculer efficacement et rapidement les polynômes cyclotomiques Φ_n. Il s'agit de comprendre les méthodes utilisées et mises en œuvre, ainsi que d'expliquer, dans le cadre du programme du Master de Mathématiques, divers

points restés obscurs ou imprécis : propriétés des polynômes cyclotomiques, irréductibilité, taille des coefficients ; multiplications et divisions rapides, transformation de Fourier rapide. Les ouvrages [GG13, AECF, SM] pourront utilement être consultés. Le travail peut mener à des calculs explicites et un premier contact avec des logiciels de calcul formel (Sage, Maple, etc). »

Il s'est rapidement avéré que le sujet est vaste et que des choix étaient nécessaires. J'avais dès le départ le désir de faire intervenir une part d'implémentation dans le TER. L'un des mots clés du sujet étant *calcul formel*, et me voyant difficilement calculer à la main ne serait-ce que $\Phi_{3\cdot5\cdot7\cdot11}$, mettre la main dans du logiciel faisait sens. Pour ce faire, la première direction qui s'est imposée à moi, a été de comprendre les algorithmes décrits dans l'article et les mathématiques sous-jacentes. Réflexe subliminal : pourquoi ça marche ? La première section de ce mémoire contient les preuves des propriétés utilisées pour les algorithmes d'Arnold et Monagan, et en particulier l'algorithme récursif `SPS4`.

Si pour $p < q$ premiers, les coefficients du polynôme cyclotomique Φ_{pq} appartiennent à l'ensemble $\{-1, 0, 1\}$, ce n'est pas toujours le cas pour Φ_n où n est le produit de trois entiers premiers ou plus. Donc si l'on souhaite implémenter un algorithme de calcul de polynômes cyclotomiques, se pose la question de la magnitude des coefficients de ces polynômes. Et ce parce que les calculs sur des entiers de « longueur infinie » sont significativement plus lents que sur ceux de « taille machine ». En conséquence, une section du mémoire est consacrée à la hauteur des polynômes cyclotomiques. C'est un sujet vaste auquel de nombreux articles sont consacrés. Je n'en aborde dans ce mémoire qu'une petite partie. Sans avoir pu résister cependant à reproduire la preuve que tout entier est le coefficient d'un polynôme cyclotomique.

Une autre partie du mémoire introduit l'implémentation que j'ai réalisé en Python de l'algorithme `SPS4`. Les programmes, ainsi que l'ensemble des documents du mémoire sont accessibles dans le dépôt Git jpmjpmjpm/cyclotom-fast.

Mathématiques des polynômes cyclotomiques

Définitions

Pour un entier $n \geq 1$, on désigne le polynôme

$$P_n(X) = X^n - 1 \in \mathbb{Q}[X]$$

et par U_n les racines de P_n, c'est à dire les racines $n^{\text{ème}}$ de l'unité. U_n est un sous-groupe du groupe $\mathbb{U}$ des complexes de module un. C'est un groupe cyclique fini. On peut le montrer en se souvenant qu'un groupe abélien fini G est somme directe de ses sous-groupes p-maximaux $G(p)$, c'est à dire de ses éléments qui sont une puissance de p. Ici, on peut montrer que $G(p) = \{x \in U_n \mid x^{m_p} = 1\}$ où m_p est l'ordre de $G(p)$ et donc que $G(p)$ est cyclique. On conclut sachant qu'un groupe abélien fini somme directe de groupes d'ordres premiers est cyclique.

On désigne dans la suite par $U_n^* \subseteq U_n$ les générateurs de U_n, c'est à dire les éléments dont l'ordre est premier avec n. Le $n^{\text{ème}}$ polynôme cyclotomique $\Phi_n \in \mathbb{Q}[X]$ est alors défini par :

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

et le $n^{\text{ème}}$ polynôme cyclotomique inverse $\Psi_n \in \mathbb{Q}[X]$ par :

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

En particulier : $\Phi_1(X) = X - 1$ et $\Psi_1(X) = 1$.

Propriétés utiles au calcul des polynômes cyclotomiques

On s'attache maintenant à décrire et démontrer des propriétés des polynômes Φ_n, Ψ_n qui vont permettre de les calculer efficacement.

Propriété 1
Pour $n \geq 1$ on a $P_n(X) = X^n - 1 = \prod_{d|n} \Phi_d(X)$.

En effet, U_n est la réunion des U_d^* pour $d \mid n$ et les U_d^* sont deux à deux disjoints.

Propriété 2
Les polynômes Φ_n sont à coefficients entiers et unitaires.

Si P, Q sont des polynômes à coefficients entiers, et que Q est unitaire, alors il existe des polynômes A, R appartenant à $\mathbb{Z}[X]$ tels que $P = AQ + R$. De plus, A est unitaire si P, Q le sont. On prouve ce résultat en appliquant l'algorithme de calcul de la division euclidienne de polynômes dans un corps et en remarquant qu'à chaque étape le terme obtenu est à coefficient entier.
L'utilisation de la propriété 1 et une récurrence forte permet de conclure que Φ_n est un polynôme à coefficients entiers unitaire pour $n \geq 1$.

Propriété 3 (Irréductibilité des polynômes cyclotomiques)
Les polynômes $\Phi_n \in \mathbb{Z}[X]$ sont irréductibles.

On démontre cette propriété bien qu'elle ne soit pas utile pour les algorithmes utilisés ici pour leur calcul.

Soit ζ une racine primitive n-ième de l'unité, p un nombre premier avec n et $f, g \in \mathbb{Q}[X]$ les polynômes minimaux unitaires de ζ et ζ^p. Montrons que f, g sont à coefficients entiers. $\mathbb{Z}$ étant factoriel, $\mathbb{Z}[X]$ l'est également et $\Phi_n = f_1^{\alpha_1} \cdots f_r^{\alpha_r}$ où $f_1, \ldots, f_r \in \mathbb{Z}[X]$. Φ_n étant unitaire, on peut supposer que les f_i le sont aussi. ζ est racine de l'un des f_i, qui est irréductible sur $\mathbb{Z}$ et donc sur $\mathbb{Q}$. Par conséquent, f est l'un des f_i, g aussi puisque ζ^p est racine de Φ_n et f, g divisent Φ_n dans $\mathbb{Z}[X]$.

Montrons que $f = g$. Dans le cas contraire, f, g étant irréductibles, le produit $f \cdot g$ diviserait Φ_n dans $\mathbb{Z}[X]$. ζ^p étant racine de g, ζ est racine de $g(X^p)$ et $f(X)$ divise $g(X^p)$ dans $\mathbb{Q}[X]$, donc aussi dans $\mathbb{Z}[X]$ puisque f est unitaire. On écrit $g(X^p) = f(X)h(X)$ avec $h \in \mathbb{Z}[X]$.

Projetons cette égalité dans $\mathbb{F}_p$. $g(X) = a_r X^r + \cdots + a_0$ avec $a_i \in \mathbb{Z}$. Par Frobenius :

$$\overline{g}(X^p) = (\overline{a_r} X^r + \cdots + \overline{a_0})^p = \overline{g}(X)^p = \overline{f}(X)\overline{h}(X).$$

Si $\varphi \in \mathbb{F}_p[X]$ est un facteur irréductible de $\overline{f}$, φ divise $\overline{g}$ et donc φ^2 divise $\overline{\Phi}_n$ dans $\mathbb{F}_p[X]$. Ce qui est absurde, puisque $\overline{\Phi}_n$ divise le polynôme $X^n - 1$ qui n'a que des racines simples dans un corps de décomposition lorsque p ne divise pas n.

Maintenant, si ζ' est une racine n-ième de l'unité, on a $\zeta' = \zeta^m$ avec $m = p_1^{\alpha_1} \cdots p_s^{\alpha_s}$ où $p_i \nmid n$. Des applications successives de ce que l'on vient de montrer implique que ζ' et ζ ont le même polynôme minimal. De sorte que f admet toutes les racines primitives de l'unité comme zéro. Comme $f \mid \Phi_n$, et que ces deux polynômes sont unitaires, on obtient que $f = \Phi_n$ et que Φ_n est irréductible.

Notons que les polynômes cyclotomiques, ne sont pas toujours irréductibles sur les corps finis. Ainsi, sur $\mathbb{F}_2$:

$$\Phi_7(X) = (X^2 + X^2 + 1)(X^3 + X + 1).$$

Nous définissons maintenant la fonction de Möbius $\mu : \mathbb{N}^* \to \{-1, 0, 1\}\}$:

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

Propriété 4 (Deux propriétés de la fonction de Möbius)

1. *μ est multiplicative : si m, n sont premiers entre eux : $\mu(mn) = \mu(m)\mu(n)$.*
2. *Si $n > 1$, $\sum_{d|n} \mu(d) = 0$.*

Si m ou n est égal à 1, la première propriété est évidente. Si m, n sont premiers entre eux, $\mu(mn) = \mu(m)\mu(n)$. Quantité qui est nulle si m ou n a un facteur premier carré. Dans le cas contraire, le nombre de facteurs premiers de mn est la somme des nombres de facteurs premiers de m et n.

Concernant la seconde propriété, supposons que $n = \prod_{n=1}^{r} p_1^{\alpha_1} \cdots p_r^{\alpha_r}$ où $\{p_1, \ldots p_r\}$ sont des premiers distincts et $\alpha_1, \ldots, \alpha_r$ des entiers supérieurs ou égaux à 1. Dans la somme $\sum_{d|n} \mu(d)$, seuls les diviseurs d de n sans facteur premier carré ont une contribution, ce qui conduit à l'égalité :

$$\sum_{d|n} \mu(d) = \sum_{i=0}^{r} \sum_{\substack{S \subseteq \{p_1, \ldots, p_r\} \\ |S| = i}} (-1)^i = \sum_{i=0}^{r} \binom{r}{i} (-1)^i = (1-1)^r = 0.$$

Nous fournissons à présent une autre définition des polynômes cyclotomiques utilisant la fonction de Möbius.

Propriété 5 (Définition équivalente des polynômes cyclotomiques)

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

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

La seconde formule est une conséquence immédiate de la première et de l'égalité $\Phi_n(X)\Psi_n(X) = X^n - 1$. Pour montrer la première, notons

$$F_n(X) = \prod_{d|n}(X^d - 1)^{\mu(\frac{n}{d})} = \prod_{d|n}(X^{\frac{n}{d}} - 1)^{\mu(d)}.$$

Il suffit de prouver que $\prod_{d|n} F_d(X) = X^n - 1$, puisque l'on obtient alors par récurrence $F_n(X) = \Phi_n(X)$ partant du constat que $F_1(X) = \Phi_1(X) = 1$. Nous avons :

$$\prod_{d|n} F_d(X) = \prod_{d|n}\prod_{d'|d}(X^{d'} - 1)^{\mu(\frac{d}{d'})}$$

et allons montrer que pour d' divisant n fixé

$$\sum_{d\in S'_d} \mu\left(\frac{d}{d'}\right) = \begin{cases} 1 & \text{si } d = n \\ 0 & \text{autrement} \end{cases}$$

où $S_{d'} = \{d; d' \mid d \text{ et } d \mid n\}$. Dans la somme ci-dessus, on peut changer d'indice de sommation en prenant $e = d/d'$ et effectuer la somme sur l'ensemble $\overline{S}_{d'} = \{e; e \mid \frac{n}{d'}\}$, c'est à dire l'ensemble des diviseurs de $\frac{n}{d'}$. D'où

$$\sum_{d\in S'_d} \mu\left(\frac{d}{d'}\right) = \sum_{e|\frac{n}{d'}} \mu(e)$$

et le résultat compte tenu de la seconde propriété ci-dessus de la fonction de Möbius.
Cette définition permettra le calcul des polynômes cyclotomiques en effectuant des multiplications et des divisions par des polynômes du type $X^d - 1$.
Les propriétés suivantes des polynômes cyclotomiques sont importantes pour les algorithmes de calcul. Elles permettent en effet d'une part de limiter les calculs principaux au cas où n est un produit de nombres premiers distincts, et pour ce dernier cas d'élaborer un algorithme récursif pour leur calcul : l'algorithme `SPS4` de l'article étudié.

Propriété 6
Soient p, q des entiers premiers tels que $p \nmid n$ et $q \mid n$. Alors :

$$\Phi_{np}(X) = \frac{\Phi_n(X^p)}{\Phi_n(X)} \tag{2.8a}$$
$$\Phi_{nq}(X) = \Phi_n(X^q) \tag{2.8b}$$
$$\Psi_{np}(X) = \Psi_n(X^p)\Phi_n(X) \tag{2.8c}$$
$$\Psi_{nq}(X) = \Psi_n(X^q) \tag{2.8d}$$

Remarquons tout d'abord que si m est un entier quelconque

$$\Phi_{nm}(X)\Psi_{nm}(X) = X^{nm} - 1 = (X^p)^m - 1 = \Phi_n(X^m)\Psi_n(X^m).$$

Si 2.8a est vraie on obtient

$$\Psi_{np}(X) = \frac{X^{np} - 1}{\Phi_{np}(X)} = \frac{X^{np} - 1}{\Phi_n(X^p)}\Phi_n(X) = \Psi_n(X^p)\Phi_n(X)$$

montrant 2.8*c*. De manière similaire, on déduit 2.8*d* de 2.8*b*.

Pour n entier, nous noterons dorénavant D_n l'ensemble des diviseurs de n sans facteurs carré. On peut réécrire 2.4

$$\Phi_n(X) = \prod_{d \in S_n} (1 - X^{\frac{n}{d}})^{\mu(d)}.$$

Si d est un diviseur de np sans facteur premier, ou bien d divise n, ou bien d est de la forme $d'p$. D'où l'égalité

$$\Phi_{np}(X) = \prod_{d \in S_n} (1 - X^{\frac{np}{d}})^{\mu(d)} \prod_{dp \in S_{np}} (1 - X^{\frac{np}{dp}})^{\mu(dp)}.$$

Le premier facteur est $\Phi_n(X^p)$. Observons que dans le second, dp étant sans facteurs carré, d est premier avec p et donc $\mu(dp) = \mu(d)\mu(p) = -\mu(d)$. Ce second facteur est simplement l'inverse de Φ_n : nous avons prouvé 2.8*a*.

Il nous reste à prouver 2.8*b*. Puisque q divise n, les diviseurs sans facteurs carré de nq sont ceux de n et

$$\Phi_{nq}(X) = \prod_{d \in S_n} (1 - X^{\frac{nq}{d}})^{\mu(d)} = \Phi_n(X^q),$$

qui est le résultat souhaité.

Portant notre attention sur le cas où 2 divise n, nous allons obtenir la propriété suivante :

Propriété 7
Pour $n > 1$:

$$\Phi_{2n}(X) = \begin{cases} \Phi_n(X^2) & \textit{si } n \textit{ est pair} \\ \Phi_n(-X) & \textit{sinon} \end{cases}$$

Le cas n pair est une application immédiate de 2.8*b*. Si n est impair, les polynômes $\Phi_{2n}(X)$ et $\Phi_n(-X)$ sont égaux puisqu'ayant les mêmes racines qui sont simples. Ce résultat nous permet de nous concentrer sur le calcul des polynômes cyclotomiques Φ_n avec n impair.

On déduit d'autre part des formules précédentes 2.8*b* et 2.8*d* pour le cas où $q \mid n$ que

Propriété 8 (Réduction aux cas des entiers sans facteurs carré)
Si $n = q_1^{e_1} \cdots q_k^{e_k}$ où $q_1, \ldots, q_k$ sont des premiers distincts :

$$\Phi_n(X) = \Phi_{q_1 \cdots q_k}(X^{q_1^{e_1-1} \cdots q_k^{e_k-1}})$$
$$\Psi_n(X) = \Psi_{q_1 \cdots q_k}(X^{q_1^{e_1-1} \cdots q_k^{e_k-1}})$$

Cette propriété et la précédente permettent de limiter le calcul des polynômes cyclotomiques au cas où n est un produit de nombres premiers impairs distincts.

Pour ce faire, supposons que $n = p_1 \cdots p_k$ soit le produit de k nombres premiers **impairs** distincts. Pour $1 \leq i \leq k$, notons $m_i = p_1 \cdots p_{i-1}$ et $e_i = p_{i+1} \cdots p_k$. En particulier $m_1 = e_k = 1$ et on note $e_0 = n$. Pour $1 \leq i \leq k$, nous avons alors $n = e_i p_i n_i$, ainsi que $e_{i-1} = p_i e_i$ et $m_{i+1} = m_i p_i$. À partir de la formule 2.8*a* et de l'identité $\Phi_{m_k}(X)\Psi_{m_k}(X) = X^{m_k} - 1$ on obtient

$$\Phi_n(X) = -\frac{\Psi_{m_k}(X^{e_k})}{1 - X^{n/p_k}} \Phi_{m_k}(X^{e_{k-1}}).$$

Sachant que $\Phi_1(X^{e_0}) = \Phi_1(X^n) = X^n - 1$ on montre alors par récurrence :

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

ou encore

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

puisque $\Psi_{m_1}(X^{e_1}) = \Psi_1(X^{e_1}) = 1$. L'utilisation de la formule 2.8*c* et une récurrence similaire permet d'écrire $\Psi_n(X)$ comme produit de polynômes cyclotomiques d'ordres inférieurs ; ce que l'on résume dans la propriété suivante :

Propriété 9 (Formules récursives de calcul des polynômes cyclotomiques)

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

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

On note dans la suite du document $\varphi(n)$ l'indicatrice d'Euler de n.

Propriété 10 (Palindromie des polynômes cyclotomiques)
Pour $n > 1$ impair, le polynôme $\Phi_n(X) = \sum_{i=0}^{\varphi(n)} a_i X^i$ est palindromique tandis que $\Psi_n(X) = \sum_{j=0}^{n-\varphi(n)} b_j X^j$ est anti-palindromique, ce qui signifie que

$$a_i = a_{\varphi(n)-i} \text{ et } b_j = -b_{n-\varphi(n)-j}.$$

Si ω est une racine primitive n-ième de l'unité, ω^{-1} aussi. n étant supposé impair, $\omega \neq \omega^{-1}$ et le produit des racines de Φ_n est égal à $1 = a_0$ puisque $\varphi(n)$ est pair pour $n > 2$. $X^{\varphi(n)}\Phi_n(\frac{1}{X})$ est un polynôme unitaire, dont les racines sont exactement celles de $\Phi_n(X)$, ce qui prouve que $\Phi_n(X)$ est palindromique.

De l'égalité $\Phi_n(X)\Psi_n(X) = X^n - 1$ il résulte d'une part que $\Psi_n(X)$ est unitaire et d'autre part que $b_0 = -1$. On remarque aussi que si ζ n'est pas une racine primitive n-ième de l'unité, alors ζ^{-1} non plus. $-X^{n-\varphi(n)}\Psi_n(\frac{1}{X})$ est donc un polynôme unitaire, comme $\Psi_n(X)$ et qui possède précisément les racines de $\Psi_n(X)$. Ces deux polynômes sont donc égaux, et nous concluons que $\Psi_n(X)$ est anti-palindromique.

Propriété 11 (Produit de polynômes palindromiques / anti-palindromiques)
Le produit de deux polynômes palindromiques ou anti-palindromiques est palindromique.
Le produit d'un polynôme palindromique par un polynôme anti-palindromique est anti-palindromique.

Démonstration claire à partir des définitions.

Calcul des polynômes cyclotomiques

Comme mentionné à la section précédente, les propriétés 7 et 8 permettent de restreindre les calculs aux polynômes cyclotomiques Φ_n pour n impair et sans facteur carré. C'est l'hypothèse

constamment retenue par Arnold et Monagan dans les algorithmes qu'ils présentent. On décrit maintenant succinctement les algorithmes de l'article pour se concentrer ensuite sur le dernier à savoir `SPS4`.

Le premier, utilise de manière itérative la formule 2.8a, à savoir :

$$\Phi_{np}(X) = \frac{\Phi_n(X^p)}{\Phi_n(X)}$$

où p premier ne divise pas n. La mise en œuvre de cet algorithme nécessite des divisions de polynômes. C'est la méthode utilisée pour calculer les polynômes cyclotomiques du package Python SymPy. Dans ce package, la division des polynômes est effectuée par « division euclidienne classique ». Arnold et Monagan mentionnent que de leur côté, il ont implémenté cet algorithme en faisant appel à la FFT et à la méthode de Newton appliquée aux séries formelles pour les divisions de polynômes selon [8].

Les autres algorithmes de l'article s'appuient sur la formule close 2.4 :

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

L'algorithme `SPS` utilise directement cette formule et la palindromicité de Φ_n. Cependant, les calculs sont effectués à chaque étape sur l'ensemble des coefficients de Φ_n, qui rappelons le sont au nombre de $\varphi(n)$. L'algorithme `SPS-Psi` est le pendant de `SPS` pour les polynômes cyclotomiques inverses.

L'algorithme `SPS2` utilise la formule :

$$\Phi_{mp}(X) = -\Psi_m(X)\Phi_m(X^p)\frac{1}{1 - X^m}.$$

Si bien que pour calculer Φ_n où $n = mp$ et p le plus grand diviseur premier de n, on peut utiliser la procédure `SPS-Psi` sur un polynôme de degré $m - \varphi(m)$ au lieu d'effectuer des calculs sur des polynômes de degré $\varphi(n)$. Cependant, cet avantage n'est utilisé qu'une fois pour $\Psi_m(X)$.

L'algorithme `SPS3` utilise quant à lui ce bénéfice de manière itérative en appliquant la formule précédente à $\Phi_m(X^p)$.

L'article d'Arnold et Monagan compare les performances des algorithmes des algorithmes proposés en fournissant un tableau contenant les temps de calcul pour différents polynômes cyclotomiques. Cependant, il n'y a pas d'analyse détaillée des échelles en temps des algorithmes. Ces échelles de temps sont mentionnées dans une autre version de l'article [1].

Dans la suite de cette section, on décrit la méthode utilisée dans l'algorithme `SPS4` de l'article pour calculer de manière rapide les polynômes cyclotomiques, ainsi qu'une implémentation effectuée en Python. On commence par quelques considérations techniques.

Éléments techniques à prendre en compte

Le calcul d'un polynôme cyclotomique d'ordre n comportant des facteurs carrés s'effectue en utilisant la propriété 8. Dans ce cas, on obtient des polynômes creux. Par souci d'économie de

mémoire, on utilise alors une représentation sous la forme par exemple de dictionnaires plutôt que de simples tableaux. C'est ce qui est réalisé dans l'implémentation SAGE `cyclotomic.pyx`.

On cherche donc à calculer de manière rapide les polynômes cyclotomiques Φ_n pour lesquels n "est grand". Mais qu'entend-on par grand ? Il y a au moins deux éléments importants à considérer.

Tout d'abord l'espace mémoire nécessaire pour le stockage d'un polynôme cyclotomique Φ_n. Si $n = p_1 \cdots p_k$ où $p_1, \ldots, p_k$ sont k premiers impairs distincts, le degré de Φ_n est égal à l'indicatrice d'Euler $\varphi(n) = (p_1 - 1) \cdots (p_k - 1)$. Ainsi pour $n = 3 \cdot 5 \cdot 7 \cdot 11 \cdot 13 \cdot 17 \cdot 19 \cdot 23 \cdot 29$, le degré de Φ_n est égal à $1,021,870,080$ et à supposer que chacun des coefficients de Φ_n tienne sur 64 bits, on obtient un polynôme nécessitant 8 Go de mémoire. On approche ici des limites de ce qui calculable en mémoire vive sur un ordinateur personnel.

Un autre paramètre majeur pour les implémentations est la magnitude des coefficients de Φ_n. On désigne par $A(n)$ la hauteur de Φ_n, c'est à dire la valeur maximale des valeurs absolues des coefficients de Φ_n. Nous verrons que l'algorithme `SPS4` n'utilise que des calculs sur les entiers et que seules des additions (et soustractions) sont mises en œuvre. Encore faut-il pour obtenir des calculs rapides que les entiers sur lesquels on travaille puissent-être représentés par des "entiers machine", ce qui veut généralement dire en 2025 sur 64 bits. Si ce n'est pas le cas, on fait face à un dilemme. Ou bien l'on travaille en précision entière infinie (ce qui est le cas par défaut de Python), mais on perd beaucoup en rapidité. Ou bien on teste dans tous les calculs l'overflow, ce qui est mieux, mais cependant plus lent que d'effectuer directement les additions et soustractions. L'écart de rapidité est cependant ici simplement proportionnel. Une troisième méthode est possible. Elle consiste à avoir calculé par un programme où l'on se prémunit de l'overflow le plus petit n, nommé n_{64} pour lequel $A(n)$ reste représentable sur 64 bits, puis dans le *programme définitif*, à mettre en place un calcul différent selon que l'on souhaite calculer Φ_n avec $n \geq n_{64}$ ou $n < n_{64}$.

C'est cette troisième stratégie qui est implémentée dans SAGE. `cyclotomic.pyx` utilise les résultats suivants de Michael Managan :

- Pour $n < 10163195$, $A(n)$ est inférieure ou égale à 74989473, soit 26.16 bits, et on peut effectuer les calculs en 32 bits.
- Pour $n = 10163195$, $A(n) = 1376877780831$, soit 40.32 bits.
- Si $n < 169,828,113$, $A(n)$ tient sur 60 bits.
- Pour $n = 169828113$, on obtient une hauteur égale à 31484567640915734951 qui nécessite 65 bits pour être représenté. Dans ce dernier cas, `cyclotomic.pyx` bascule sur l'utilisation de PARI avec le joli warning : `print("Warning: using PARI (slow!)")`.

On le voit, la hauteur $A(n)$ est importante pour la mise en œuvre d'un algorithme de calcul rapide des polynômes cyclotomiques. Pour $n \geq 3 \cdot 7 \cdot 13 \cdot 17 \cdot 23 \cdot 37 \cdot 43 = 169828113$, on n'est plus à même d'utiliser l'arithmétique 64 bits des machines. Il est alors nécessaire de faire appel à des méthodes de calcul en arithmétique infinie.

On aborde dans une section dédiée la question de la hauteur des polynômes cyclotomiques. Revenons pour le moment à l'algorithme `SPS4`.

Algorithme SPS4

L'algorithme SPS4 de l'article est récursif basé sur la formule 3.17 qui permet de calculer un polynôme cyclotomique à partir de polynômes cyclotomiques inverses d'ordres inférieurs et de la formule 3.25 qui donne un polynôme cyclotomique inverse comme produit de polynômes cyclotomiques d'ordres inférieurs. L'algorithme alterne l'utilisation des polynômes cyclotomiques et cyclotomiques inverses.

Prenons par exemple $n = 3 \cdot 5 \cdot 7 = 105$. On obtient successivement les équations :

$$\begin{aligned}
\Phi_{105}(X) &= \Psi_{15}(X)\Psi_3(X^7)(1-X^{105})(1-X^{15})^{-1}(1-X^{21})^{-1}(1-X^{35})^{-1} \\
\Psi_{15}(X) &= \Phi_5(X)\Phi_1(X^3) \\
\Psi_3(X^7) &= \Phi_1(X^7)
\end{aligned}$$

En analysant le processus récursif, on constate qu'une seule procédure récursive est suffisante pour effectuer ces calculs si elle possède un paramètre permettant de choisir une récursion sur un polynôme cyclotomique ou sur un polynôme cyclotomique inverse.

Au terme de ce processus récursif, les seules opérations arithmétiques utilisées sont des additions et des soustractions. En effet, dans les formules 3.17 et 3.25, on n'effectue effectivement que des multiplications par des polynômes de la forme $(1-X^d)$ et des divisions par de tels polynômes. Mais diviser par $(1-X^d)$, revient à multiplier par la série formelle $\sum_{k=0}^{\infty} X^{kd}$ en s'arrêtant à un nombre de termes suffisants. C'est à dire $\varphi(n)$ pour le calcul de $\Phi_n(X)$ ou $n-\varphi(n)$ pour celui de $\Psi_n(X)$.

La procédure récursive est la suivante :

Procedure SPS4(m, e, λ, D, D_{max}, a) : Multiply by $\Phi_m(z^e)$ or $\Psi_m(z^e)$

A recursive algorithm to multiply by $\Phi_m(z)$ or $\Psi_m(z)$

Input:

- m, a positive, squarefree odd integer
- λ, a boolean
- a, an array of integers $a(0), a(1), \ldots$ satisfying $f(z) \equiv \sum a(i)z^i \pmod{z^{\lfloor \frac{D_{max}}{2} \rfloor + 1}}$ modulo $z^{\lfloor \frac{1}{2} D_{max} \rfloor} + 1$, where $f(z)$ is some product of cyclotomic polynomials
- D, the degree of $f(z)$
- D_{max}, a bound on the degree

Output:
D^*, the degree of the resulting polynomial. If λ is true, we compute $g(z) = f(z)\Phi_m(z^e)$, truncated to degree $\frac{1}{2}D_{max}$. Otherwise, we compute $g(z) = f(z)\Psi_m(z^e)$, truncated to degree $\frac{1}{2}D_{max}$. We write the coefficients of g to array a, and return the degree of g, D_g.

if λ **then** $D_g \longleftarrow D + \phi(m)e$ **else**
$D_g \longleftarrow D + (m - \phi(m))e$
`//`D^*_{max} `is our new degree bound`
$D^*_{max} \longleftarrow \min(D_g, D_{max})$

$e^* \longleftarrow e$, $m^* \longleftarrow m$, $D^* \longleftarrow D$
while $m^* > 1$ **do**
  $p \longleftarrow$ largest prime divisor of m^*
  $m^* \longleftarrow \frac{m}{p}$
  `//multiply by` $\Phi_{m^*}(z^{e^*})$ `(or` $\Psi_{m^*}(z^{e^*})$`)`
  $D^* \longleftarrow$ SPS4(m^*, e^*, not λ, D^*, D^*_{max}, a)
  $e^* \longleftarrow e^* p$
if λ **then**
  for *each prime* $p|m$ **do**
    $d \longleftarrow \frac{m \cdot e}{p}$ `//Divide by` $1 - z^{m \cdot e/p}$
    for $i = d$ **to** $\lfloor \frac{D^*_{max}}{2} \rfloor$ **do** $a(i) \longleftarrow a(i - d)$
  $d \longleftarrow m \cdot e$ `//Multiply by` $1 - z^{m \cdot e}$
  for $i = \lfloor \frac{D^*_{max}}{2} \rfloor$ **down to** d **do** $a(i) \longleftarrow a(i - d)$

`//Get higher-degree terms of` $g(z)$ `as necessary`
if $D_g \equiv 0 \bmod 2$ **then**
  for $i = \lfloor \frac{D_g}{2} \rfloor + 1$ **to** $\min(D_g, \frac{D_{max}}{2})$ **do**
  $a(i) \longleftarrow a(D_g - i)$
else
  for $i = \lfloor \frac{D_g}{2} \rfloor + 1$ **to** $\min(D_g, \frac{D_{max}}{2})$ **do**
  $a(i) \longleftarrow -a(D_g - i)$
return D_g

J'ai réalisé une implémentation en Python disponible dans le dépôt Git sps4.py. J'ai validé son

fonctionnement en vérifiant que la hauteur des polynômes calculés est bien celle fournie dans l'article d'Arnold et Monagan. Cette version n'est qu'un prototype qui pourrait-être amélioré d'une part n'allouant qu'une fois la mémoire nécessaire au calcul du polynôme cyclotomique et d'autre part en faisant appel à la compilation pour les boucles internes de calcul.

Analyser en détail le nombre de termes utilisés dans `SPS4` pour les produits par $(1-X^d)^{-1}$.

Hauteur des polynômes cyclotomiques

Dans cette section, on étudie la magnitude des coefficients des polynômes cyclotomiques $\Phi_n(X)$. Dans un premier temps, pour n premier ou produit de deux entiers premiers. Dans un second temps, on énoncera des résultats concernant la croissance asymptotique de la hauteur $A(n)$ des polynômes cyclotomiques.

On constate que pour p premier :

$$\Phi_p(X) = \frac{X^p-1}{\Phi_1(X)} = \frac{X^p-1}{X-1} = \sum_{k=0}^{p-1} X^k,$$

et tous les coefficients de $\Phi_p(X)$ sont égaux à 1. Si n est produit de deux entiers premiers p, q, on va montrer que les coefficients du polynôme cyclotomique :

$$\Phi_{pq}(X) = \frac{(1-X)(1-X^{pq})}{(1-X^p)(1-X^q)}$$

sont dans $\{-1, 0, 1\}$. Le degré de $\Phi_{pq}(X)$ est égal à $(p-1)(q-1)$. En considérant l'égalité précédente dans l'anneau $\mathbb{Z}[[X]]$, où la division par $(1-X^p)(1-X^q)$ a un sens puisque le coefficient constant est égal à un, les coefficients de $\Phi_{pq}(X)$ sont donc ceux de $(1-X)(1-X^p)^{-1}(1-X^q)^{-1}$ tronqués à l'ordre $\varphi(pq)$. Prouvons tout d'abord que les coefficients a_m de la série formelle

$$S_{pq}(X) = (1-X^p)^{-1}(1-X^q)^{-1} = \sum_{m\geq 0} a_m X^m$$

sont égaux à zéro ou un pour $0 \leq m < pq$. Il suffit pour cela de montrer qu'il existe au plus un couple d'entiers $(a, b) \in \mathbb{N}^2$ tel que $ap + bq = m$ pour $0 \leq m < pq$. Or le lemme de Gauss implique que si $ap + bq = a'p + b'q$, il existe r entier tel que $a' = a - rq$ et $b' = b + rp$. Sans perte de généralité, on peut supposer $b' \geq b$, c'est à dire $r \geq 0$. L'hypothèse $m < pq$ implique

$$pq > a'p + b'q \geq (b + rp)q \geq rpq$$

d'où $r = 0$ et $(a, b) = (a', b')$.
Les coefficients de $\Phi_{pq}(X)$ sont ceux de $(1-X)S_{pq}(X)$ tronqués à l'ordre $\varphi(pq)$. Étant différence de deux entiers de $\{0, 1\}$, ils appartiennent à $\{-1, 0, 1\}$, ce que nous souhaitions prouver. Cette démonstration s'appuie sur un sujet de préparation à l'agrégation [3]. Notons que la réciproque est fausse : $\Phi_{651} = \Phi_{3\cdot 5\cdot 31}$ est un polynôme cyclotomique d'ordre le produit de trois premiers distincts dont les coefficients sont dans $\{-1, 0, 1\}$.

On dit de Φ_{pq} dont l'ordre est produit de deux premiers distincts qu'il est binaire. Et de Φ_{pqr} qu'il est ternaire. $\Phi_{105} = \Phi_{3\cdot 5\cdot 7}$ est en quelque sorte le premier polynôme cyclotomique ternaire

d'ordre impair. -2 est l'un de ses coefficients. N'importe quel entier est-il le coefficient d'un polynôme cyclotomique ? La réponse est positive et on reproduit ici la preuve de cet article de Jiro Suzuki [5].

On note C les valeurs prises par l'ensemble des coefficients de tous les polynômes cyclotomiques et on commence par prouver que

Propriété 12
pour tout entier $t > 2$, il existe t nombres premiers distincts $p_1 < p_2 < \cdots < p_t$ tels que $p_1 + p_2 > p_t$.

Dans le cas contraire, il existerait un entier $t > 2$ tel que pour tous premiers vérifiant $p_1 < p_2 < \cdots < p_t$ on ait $p_1 + p_2 \leq p_t$ et donc $2p_1 < p_t$. Cela implique que pour tout entier k, il y a au plus t nombres premiers entre 2^{k-1} et 2^k et que $\pi(2^k) < kt$, contrairement à ce qu'affirme le théorème des nombres premiers à savoir :

$$\pi(n) \underset{n\to\infty}{\sim} \frac{n}{\ln n}.$$

Revenons à notre affirmation initiale qui spécifie que tout entier $s \in \mathbb{Z}$ est le coefficient $c_i^{(n)}$ d'un polynôme cyclotomique Φ_n. Soit t un entier impair supérieur à 2 et t nombres premiers $p_1 < p_2 < \cdots < p_t$ tels que $p_1 + p_2 > p_t$. Notons $p = p_t$ et $n = p_1 p_2 \cdots p_t$. On a

$$\Phi_n(X) = \prod_{d|n} (1 - X^d)^{\mu(\frac{n}{d})} = \sum_{i=0}^{\varphi(n)} c_i^{(n)} X^i.$$

Pour $r > s$, on a $p_r + p_s > p_t = p$ et par conséquent

$$\begin{aligned} \Phi_n(X) &\equiv \prod_{i=1}^{t} (1 - X^{p_i})/(1 - X) && \pmod{X^{p+1}} \\ &\equiv (1 + X + \cdots + X^p)(1 - X^{p_1} + X^{p_2} + \cdots + X^{p_t}) && \pmod{X^{p+1}}. \end{aligned}$$

On constate que $c_p^{(n)} = -t + 1$ et $c_{p-2}^{(n)} = -t + 2$ ce qui montre que $\{s \in \mathbb{Z} \mid s \leq -1\} \subseteq C$ puisque t est n'importe quel nombre naturel impair supérieur ou égal à 3.

On sait d'autre part que pour m entier positif impair

$$\Phi_{2m}(X) = \Phi_m(-X).$$

Si $p_1 \geq 3$, $n = p_1 p_2 \cdots p_t$ est impair, $c_p^{(2n)} = t - 1$ et $c_{p-2}^{(2n)} = t - 2$. On en déduit que $\{s \in \mathbb{Z} \mid s \geq 1\} \subseteq C$. Et finalement que $C = \mathbb{Z}$ puisque $c_2^{15} = 0$.

Dans la preuve ci-dessus, on augmente indéfiniment le nombre de facteurs premiers t de n pour exhiber un coefficient arbitrairement grand d'un polynôme cyclotomique Φ_n. Emma Lehmer démontra en 1936 [6] que les coefficients des polynômes cyclotomiques ternaires forment un ensemble infini. Pour montrer que $\limsup_{n\to\infty} A(n) = \infty$, il "suffit" de considérer les polynômes cyclotomiques ternaires.

Le mathématicien suédois Carl Severin Wigert a prouvé que :

$$\limsup_{n\to\infty} \frac{\log d(n)}{\log n / \log\log n} = \log 2$$

où $d(n)$ désigne le nombre de diviseurs de n ; voir par exemple le théorème 317 de [7]. On utilise ce résultat pour établir (en suivant le mathématicien américain Paul T. Bateman) que

Propriété 13

$$\limsup_{n\to\infty} A(n) \leq \exp(n^{\log 2/\log\log n}).$$

Partons de nouveau de l'égalité $\Phi_n(X) = \prod_{d|n}(1-X^d)^{\mu(\frac{n}{d})}$ et remarquons que

- la hauteur d'un polynôme P produit de polynômes $P_i = \sum_{j=0}^{n_i} p_j^{(i)} X^j$ est inférieure ou égale à la hauteur du polynôme Q produit des polynômes $\widetilde{P_i} = \sum_{j=0}^{n_i} |p_j^{(i)}| X^j$,
- $1/(1-X^d)$ est égale à la série formelle $\sum_{k\geq 0} X^{kd}$, et les termes de degré supérieurs à $\varphi(n)$, donc supérieurs ou égaux à n ne contribuent pas au calcul de Φ_n.

On déduit de cela que la hauteur de Φ_n est inférieure ou égal à celle du polynôme

$$\Delta_n(X) = \prod_{d|n}(1 + X^d + X^{2d} + \cdots + X^{(n/d-1)d}).$$

La hauteur d'un produit de polynômes dont les coefficients sont tous égaux à un étant moindre que le produit des valeurs de ces polynômes en un (ce que l'on prouve par récurrence sur le nombre de polynômes), on obtient en utilisant le résultat de Carl Severin Wigert mentionné plus haut :

$$\begin{aligned} A(n) &< \prod_{d|n}(n/d) = n^{d(n)/2} = \exp\left(\frac{1}{2}d(n)\log n\right) \\ &< \exp\left(\frac{1}{2}2^{(1+\epsilon/2)\log n/\log\log n}\log n\right) \\ &< \exp\left(2^{(1+\epsilon)\log n/\log\log n}\right) \\ &= \exp\left(n^{(1+\epsilon)\log 2/\log\log n}\right) \end{aligned}$$

pour tout $\epsilon > 0$ et n suffisamment grand, ce qui permet de conclure.

Le mathématicien britannique Robert Charles Vaughan a lui prouvé que pour une infinité d'entiers n

$$A(n) > \exp\left(n^{\log 2/\log\log n}\right).$$

Éléments clefs

- Utilisation de la formule d'inversion de Mobiüs.
- Et des polynômes cyclotomiques inverses.
- Formules (LEMME 1) liant les polynômes cyclotomiques et polynômes cyclotomiques inverses pour les produits d'un entier par une entier premier. Ces formules font appel à la division de polynômes.
- Pour la division de polynômes, il est possible d'utiliser un algorithme FFT. L'article décrit des calculs dans des corps de cardinal premier et reconstruction par le théorème des restes chinois. Il mentionne *"For even though the numerator is sparse, the denominator and quotient are typically dense."*

— Analyser les problèmes provenant de la taille des entiers manipulés. En particulier, `Python` utilise pour les entiers une arithmétique sans limite de taille. *Quelle est l'implication pour les performances ?*
— La formule 3.14 est importante, car elle permet de transformer une division par un polynôme cyclotomique en un produit par un polynôme cyclotomique inverse et une division par un polynôme de la forme $1 - z^m$.
— Et en utilisant des résultats de séries formelles, cela revient à effectuer une multiplication par une série $1 + z^m + z^{2m} + \ldots\ldots$ où il est nécessaire de contrôler le nombre de termes à conserver de la série.
— Les propriétés palindromiques des polynômes cyclotomiques permettent quant à elle de diviser par deux le nombre de coefficients à calculer. Le LEMME 4 étend ces propriétés aux produits de polynômes cyclotomiques (et de polynômes cyclotomiques inverses).
— Ces propriétés peuvent-être utilisées dans tous les calculs intermédiaires.
— Les formules 3.17 et 3.19 indiquent comment calculer les polynômes cyclotomiques (et cyclotomiques inverses) à partir des précédents.
— Pour de "très grands" polynômes cyclotomiques, on peut être amené à calculer "trop de termes" $\Longrightarrow$ voir remarque 3.22.
— *Analyser le dernier algorithme récursif !!!*

Algorithmes de l'article

Implémentation SAGE

SAGE implémente la méthode `cyclotomic_coeffs` ici. Lorsque la hauteur du polynôme à calculer dépasse :

```
cdef long fits_long_limit = 169828113 if sizeof(long) >= 8 else 10163195
```

SAGE bascule sur un algorithme (plus lent) basé sur PARI/GP en précision infinie.

Implémentation SymPy

De son côté, SymPy implémente la méthode dup_zz_cyclotomic_poly, à analyser. Il y a visiblement une arithmétique en précision infinie, basée sur gmpy2, *comment s'effectue le basculement ?*

Références

[1] Andrew Arnold et Michael Monagan (2011) *Calculating cyclotomic polynomials*, Mathematics of computation - Volume 80, Number 276.

[2] Andrew Arnold et Michael Monagan (2010) *A high-performance algorithmfor calculating cyclotomic polynomials*, Simon Fraser University - Burnaby, B.C. Canada.

[3] Pierre Charollois (2015) *Problème de Mathématiques Générales no. 2.*, Université Pierre et Marie Curie.

[4] Carlo Sanna (2021) *A Survey on Coefficients of Cyclotomic Polynomials*, Politecnico di Torino.

[5] Jiro Suzuki (1987) *On Coefficients of Cyclotomic Polynomials*, Department of Mathematics, Sophia University Wesley, Massachusetts, 2nd ed.

[6] Emma Lehmer (1936) *On the magnitude of the coefficient of the cyclotomic polynomial*, Bull. Amer. Math. Soc. 42(6) : 389-392.

[7] G. H. Hardy et E. M. Wright (1936) *An introduction to the theory of numbers*, Oxford University Press 4th edition.

[8] K.O. Geddes, S.R. Czapor, and G. Labahn (1992). *Algorithms for Computer Algebra*. Kluwer Academic Publishers, Boston.