HAL
open science

Calcul formel dans la base des polynômes unitaires de Chebyshev

Cuong Tran

To cite this version:

Cuong Tran. Calcul formel dans la base des polynômes unitaires de Chebyshev. Mathématiques générales [math.GM]. Université Pierre et Marie Curie - Paris VI, 2015. Français. NNT : 2015PA066379 . tel-01273287

HAL Id: tel-01273287

https://theses.hal.science/tel-01273287

Submitted on 27 Feb 2016

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

Université Paris VII - Denis Diderot

Université Pierre et Marie Curie

École Doctorale de Science Mathématiques de Paris Centre

Thèse de doctorat

Discipline : Mathématiques

présentée par

Cường TRẦN

Calcul formel dans la base des polynômes unitaires de Chebyshev

---

dirigée par Pierre-Vincent Koseleff & Fabrice Rouillier

Soutenue le 09 octobre 2015 devant le jury composé de :

M. Alin Bostan	INRIA Saclay Île-de-France	
M. Pierre-Vincent Koseleff	IMJ-PRG	
M. Sylvain Lazard	INRIA Nancy Grand Est	rapporteur
Mme Ariane Mézard	IMJ-PRG	
M. Daniel Pecker	IMJ-PRG	
M. Fabrice Rouillier	INRIA Paris-Rocquencourt	
Mme Annick Valibouze	LIP6	
M. Jacques-Arthur Weil	XLIM, Université de Limoges	rapporteur

Institut de Mathématiques de Jussieu -
Paris Rive Gauche
4, place Jussieu
75 005 Paris

UPMC
École Doctorale de Sciences
Mathématiques de Paris Centre
4 place Jussieu
75252 Paris Cedex 05
Boite courrier 290

23 octobre 2015

Kính tặng Bố.

Remerciements

Je tiens tout d'abord à exprimer ma gratitude à mes directeurs de thèse Monsieur Pierre-Vincent Koseleff et Monsieur Fabrice Rouillier. Je les remercie pour leur confiance, ainsi que leur soutien permanent depuis mon arrivée à l'IMJ-PRG. Ils se sont toujours montrés extrêmement disponibles et m'ont guidé avec beaucoup d'enthousiasme, de compréhension et de professionnalisme. Ils ont su me donner de précieux renseignements depuis mes premiers pas en Calcul Formel avec compétence et patience.

Je remercie Messieurs Sylvain Lazard et Jacques-Arthur Weil pour avoir accepté de rapporter cette thèse, pour les remarques constructives et pour l'intérêt qu'ils ont porté à mes travaux. Je tiens également à remercier Mesdames Annick Valibouze et Ariane Mézard, Messieurs Alin Bostan et Daniel Pecker pour avoir accepté de faire partie de mon jury.

Mes remerciements vont également à tout le personnel de l'équipe Analyse Algébrique qui a su apporter des réponses à mes questions, qui a toujours su être disponible pour discuter et résoudre mes problèmes, qui a contribué significativement à mes développements. Je n'oublie pas les thésards dans le couloir 15-16 : Thibaud, Malick, Rafael, Martin, Andrés, Hóa, Việt, ... Un grand merci à tout ceux à l'IMJ-PRG, qui m'ont aidé pendant ma thèse.

Je ne peux pas me permettre d'oublier de remercier Monsieur Đinh Tiến Cường. Je te remercie pour ton aide dans mes recherches ainsi que ton accompagnement dans ma vie. Je pense également au groupe "việt-upmc" et les autres amis vietnamiens que je n'ai pas cités. Mes remerciements les plus sincères vont à vous qui avez partagé mes joies ainsi que mes difficultés dans la vie en France.

Je pense maintenant à ma femme, mes enfants Bốp et Bông qui m'ont toujours encouragé et soutenu dans les moments difficiles. J'ai vécu, grâce à eux, des années inoubliables en France.

Enfin, j'ai toujours pu compter sur l'affection et le soutien de ma famille au Vietnam. Je pense à mes parents et à mon frère, que je ne remercierai jamais assez.

Résumé

Résumé

Nous proposons des méthodes simples et efficaces pour manipuler des expressions trigonométriques de la forme $F = \sum_{k=0}^{d} f_k \cos \frac{k\pi}{n}$, $f_k \in \mathbb{Z}$ où $d < n$ fixé. Nous utilisons les polynômes unitaires de Chebyshev qui forment une base de $\mathbb{Z}[x]$ avec laquelle toutes les opérations arithmétiques peuvent être exécutées aussi rapidement qu'avec le base de monômes, mais également déterminer le signe et une approximation de F, calculer le polynôme minimal de F. Dans ce cadre nous calculons efficacement le polynôme minimal de $2\cos\frac{\pi}{n}$ et aussi le polynôme cyclotomique Φ_n. Nous appliquons ces méthodes au calcul des diagrammes de nœuds de Chebyshev.

Mots-clefs

polynôme de Chebyshev, multiplication rapide, polynôme minimal, $\cos\frac{\pi}{n}$, polynôme cyclotomique, nœuds de Chebyshev,

---

Fast computing with the Chebyshev's monic polynomial

Abstract

We propose a set of simple and fast algorithms for evaluating and using trigonometric expressions in the form $F = \sum_{k=0}^{d} f_k \cos \frac{k\pi}{n}$, $f_k \in \mathbb{Z}$ where $d < n$ fixed. We make use of the monic Chebyshev polynomials as a basis of $\mathbb{Z}[X]$. We can perform arithmetic operations (multiplication, division, gcd) on polynomials expressed in a Chebyshev basis (with the same bit-complexity as in the monomial basis), compute the sign of F, evaluate it numerically and compute its minimal polynomial in $\mathbb{Q}[X]$. We propose simple and efficient algorithms for computing the minimal polynomial of $2\cos\frac{\pi}{n}$ and also the cyclotomic polynomial Φ_n. As an application, we give a method to determine the Chebyshev knot's diagrams.

Keywords

Chebyshev's monic polynomial, fast multiplication, minimal polynomial, $\cos\frac{\pi}{n}$, cyclotomic polynomial, Chebyshev knots

Table des matières

Introduction **9**

1 Polynômes **15**
1.1 Le polynôme unitaire de Chebyshev . . . 15
1.2 Le polynôme cyclotomique . . . 19
1.3 L'application Doublage . . . 26
1.4 Le polynôme minimal de $2\cos\frac{\pi}{n}$ . . . 28

2 Opérations rapides avec les formes de Chebyshev **33**
2.1 Résultats utiles ou classiques . . . 34
2.2 La multiplication et la division de formes de Chebyshev . . . 39
2.3 Stratégie "Diviser pour régner" . . . 44
2.4 Changement de base . . . 49

3 Calcul des polynômes minimaux **59**
3.1 Calcul du polynôme cyclotomique . . . 60
3.2 Calcul du polynôme minimal de $2\cos\frac{\pi}{n}$ . . . 63

4 Évaluation des expressions trigonométriques **69**
4.1 Évaluation d'une expression trigonométrique . . . 69
4.2 Calcul dans l'anneau $\mathbb{Z}[x]/\langle M_n\rangle$ . . . 78
4.3 Le polynôme minimal d'un élément de $\mathbb{Z}[2\cos\frac{\pi}{n}]$ . . . 81

5 Applications aux diagrammes de nœuds de Chebyshev **87**
5.1 Introduction . . . 88
5.2 Calcul du polynôme caractéristique . . . 90
5.3 Calculer les racines réelles de $R_{a,b,c}$ . . . 96
5.4 Calculer les diagrammes des nœuds . . . 101

A Calculs faits avec Maple 18 **105**
A.1 Le paquetage `ChebUnit` . . . 105
A.2 Trouver la forme de Chebyshev . . . 105
A.3 Minorer une somme de cosinus . . . 107

Table des figures **115**

Bibliographie **119**

Index des notations **123**

Introduction

Motivation

Les polynômes de Chebyshev sont utilisés dans de nombreux domaines des mathématiques, notamment dans le secteur de l'Analyse Numérique. Une remarque attribuée à de nombreux mathématiciens et numériciens, citée au début de [MH03], témoigne de cette importance : *"Chebyshev polynomials are everywhere dense in numerical analysis."*

Il est possible de lister plusieurs secteurs des mathématiques où les polynômes de Chebyshev jouent des rôles importants : théorie de l'interpolation, polynômes orthogonaux, théorie des approximations, intégration numérique, analyse numérique, théorie d'ergodique, etc. [Riv90, Préface]. Mais aussi en théorie des nœuds où il est démontré que : *tous les nœuds sont des nœuds de Chebyshev* [KP11].

Le point de départ du travail effectué dans cette thèse est : *déterminer le signe d'une expression trigonométrique de la forme* $F = \sum_{k=0}^{d} \hat{f}_k \cos k\frac{\pi}{n}, \hat{f}_k \in \mathbb{Q}$, *pour* $d < n$ *fixé dans* $\mathbb{Z}_{>0}$. Bien entendu, cette question est classique et il s'agit de l'évaluation du nombre algébrique réel $\cos\frac{\pi}{n}$ en un polynôme de degré d.

Nous traitons d'une question plus générale qui est celle de manipuler de telles expressions.

Il apparaît rapidement que le nombre algébrique $2\cos\frac{\pi}{n}$ est plus adapté que $\cos\frac{\pi}{n}$ car son polynôme minimal M_n est unitaire à coefficients entiers. Il convient alors de considérer plutôt les polynômes unitaires de Chebyshev $T_n(2\cos x) = 2\cos nx$ et l'expression à étudier devient alors $F = f_0 + \sum_k f_k T_k(2\cos\frac{\pi}{n})$.

Les polynômes unitaires de Chebyshev forment une base de $\mathbb{Z}[x]$ qui est particulièrement adaptée au calcul du polynôme minimal M_n de $2\cos\frac{\pi}{n}$. Nous montrons que nous pouvons effectuer les opérations arithmétiques usuelles de $\mathbb{Z}[x]$ (multiplication, division, pgcd) avec la même complexité qu'avec la base des monômes.

Dans ce contexte, nous montrons que nous pouvons décider de la nullité de F en $\widetilde{\mathcal{O}}(n^2\tau)$ opérations binaires et que nous pouvons calculer le polynôme minimal de F en $\widetilde{\mathcal{O}}(n^3\tau)$ opérations binaires, où τ est la taille binaire des f_k.

Ces résultats sont ensuite utilisés pour étudier les diagrammes des nœuds de Chebyshev. Il s'agit de décider si la courbe gauche $\mathcal{C}(a, b, c, \phi) : x = T_a(t), y = T_b(t), z = T_c(t+\phi)$ admet des points multiples, et dans la négative, de déterminer son diagramme.

Contributions et plan de la thèse

La thèse comporte cinq parties.

Dans la première partie, nous décrivons les polynômes unitaires de Chebyshev, les polynômes cyclotomiques et certaines de leurs propriétés. Nous introduisons l'application $\mathcal{D}$ et ces propriétés, mettons en évidence le lien entre le polynôme minimal M_n de $2\cos\frac{\pi}{n}$

et le polynôme cyclotomique Φ_{2n}. Nous rappelons que les sommes de Newton de Φ_n sont aussi des sommes de Ramanujan.

La famille des polynômes unitaires de Chebyshev forme une base orthogonale de $\mathbb{Z}[x]$ et elle a l'avantage de rendre les calculs beaucoup plus concis que fait dans la base des polynômes de Chebyshev de type original. Avec cette base nous pouvons exécuter toutes les opérations arithmétiques dans $\mathbb{Z}[x]$ aussi rapidement qu'avec la base de monômes, ce que nous décrivons dans la seconde partie.

En introduisant l'application *doublage*

$$\begin{aligned}\mathcal{D} : \mathbb{Q}[x] &\to \mathbb{Q}[x]\\ P &\mapsto x^{\deg P} \cdot P(x + \tfrac{1}{x})\end{aligned}$$

nous transférons toutes les complexités classiques du calcul dans la base des monômes à la base des polynômes unitaires de Chebyshev. À titre de comparaison, l'application $\mathcal{D}$ permet de décrire très simplement l'algorithme de multiplication de deux polynômes exprimés dans la base des polynômes de Chebyshev, proposé par Giorgi en 2012 [Gio12], voir la sous-section 2.2.1.

Dans la seconde partie, nous rappelons aussi les résultats classiques et utiles de complexité des opérations arithmétiques de base dans $\mathbb{Z}[x]$.

Nous proposons des algorithmes pour passer de la base des monômes à la base des polynômes unitaires de Chebyshev et inversement, dans la partie 2.4. Nous considérons ces transformations comme des cas particuliers de composition polynomiale, développée par Hart et Novocin dans [HN11] (2011), et basée sur la stratégie "diviser pour régner", dont nous rappelons le principe en section 2.3.

Nous démontrons que les changements de base demandent une complexité binaire de $\widetilde{\mathcal{O}}(d^2 + d\tau)$ où d, τ sont le degré et la taille des coefficients. Nous passons de la base des monômes à la base des polynômes unitaires de Chebychev (algorithme 2.31) en adaptant l'algorithme de composition polynomiale et en utilisant les opérations "rapides" dans la base des polynômes unitaires de Chebyshev. A l'inverse, en gardant la structure de l'algorithme de composition, la particularité des polynômes unitaires de Chebyshev permet d'utiliser la récurrence matricielle introduite dans [BSS10] (2010) pour alléger le calcul du changement de base entre la base de Chebyshev et la base des monômes (algorithme 2.36).

Dans la troisième partie, nous proposons les algorithmes de calcul des polynômes minimaux M_n et Φ_n. Nous avons trouvé une bonne méthode pour obtenir M_n. L'algorithme 3.14 proposé dans le Chapitre 3 permet de calculer M_n dans la base des polynômes de Chebyshev en $\widetilde{\mathcal{O}}(n_0^2)$ opérations binaires (où n_0 est la partie radicale impaire de n). C'est une adaptation de l'algorithme donné dans [AM10] (2010) qui concerne le polynôme cyclotomique Φ_n, le polynôme minimal des racines primitives n-ième de l'unité. La relation entre Φ_n et M_n est très simple :

$$\Phi_{2n} = \mathcal{D}(M_n).$$

Nous en déduisons que nous pouvons également calculer la représentation dans la base des monômes du polynôme cyclotomique Φ_n en $\widetilde{\mathcal{O}}(n_0^2)$ opérations binaires.

Une fois que les calculs dans la base des polynômes unitaires de Chebyshev sont résolus, nous retournons à la première question de déterminer le signe de F.

Puisque $F = f(2\cos\frac{\pi}{n})$ où $f = f_0 + \sum_{k=1}^{d} f_k T_k$, $f_k \in \mathbb{Z}$, $d < \frac{n}{2}$ l'évaluation de F peut être considérée comme l'évaluation de $f \in \mathbb{Q}[x]$ en une valeur réelle $2\cos\frac{\pi}{n}$. Tester si $F = 0$ peut amener à décider si le polynôme minimal $M_n \in \mathbb{Q}[x]$ de $2\cos\frac{\pi}{n}$ est diviseur de f.

Grâce au calcul de M_n et aux opérations rapides dans base des polynômes unitaires de Chebyshev, notre première question concernant la détermination le signe de F est résolue avec l'algorithme 4.15 en complexité binaire $\widetilde{\mathcal{O}}(n^2\tau)$. Cela est mieux que la méthode d'isolation, même si l'on applique les résultats les plus modernes, qui demande une complexité de $\widetilde{\mathcal{O}}(n^3 + n^2\tau)$, voire l'algorithme 4.9.

Comme extension du calcul du polynôme minimal M_n de $2\cos\frac{\pi}{n}$, nous cherchons le polynôme minimal M_F de F dans $\mathbb{Q}[x]$. Calculer ce polynôme minimal est équivalent à trouver le polynôme minimal M_f de $f \in \mathbb{Q}[x]/(M_n)$. Il est possible d'appliquer quelques résultats des plus récents utilisant les résultants, mais cette méthode demanderait une complexité $\widetilde{\mathcal{O}}(n^4 + n^3\tau)$. Nous préférons exposer une méthode basée sur le calcul des sommes de Newton d'un polynôme annulateur P_f de f, considéré comme un polynôme à coefficients dans $\mathbb{Z}[2\cos\pi/n]$. Grâce aux particularités de l'ensemble des racines de P_f et aussi aux calculs rapides dans la base des polynômes unitaires de Chebyshev, nous pouvons calculer le polynôme M_F en complexité binaire $\widetilde{\mathcal{O}}(n^3\tau)$, où τ est un majorant de la taille binaire des f_k.

Nous appliquons enfin nos résultats à la question du calcul des diagrammes des nœuds de Chebyshev, dans la partie 5. Il a été démontré [Vas90] que tous les nœuds $K \subset \mathbb{R}^3 \subset \mathbf{S}^3$ peuvent être obtenus à l'aide des plongements polynomiaux $t \mapsto (f(t), g(t), h(t)), t \in \mathbb{R}$. Très peu d'exemples ont été donnés. C'est un problème difficile de trouver pour un nœud donné une paramétrisation polynomiale.

Il a été démontré [KP11] que tout nœud $K \subset \mathbb{R}^3 \subset \mathbf{S}^3$ est un nœud de Chebyshev. Malheureusement, il n'existe pas de borne a priori sur le degré du plongement polynomial. Nous avons construit un algorithme permettant de calculer tous les diagrammes de nœuds de Chebyshev possibles pour un triplet (a, b, c) donné en $\widetilde{\mathcal{O}}(n^2)$ opérations binaires, où $n = abc$.

En réalité, le calcul des diagrammes des nœuds de Chebyshev est lié à l'étude des solutions d'un système algébrique de dimension zéro et de degré $\frac{(a-1)(b-1)(c-1)}{2}$, où l'on cherche les valeurs de ϕ qui rendent la courbe $\mathcal{C}(a, b, c, \phi)$ singulière. Ces valeurs sont annulées par un polynôme à coefficients entiers $R_{a,b,c}$, que l'on pourrait calculer en utilisant des bases de Gröbner. Nous montrons en fait que ce polynôme se factorise en produit de polynôme de degré 1 ou 2 à coefficients dans $\mathbb{Z}[2\cos\frac{\pi}{n}]$, où $n = abc$. Nous montrons que nous pouvons calculer ce polynôme en $\widetilde{\mathcal{O}}(n^4)$ opérations binaires dans la base des polynômes de Chebyshev, mais que nous pouvons calculer plus rapidement avec des approximations numériques de ses coefficients entiers en $\widetilde{\mathcal{O}}(n^3)$ opérations binaires.

Ces sont les racines de ce polynôme qui nous intéressent et nous montrons par la suite, que nous pouvons les obtenir en $\widetilde{\mathcal{O}}(n^2)$ opérations binaires, parce que nous montrons que les coefficients de $R_{a,b,c}$ sont de taille binaire $\widetilde{\mathcal{O}}(n)$ et ses racines sont bien séparées.

Ce nouvel algorithme permet de donner une compléxité $\widetilde{\mathcal{O}}(n^2)$ pour la détermination de tous les diagrammes de nœud possibles $\mathcal{C}(a, b, c, \phi)$ où a, b et c sont fixés et $n = abc$. Il apparaît, dans la partie 5.4, qu'il est moins coûteux de décrire tous les diagrammes possibles $\mathcal{C}(a, b, c, \phi)$, que de calculer un seul diagramme $\mathcal{C}(a, b, c, \phi)$ pour un ϕ donné.

Nous avons également codé deux paquetages sous Maple qui s'appellent `ChebUnit` et `NoeudCheb` où sont implantés certains algorithmes proposés.

Les résultats des chapitres 1, 2, 3 et 4 sont publiés pour leur plus grande part dans [KRT15], l'article a été accepté à être présenté à ISSAC '15 (juillet 2015, Bath, UK).

Les résultats du chapitre 5 sont intégrés dans un article traitant du calcul des diagrammes de nœuds de Chebyshev, qui est en cours de finalisation [KPRT].

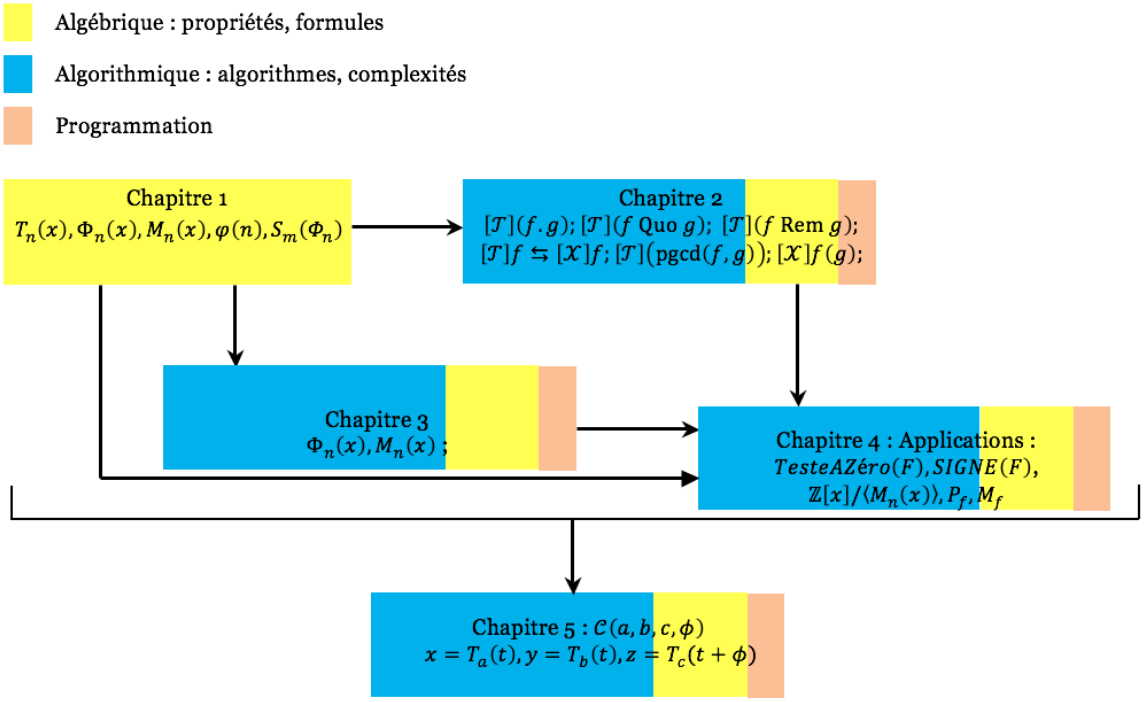


FIGURE 1 – Plan de Thèse.

Chapitre 1

Polynômes

Sommaire

1.1 Le polynôme unitaire de Chebyshev 15
1.1.1 Définitions 16
1.1.2 Propriétés fondamentales 16
1.2 Le polynôme cyclotomique 19
1.2.1 La fonction d'Euler et la fonction de Möbius 20
1.2.2 Propriétés des polynômes cyclotomiques 22
1.2.3 Les sommes de Newton et le polynôme cyclotomique 24
1.3 L'application Doublage 26
1.4 Le polynôme minimal de $2\cos\frac{\pi}{n}$ 28
1.4.1 Le polynôme minimal de $2\cos\frac{\pi}{n}$ et le polynôme cyclotomique . . 28
1.4.2 La factorisation des polynômes unitaires de Chebyshev 29

Résumé : Dans ce chapitre, nous introduisons les polynômes unitaires de Chebyshev et leurs propriétés principales.
L'un des nos objectifs est le calcul de M_n, le polynôme minimal dans $\mathbb{Q}[x]$ de $2\cos\frac{\pi}{n}$. Il est similaire à $\Phi_n(x)$, le n-ième polynôme cyclotomique, qui est classique.
En développant l'idée initiée par Rivlin (1990), nous utilisons l'application *Doublage* qui vérifie $\mathcal{D}(M_n) = \Phi_{2n}$ et permet une équivalence entre la plupart des propriétés de Φ_n et celles de M_n parmi lesquelles, la factorisation des polynômes unitaires de Chebyshev ou encore l'expression de leurs sommes de Newton.

Notations

$\mathbb{N}$ désigne l'ensemble de nombres naturels ; $\mathbb{Z}$ est l'anneau des entiers ; $\mathbb{Q}$ est le corps des nombres rationnels ; $\mathbb{R}$ est le corps des nombres réels ; $\mathbb{C}$ est le corps des nombres complexes.

Soit $P \in \mathbb{R}[x]$ un polynôme univarié, $\deg P$ est le degré de P, $\mathrm{lc}(P)$ est son coefficient dominant.

1.1 Le polynôme unitaire de Chebyshev

L'expression *polynômes de Chebyshev* désigne en général les polynômes $\mathbf{T}_n(x) = \cos(nt)$ et $\mathbf{U}_n(x) = \dfrac{\sin(n+1)t}{\sin t}$, où $x = \cos t$. Leurs coefficients dominants sont alors des puissances de 2.

1.1.1 Définitions

Ici, nous considérerons plutôt les *polynômes unitaires de Chebyshev* .

Définition 1.1 (Polynôme unitaire de Chebyshev)**.** Définissons deux suites $(T_n, n \in \mathbb{N})$ et $(U_n, n \in \mathbb{N})$, par :

$$T_n(x) = 2\cos(nt), \quad U_n(x) = \frac{\sin nt}{\sin t}$$

où $x = 2\cos t$.

Les éléments de l'ensemble $(T_n)_{n\in\mathbb{N}}$ s'appellent polynômes unitaires de Chebyshev de première espèce ou également *polynômes de Dickson* (voir [BC12]). Les éléments de l'ensemble $\mathcal{U} = (U_n)_{n\in\mathbb{N}}$, s'appellent *polynômes unitaires de Chebyshev de seconde espèce.*

En utilisant les relations

$$\cos(n+1)t + \cos(n-1)t = 2\cos t \cdot \cos nt, \quad \sin(n+1)t + \sin(n-1)t = 2\cos t \cdot \sin nt,$$

on peut voir que les deux ensembles sont déterminés par une même relation de récurrence :

$$P_{n+1}(x) = x \cdot P_n(x) - P_{n-1}(x), \tag{1.1}$$

avec comme conditions initiales : $\{T_0 = 2, T_1 = x\}$, $\{U_0 = 0, U_1 = 1\}$.

Remarque 1.2. Dans ce travail, les deux lettres T, U sont réservées pour indiquer les polynômes unitaires, alors les caractères gras $\mathbf{T}, \mathbf{U}$ vont être utilisés chaque fois que l'on parle du type original (non forcément unitaire) défini classiquement par

$$\mathbf{T}_n(x) = \cos nt, \quad \mathbf{U}_n(x) = \frac{\sin(n+1)t}{\sin t} \quad \text{avec } x = \cos t,$$

elles satisfont la relation de récurrence $P_{n+1}(x) = 2x \cdot P_n(x) - P_{n-1}(x)$ avec comme conditions initiales $(\mathbf{T}_0 = 1, \mathbf{T}_1 = x)$, $(\mathbf{U}_0 = 1, \mathbf{U}_1 = 2x)$ (voir [MH03, Chap. 1]), ou encore :

$$2\mathbf{T}_n(x) = T_n(2x), T_n(x) = 2\mathbf{T}_n(\tfrac{x}{2}) \quad ; \quad \mathbf{U}_{n-1}(x) = U_n(2x), U_n(x) = \mathbf{U}_{n-1}(\tfrac{x}{2}).$$

1.1.2 Propriétés fondamentales

Pour $n > 0$, le polynôme T_n est unitaire de degré n. La famille $\mathcal{T} = (1, T_n, n > 0)$ est une base de $\mathbb{Z}[x]$. La relation entre les deux familles $\mathcal{T}$ et $\mathcal{X} = (x^n)_{n\in\mathbb{N}}$ est donnée par :

Lemme 1.3. *Pour tout $n \in \mathbb{N}$ on a :*

$$x^n = \sum_{k=0}^{\lfloor n/2 \rfloor - 1} \tbinom{n}{k} T_{n-2k}(x) + \tfrac{1+(-1)^n}{2} \tbinom{n}{\lfloor \frac{n}{2} \rfloor}; \tag{1.2a}$$

$$T_n(x) = \sum_{k=0}^{\lfloor n/2 \rfloor} (-1)^k \tfrac{n}{n-k} \tbinom{n-k}{k} x^{n-2k}. \tag{1.2b}$$

Démonstration.

1. Comme $2\cos t = e^{\imath t} + e^{-\imath t}$, alors :

$$\begin{aligned} 2^n \cos^n t &= \sum_{k=0}^{n} \tbinom{n}{k} e^{\imath(n-k)t} e^{-\imath kt} = \sum_{k=0}^{n} \tbinom{n}{k} e^{\imath(n-2k)} \\ &= \sum_{2k<n} \tbinom{n}{k} (\cos(n-2k)t + \imath \sin(n-2k)t) + \sum_{2k=n} \tbinom{n}{k} e^0 \\ &\quad + \sum_{2k>n} \tbinom{n}{k} (\cos(n-2k)t + \imath \sin(n-2k)t), \end{aligned}$$

Ceci équivaut à :

$$2^n \cos^n t = \sum_{k=0}^{\lfloor n/2 \rfloor - 1} 2\tbinom{n}{k} \cos(n-2k)t + \tfrac{1+(-1)^n}{2} \tbinom{n}{\lfloor \frac{n}{2} \rfloor},$$

ce qui induit l'identité (1.2a).

2. Nous montrons que T_n est la solution d'une équation différentielle linéaire du second ordre, l'identification permet alors d'établir une relation de récurrence entre ses coefficients.
Partant de $T_n(2\cos t) = 2\cos nt$, on a $T_n'(2\cos t)\sin t = n \sin nt$, et donc

$$T_n'(x) = nU_n(x). \tag{1.3}$$

En dérivant encore nous obtenons :

$$T_n''(2\cos t)(-2\sin t) \cdot \sin t + T_n'(2\cos t) \cdot \cos t = n^2 \cos nt,$$

avec $x = 2\cos t$, ce qui nous donne :

$$(4 - x^2) \cdot T_n''(x) - x \cdot T_n'(x) + n^2 T_n(x) = 0. \tag{1.4}$$

Supposons que T_n s'écrive $T_n = \sum_{k=0}^{n} a_k x^{n-k}$ ($a_0 = 1$), alors l'identité (1.4) donne :

$$\begin{aligned} n^2 T_n(x) &= \sum_{k=0}^{n} n^2 a_k x^{n-k} = n^2 x^n + n^2 a_1 x^{n-1} + \sum_{k=2}^{n} n^2 a_k x^{n-k}, \\ -x \cdot T_n'(x) &= \sum_{k=0}^{n} -(n-k) a_k x^{n-k} = -nx^n - (n-1)a_1 x^{n-1} + \sum_{k=2}^{n} -(n-k) a_k x^{n-k}, \\ (4-x^2) \cdot T_n''(x) &= \sum_{k=2}^{n} 4(n-k+2)(n-k+1) a_{k-2} x^{n-k} - \sum_{k=0}^{n-2} (n-k)(n-k-1) a_k x^{n-k} \\ &= -n(n-1)x^n - (n-1)(n-2)a_1 x^{n-1} + \\ &\quad + \sum_{k=2}^{n} [4(n-k+2)(n-k+1)a_{k-2} - (n-k)(n-k-1)a_k] x^{n-k} \end{aligned}$$

Dans (1.4), le coefficient de x^{n-1} est

$$0 = n^2 a_1 - (n-1)a_1 - (n-1)(n-2)a_1,$$

donc $a_1 = 0$. Pour $k = 2, \ldots, n$, l'expression du coefficient x^{n-k}, induit :

$$0 = n^2 a_k - (n-k)a_k + 4(n-k+2)(n-k+1)a_{k-2} - (n-k)(n-k-1)a_k,$$

et

$$a_k = -4\frac{(n-k+2)(n-k+1)}{k(2n-k)}a_{k-2}.$$

Mais comme $a_1 = 0$, tous les coefficients d'indice impair sont nuls. Pour les indices pairs $2k$, $(k = 0, \dots, \lfloor \frac{n}{2} \rfloor)$:

$$a_{2k} = -\frac{(n-2k+2)(n-2k+1)}{k(n-k)}a_{2k-2},$$

et on en déduit que

$$a_{2k} = (-1)^k a_0 \frac{n \dots (n-2k+1)}{k!(n-1)\dots(n-k)} = (-1)^k \tfrac{n}{n-k}\tbinom{n-k}{k},$$

ce qui prouve l'identité (1.2b).

□

Remarque 1.4.

1. Les formules (1.2b) et (1.3) fournissent l'expression de U_n dans la base des monômes :

$$U_n(x) = \sum_{k=0}^{\lfloor \frac{n-1}{2} \rfloor} (-1)^k \tbinom{n-k-1}{k} x^{n-2k-1}. \tag{1.5}$$

2. D'après l'équation différentielle (1.4), T_n est une fonction hypergéométrique[1] (voir [OLBC10, Sec. 15.9.5]), et on a :

$$2T_n(x) = F(-n, n, \tfrac{1}{2}\,; \tfrac{2-x}{4}).$$

n	$x^n =$	$T_n =$
0	1	2
1	T_1	x
2	$T_2 + 2$	$x^2 - 2$
3	$T_3 + 3T_1$	$x^2 - 3x$
4	$T_4 + 4T_2 + 6$	$x^4 - 4x^2 + 2$
5	$T_5 + 5T_3 + 10T_1$	$x^5 - 5x^3 + 5x$
6	$T_6 + 6T_4 + 15T_2 + 20$	$x^6 - 6x^4 + 9x^2 - 2$

TABLE 1.1 – 7 premières éléments de base.

Étudions maintenant les zéros des polynômes unitaires de Chebyshev.

Lemme 1.5 (Racines des polynômes unitaires de Chebyshev)**.** *Soit $n \in \mathbb{N}^*$, alors T_n et U_n se factorisent en :*

$$T_n(x) = \prod_{k=0}^{n-1}\left(x - 2\cos\tfrac{(2k+1)\pi}{2n}\right) \quad ; \quad U_n(x) = \prod_{k=1}^{n-1}\left(x - 2\cos\tfrac{k\pi}{n}\right) \tag{1.6}$$

1. La fonction hypergéométrique $F(a, b; c; z)$ est définie par la suite de Gauss :

$$F(a,b;c;z) = \sum_{s=0}^{\infty} \frac{(a)_s(b)_s}{(c)_s s!} z^s = 1 + \frac{ab}{c}z + \frac{a(a+1)b(b+1)}{c(c+1)\cdot 2!}z^2 + \dots$$

sur le disque $|z| < 1$ et par ailleurs, sa continuation analytique.

Démonstration. Rappelons que $T_n(x) = 2\cos nt, U_n(x) = \frac{\sin nt}{\cos t}$, avec $x = 2\cos t$.

1. La suite $\cos\frac{(2k+1)\pi}{2n}$, $k = 0, \dots, n-1$ contient clairement n éléments distincts. De plus, pour chaque $k = 0, \dots, n-1$:
$$T_n(2\cos\tfrac{(2k+1)\pi}{2n}) = 2\cos\tfrac{(2k+1)n\pi}{2n} = 0.$$
Comme $\deg(T_n) = n$, alors cette suite caractérise entièrement les zéros de T_n.
2. De la même façon, on obtient les zéros de U_n par :
$$U_n(2\cos\tfrac{k\pi}{n}) = \frac{\sin\frac{nk\pi}{n}}{\sin\frac{k\pi}{n}} = 0,\ k = 1, \dots, n-1.$$

□

Nous utiliserons essentiellement les polynômes unitaires de Chebyshev de première espèce. Rappelons quelques propriétés qui nous seront utiles par la suite :

Lemme 1.6. *Pour tout $m, n \in \mathbb{Z}$ on a :*

$$T_n \circ T_m = T_m \circ T_n = T_{mn}; \tag{1.7a}$$

$$T_n \cdot T_m = T_{m+n} + T_{m-n}; \tag{1.7b}$$

$$T_n\left(x + \tfrac{1}{x}\right) = x^n + \tfrac{1}{x^n}. \tag{1.7c}$$

Démonstration. Comme deux polynômes de même degré d sont identiques s'ils coïncident en une infinité de points :

$$\deg T_n \circ T_m = \deg T_m \circ T_n = mn = \deg T_{mn},$$

$$\deg T_m \cdot T_n = m + n = \deg(T_{m+n} + T_{m-n}),\ \text{et}$$
$$\deg(x^n \cdot T_n(x + \tfrac{1}{x})) = 2n = \deg(x^{2n} + 1).$$

Pour tout $x \in (-2, 2)$, il existe $t \in (0, \pi)$ tel que $x = 2\cos t$:

1. $T_m(T_n(x)) = T_m(T_n(2\cos t)) = T_m(2\cos nt) = 2\cos mnt = T_n(2\cos mx) = T_n(T_m(x))$, $T_{mn}(x) = T_{mn}(2\cos t) = 2\cos mnt$, on a donc (1.7a).
2. $T_n(x)\cdot T_m(x) = 4\cos nt\cdot\cos mt = 2\cos(m+n)t + 2\cos(m-n)t = T_{m+n}(x) + T_{m-n}(x)$. On obtient alors (1.7b).
3. Considérons deux applications f_1, f_2 définies par $f_1(x) = x^n \cdot T_n(x + \frac{1}{x})$, $f_2 = x^{2n} + 1$. Comme $\deg T_n = n$, f_1, f_2 sont les polynômes de degré $2n$ dans $\mathbb{Z}[x]$.
Pour tout nombre complexe $z = e^{\imath t}$, on obtient par la formule d'Euler :
$f_1(z) = e^{\imath nt}T_n(e^{\imath t} + e^{-\imath t}) = e^{n\imath t}T_n(2\cos t) = e^{n\imath t}(2\cos nt) = e^{n\imath t}(e^{n\imath t} + e^{-n\imath t}) = e^{2n\imath t} + 1 = f_2(z)$,
mais $\deg f_1 = \deg f_2$, ils sont identiques, cela entraîne (1.7c).

□

1.2 Le polynôme cyclotomique

Nous rappelons ici quelques résultats sur les polynômes cyclotomiques qui nous seront utiles par la suite et qui peuvent être retrouvés, par exemple, dans [Riv90, Dem08, Ber14].

1.2.1 La fonction d'Euler et la fonction de Möbius

Une fonction α définie sur $\mathbb{N}$ à valeurs dans $\mathbb{C}$ s'appelle *fonction arithmétique*. Si, de plus, $\alpha(m \cdot n) = \alpha(m) \cdot \alpha(n)$ pour tout m, n étant premiers entre eux, on dit que α est une fonction *arithmétique multiplicative*.

Nous considérons deux fonctions arithmétiques importantes, la fonction d'Euler $\varphi(n)$ et la fonction de Möbius $\mu(n)$.

Définition 1.7.

1. Une solution complexe de l'équation $x^n - 1 = 0$ s'appelle racine n-ième de l'unité. L'ensemble de racines n-ièmes de l'unité sera noté
$$G_n = \{\xi^k, k = 0, \dots, n-1\} \text{ où } \xi = \exp(\tfrac{2\pi\imath}{n}),$$
où $\imath$ vérifie $\imath^2 = -1$.
G_n est un groupe multiplicatif ;
2. Si ρ est une racine de l'unité, l'entier m strictement positif le plus petit tel que $\rho^m = 1$ s'appelle l'*ordre de* ρ, noté par $\mathrm{ord}(\rho)$;
Si $\mathrm{ord}(\rho) = n$, on dit que ρ est une racine primitive n-ième de l'unité.
L'ensemble de racines primitives n-ièmes de l'unité sera noté H_n.

Remarque 1.8.

1. Pour tout $k \in \mathbb{N}$:
$$\mathrm{ord}(\xi^k) = \tfrac{n}{(n,k)}; \tag{1.8}$$
Soit d un nombre entier positif tel que $(\xi^k)^d = 1$, alors $n \mid kd$, donc $\frac{n}{(n,k)} \mid \frac{k}{(n,k)}d$, et par conséquent $\frac{n}{(n,k)} \mid d$. L'identité $(\xi^k)^{\frac{n}{(n,k)}} = (\xi^n)^{\frac{k}{(k,n)}} = 1$, démontre (1.8)
2. Supposons que α, β soient deux racines de l'unité, $\mathrm{ord}(\alpha) = p$, $\mathrm{ord}(\beta) = q$, $(p, q) = 1$ et que $\mathrm{ord}(\alpha\beta) = m$ alors : $(\alpha\beta)^{mp} = 1 \Rightarrow \beta^{mp} = 1 \Rightarrow q \mid mp$ ou $q \mid m$; de la même façon $p \mid m$ alors $pq \mid m$. Mais $(\alpha\beta)^{pq} = 1$ donc $m \mid pq$. On écrit :
$$\text{Si } (\mathrm{ord}(\alpha), \mathrm{ord}(\beta)) = 1 \text{ alors } \mathrm{ord}(\alpha\beta) = \mathrm{ord}(\alpha)\mathrm{ord}(\beta). \tag{1.9}$$
3. G_n est un groupe cyclique engendré par ξ, $|G_n| = n$. L'ensemble de générateurs est $H_n = \{\xi^k, (k, n) = 1\}$, d'après l'équation (1.8).

La fonction d'Euler

Pour tout $n \in \mathbb{N}$, le nombre d'entiers $k : 1 \leq k \leq n$, $(k, n) = 1$ sera noté $\varphi(n)$. L'application $n \mapsto \varphi(n)$ s'appelle *la fonction d'Euler*. D'après la remarque 1.8, $\varphi(n) = |H_n|$ pour tout n.

Proposition 1.9. *φ est une fonction arithmétique multiplicative [Dem08, Prop. 2.4].*

Démonstration. Soit p, q quelconques tels que $(p, q) = 1$, nous prouvons la proposition en donnant une bijection entre H_n et $H_p \times H_q$, où $n = pq$: sur l'ensemble H_n, définissons l'application
$$h : z \mapsto (z^q, z^p),$$
par l'identité (1.8) on a $\mathrm{ord}(z^q) = p$, $\mathrm{ord}(z^p) = q$, c'est-à-dire que l'ensemble d'arrivée de h est bien $H_p \times H_q$.

Comme $(p, q) = 1$, il existe $u, v \in \mathbb{Z}$ tels que $u \cdot p + p \cdot v = 1$ (théorème de Bézout).

∘ h est injective : supposons que $z, z' \in H_n$ vérifient $h(z) = h(z')$.

$$\begin{cases} z^p = z'^p \\ z^q = z'^q \end{cases} \Rightarrow z = z^{up+vq} = z'^{up+vq} = z'.$$

∘ h est surjective : soit (α, β) un élément arbitraire de $H_p \times H_q$. Posons $z = \alpha^v \beta^u$.

$up + qv = 1 \Rightarrow \begin{cases} (v,p) = 1 \overset{(1.8)}{\Longrightarrow} \mathrm{ord}(\alpha^v) = p \\ (u,q) = 1 \overset{(1.8)}{\Longrightarrow} \mathrm{ord}(\beta^u) = q \end{cases} \overset{(1.9)}{\Longrightarrow} \mathrm{ord}(z) = p.q = n$ ou $z \in H_n$. De plus $z^q = \alpha^{1-up}\beta^{uq} = \alpha$, $z^p = \alpha^{up}\beta^{1-uq} = \beta$, et donc $h(z) = (\alpha, \beta)$.

Ainsi, h est une bijection de H_{pq} à $H_p \times H_q$, ce qui complète la preuve. □

Pour tout nombre premier p et tout entier $k > 0$,

$$\varphi(p) = p - 1, \quad \varphi(p^k) = p^{k-1}(p-1).$$

Si $n = p_1^{\alpha_1} \dots p_k^{\alpha_k}$ alors, en appliquant la proposition 1.9 $k-1$ fois pour $p_1^{\alpha_1}, p_1^{\alpha_1}p_2^{\alpha_2}, \dots, n$ nous obtenons l'expression de $\varphi(n)$:

$$\varphi(n) = n \prod_{j=1}^{k} (1 - \tfrac{1}{p_j}). \tag{1.10}$$

Remarque 1.10.

1. Pour tout $n > 2$: $(k, n) = 1$ si et seulement si $(n-k, n) = 1$ et $k \neq n-k$, alors $\varphi(n)$ est toujours un nombre pair ;
2. En utilisant (1.10), il vient $m \mid n \Rightarrow \varphi(m) \mid \varphi(n)$.

La formule suivante sera utilisé régulièrement :

$$n = \sum_{d|n} \varphi(d). \tag{1.11}$$

Preuve de la formule (1.11). En notant $n = |G_n|$, $\varphi(d) = |H_d|$,

— Pour tout $d \mid n$, on a $H_d \subset G_n$;

— Soit $\alpha \in G_n$, alors $\alpha = \xi^k$. Posons $d = \frac{n}{(n,k)}$. Alors $d \mid n$ et et $\alpha \in H_d$.

On en déduit :

$$G_n = \bigcup_{d|n} H_d,$$

Enfin, $H_d \cap H_{d'} = \emptyset$ pour tout $d \neq d'$, ce qui implique la formule (1.11). □

La fonction de Möbius

Supposons que $n = p_1^{\alpha_1} \dots p_k^{\alpha_k}$ où $p_1, \dots, p_k$ sont des nombres premiers distincts, on définit :

$$\mu(n) = \begin{cases} 0 & \text{si } \max\{\alpha_1, \dots, \alpha_k\} > 1 \\ (-1)^k & \text{sinon} \end{cases}.$$

Alors, μ est aussi une fonction arithmétique multiplicative. De plus :

$$\sum_{d|n} \mu(d) = \begin{cases} 1 & \text{si } n = 1 \\ 0 & \text{si } n > 1. \end{cases} \tag{1.12}$$

Preuve de la formule (1.12). Si $n = 1$, la formule (1.12) est clairement vraie.

Sinon, $n = p_1^{\alpha_1} \dots p_k^{\alpha_k} > 1$, on pose $n_0 = p_1 \dots p_k$. L'entier d est un diviseur sans facteur carré de n si et seulement si $d \mid n_0$. Ainsi :

$$\begin{aligned}\sum_{d|n} \mu(d) = \sum_{d|n_0} \mu(d) &= \sum_{j=0}^{k} \sum_{1 \le k_1 < \dots < k_j \le k} \mu(p_{k_1} \dots p_{k_j}) \\ &= \sum_{j=0}^{k} \tbinom{k}{j} (-1)^j = (1-1)^k = 0\end{aligned}$$

□

Grâce à l'identité (1.12), on déduit *la formule d'inversion de Möbius* (1.13). Étant donnée $f : \mathbb{N} \to \mathbb{C}$ une fonction arithmétique, en posant

$$g(n) = \sum_{d|n} f(d),$$

on obtient :

$$f(n) = \sum_{d|n} \mu(d) g(\tfrac{n}{d}). \tag{1.13}$$

Preuve de la formule (1.13). Soit d un diviseur de n, appliquons la formule (1.12) pour $\frac{n}{d}$:

$$\begin{aligned}f(n) &= \sum_{d'|n} \Big(\sum_{d|\frac{n}{d'}} \mu(d) \Big) f(d') = \sum_{dd'|n} \mu(d) f(d') \\ &= \sum_{d|n} \mu(d) \Big(\sum_{d'|\frac{n}{d}} f(d') \Big) = \sum_{d|n} \mu(d) g(\tfrac{n}{d}).\end{aligned}$$

□

De la même façon, si l'on considère $G(n) = \prod_{d|n} f(d)$, alors $\log G(n) = \sum_{d|n} \log f(d)$, la formule d'inversion devient :

$$\log f(n) = \sum_{d|n} \log G(\tfrac{n}{d}) \mu(d),$$

et par conséquent :

$$f(n) = \prod_{d|n} G(\tfrac{n}{d})^{\mu(d)}.$$

1.2.2 Propriétés des polynômes cyclotomiques

Par essence, le n-ième polynôme cyclotomique peut être défini en fonction des racines primitives n-ièmes de l'unité :

Définition 1.11. Soit $n \ge 1$ alors le n-ième polynôme cyclotomique $\Phi_n(x)$ est le polynôme :

$$\Phi_n(x) = \prod_{\substack{1 \le k \le n \\ (k,n)=1}} (x - \exp(\tfrac{2\imath k \pi}{n})). \tag{1.14}$$

Comme $G_n = \bigcup_{d|n} H_d$ et que $H_d \cap H_{d'} = \emptyset$ pour tout $d \neq d'$, alors :

$$x^n - 1 = \prod_{d|n} \Phi_d(x). \tag{1.15}$$

Par l'inversion de Möbius :

$$\Phi_n(x) = \prod_{d|n} (x^{n/d} - 1)^{\mu(d)}. \tag{1.16}$$

Lemme 1.12. *Pour tout $n \in \mathbb{N}$, $\Phi_n(x)$ est unitaire à coefficients entiers [Dem08, Prop. 9.10].*

Démonstration. Supposons que $\Phi_d \in \mathbb{Z}[x]$ pour tout $d < n$. Notons $\Psi_n = \prod_{d|n, d<n} \Phi_d$, qui est un polynôme unitaire à coefficients entiers.
Ψ_n étant unitaire, divisons $x^n - 1$ par Ψ_n : il existe $q, r \in \mathbb{Z}[x]$, q unitaire, $\deg r < \deg \Psi_n$ tel que $x^n - 1 = q \cdot \Psi_n + r$.
Comme $x^n - 1 = \Phi_n \Psi_n$ alors $r = \Psi_n \cdot (\Phi_n - q)$, et comme $\deg r < \deg \Psi_n$, alors nécessairement $r = q - \Phi_n = 0$ et donc $\Phi_n = q \in \mathbb{Z}[x]$.
Par induction, $\Phi_n \in \mathbb{Z}[x]$ pour tout n. □

Lemme 1.13 (Gauss)**.** *[Dem08, Lemme 9.4] Pour tout $n \geq 1$, Φ_n est le polynôme minimal de ξ dans $\mathbb{Q}[x]$.*

Démonstration. Soit $\Phi(x) \in \mathbb{Z}[x]$ le polynôme minimal de ξ sur $\mathbb{Q}$. Puisque $\Phi_n(\xi) = 0$ alors $\Phi \mid \Phi_n$ dans $\mathbb{Q}[x]$ et Φ est unitaire. Ainsi, $\Phi_n = \Phi \cdot Q$ avec Q unitaire dans $\mathbb{Z}[x]$. Soit p un nombre premier ne divisant pas n. ξ^p est d'ordre n (remarque 1.8) et est racine de Φ. Si $\Phi(\xi^p) \neq 0$ alors $Q(\xi^p) = 0$ et Φ divise $Q(x^p)$ dans $\mathbb{Z}[x]$ (Lemme de Gauss).

Considérons la projection modulo p, $\pi : \mathbb{Z}[x] \to F_p[x]$. C'est un morphisme d'anneau et $\pi(\phi)$ divise $\pi(Q(x^p))$ dans $F_p[x]$. Mais $P \mapsto P^p$ étant une application linéaire dans $F_p[x]$, on en déduit que $\pi(Q(x^p)) = \pi(Q)^p$ et tout facteur irréductible A de $\pi(\phi)$ divise $\pi(Q)$. Ainsi A^2 divise $\pi(\Phi_n)$, $\pi(\Phi_n)$ a donc un facteur multiple au même titre que $\pi(X^n - 1)$, ce qui est impossible car $X^n - 1$ et nX^{n-1} sont premiers entre-eux modulo p.

Par conséquent, $\Phi(\xi^p) = 0$ si $(p, n) = 1$. On déduit alors que pour tout nombre k premier avec n, $\Phi(\xi^k) = 0$ et donc Φ admet comme racines toutes les racines primitives n-ièmes de l'unité. $\Phi = \Phi_n$ est donc irréductible. □

Considérons l'application :

Définition 1.14. REV est la transformation de $\mathbb{Q}[x]$ dans lui-même définie par

$$\begin{aligned} \mathrm{REV} : \mathbb{Q}[x] &\to \mathbb{Q}[x] \\ P &\mapsto x^{\deg P} P(\tfrac{1}{x}), \end{aligned}$$

Si $\xi_k = \xi^k$ est une racine primitive n-ième de l'unité, $(k, n) = 1 \Rightarrow (n - k, n) = 1$, alors $\xi_k^{-1} = \xi^{n-k}$ est aussi racine primitive n-ième de l'unité. Alors, $\Phi_n(\xi_k^{-1}) = 0$, $\mathrm{REV}(\xi_k) = 0$ et l'on a

$$\Phi_n \mid \mathrm{REV}(\Phi_n).$$

Par ailleurs, $\deg \Phi_n = \varphi(n)$ est pair (remarque 1.10), la relation entre les racines et les coefficients nous permet de calculer le coefficient constant de Φ_n qui vaut alors :

$$(-1)^{\deg \Phi_n} \prod_{(k,n)=1} \xi^k = \prod_{\substack{(k,n)=1 \\ k<\frac{n}{2}}} \xi^k \xi^{n-k} = 1,$$

$\mathrm{REV}(\Phi_n)$ est donc unitaire. De plus, $\deg \mathrm{REV}(\Phi_n) = \deg \Phi_n$ et :

Remarque 1.15. $\Phi_n = \mathrm{REV}(\Phi_n)$. On dit que Φ_n est *réversible*.

1.2.3 Les sommes de Newton et le polynôme cyclotomique

Soit $P \in K[x]$ de degré d, dont les racines sont $\{x_1, \ldots, x_d\}$. Définissons la m-ième somme de Newton de P par :

$$S_m(P) = \sum_{j=1}^{d} x_j^m.$$

Notons Newton(P) la série formelle

$$\text{Newton}(P) = \sum_{j \geq 0} S_j(P) x^j.$$

Les relations entre les sommes de Newton de P et ses coefficients sont classiques et connues sous le nom de *formules de Newton* (voir [McD99, Chap. 1][BGVPS05]) :

Théorème 1.16 (Formules de Newton)**.** *[BGVPS05, Lem. 2] Soit $P = x^d + A_1 x^{d-1} + \ldots + A_d$. Pour tout $1 \leq i \leq d$:*

$$iA_i + S_1(P)A_{i-1} + \ldots + S_{i-1}(P)A_1 + S_i(P) = 0; \tag{1.17a}$$

$$\textit{ou bien} \quad A_i = -\tfrac{1}{i}\Big[S_1(P)A_{i-1} + \ldots + S_{i-1}(P)A_1 + S_i(P)\Big]. \tag{1.17b}$$

On a en plus une belle relation reliant P et Newton(P) :

Lemme 1.17. *Soit P un polynôme unitaire de degré d dans $\mathbb{R}[x]$ alors :*

$$\text{REV}(P) = \exp\Big(\int \tfrac{1}{x}(d - \textit{Newton}(P))\Big) \tag{1.18}$$

Démonstration. Partons de la factorisation de P : $P(x) = \prod_{j=1}^{d}(x - x_j)$ on a :

$$\text{REV}(P) = \prod_{j=1}^{d}(1 - xx_j) = (-1)^d \prod_{j=1}^{d} x_j \prod_{j=1}^{d}(x - \tfrac{1}{x_j}),$$

En dérivant deux cotés on obtient :

$$\text{REV}(P)' = (-1)^d \prod_{j=1}^{d} x_j \sum_{j=1}^{d} \prod_{i \neq j}(x - \tfrac{1}{x_i}),$$

On en déduit :

$$\begin{aligned}\frac{\text{REV}(P)'}{\text{REV}(P)} &= -\sum_{j=1}^{d} \frac{x_j}{1 - xx_j} = -\sum_{k \geq 0}\Big(\sum_{j=1}^{d} x_j^{k+1}\Big)x^k \\ &= \frac{d - \text{Newton}(P)}{x}\end{aligned}$$

En prenant l'intégrale de deux côtés puis en utilisant la définition de Newton(P), on obtient l'identité (1.18). □

La somme de Newton $S_m(\Phi_n)$ s'appelle aussi la *somme de Ramanujan* (notée $c_m(n)$ dans [HW08, chap. 6]). L'ensemble de racines de Φ_n est $\mathcal{Z}(\Phi_n) = \{\exp(2k\frac{\pi\imath}{n}), (k,n) = 1\}$, et :

$$S_m(\Phi_n) = \sum_{\substack{1 \leq k \leq n \\ (k,n)=1}} \xi_k^m = \sum_{\substack{1 \leq k \leq n \\ (k,n)=1}} \exp(m \cdot \tfrac{k.2\pi\imath}{n}). \tag{1.19}$$

La valeur de la somme de Ramanujan est connue.

Théorème 1.18 (Hölder)**.** *[HW08, Théo. 272] Soit n, m des entiers positifs et $d = \frac{n}{(n,m)}$, alors la somme de Ramanujan $S_m(\Phi_n)$ peut être déterminée par :*

$$S_m(\Phi_n) = \mu(d)\frac{\varphi(n)}{\varphi(d)}. \tag{1.20}$$

Démonstration. Considérons la somme de Newton $S_m(x^n - 1) = \sum_{j=1}^{n}(\xi_n^j)^m$. Alors $(1 - \xi_n^m)S_m(x^n - 1) = 1 - \xi_n^{mn}$ donc, si $n \nmid m$, on a $S_m(x^n - 1) = 0$ et si $n|m$ alors $S_m(x^n - 1) = \sum_{k=0}^{n-1} 1 = n$. D'autre part, $x^n - 1 = \prod_{d|n} \Phi_d(x)$, si bien que :

$$S_m(x^n - 1) = \sum_{d|n} S_m(\Phi_d),$$

et donc, par l'inversion de Möbius :

$$S_m(\Phi_n) = \sum_{d|n} S_m(x^d - 1)\mu(\tfrac{n}{d}) = \sum_{d|(m,n)} d\mu(\tfrac{m}{d}). \tag{1.21}$$

Si l'on fixe m, alors l'application $a \mapsto S_m(\Phi_a)$ est une fonction arithmétique multiplicative. En fait, grâce à l'identité (1.9) $((\mathrm{ord}(\alpha), \mathrm{ord}(\beta)) = 1 \Rightarrow \mathrm{ord}(\alpha\beta) = \mathrm{ord}(\alpha)\mathrm{ord}(\beta))$ on a : si $(p, q) = 1$

$$\{\xi : \Phi_{pq}(\xi) = 0\} = \{\xi_1\xi_2 : \Phi_p(\xi_1) = \Phi_q(\xi_2) = 0\},$$

ce qui implique

$$S_m(\Phi_{pq}) = S_m(\Phi_p) \cdot S_m(\Phi_q).$$

Partant du premier cas où $n = p^\alpha$, l'identité (1.21) nous donne :

$$S_m(\Phi_{p^\alpha}) = \begin{cases} \varphi(p^\alpha) & \text{si } p^\alpha \mid m, \\ -p^{\alpha-1} & \text{si } p^{\alpha-1} \mid m \text{ et } p^\alpha \nmid m, \\ 0 & \text{sinon} \end{cases} \tag{1.22}$$

Notons la factorisation de n par $n = p_1^{\alpha_1} \dots p_k^{\alpha_k}$ et le nombre de diviseurs primitifs de n par $\omega(n)$.

- Supposons que $n = p_1^{\alpha_1}$ et considérons (1.22) dans le détail :
  - Si $p_1^{\alpha_1} \mid m$, alors $d = \frac{n}{(n,m)} = 1$, ce qui implique que

$$\mu(d)\frac{\varphi(n)}{\varphi(d)} = \varphi(n) = \varphi(\phi_1^{\alpha_1}) = S_m(\Phi_n);$$

  - Si $p_1^{\alpha_1-1} \mid m$ avec $p_1^{\alpha_1} \nmid m$ alors $d = \frac{n}{(n,m)} = p_1$, et on a :

$$\mu(d)\frac{\varphi(n)}{\varphi(d)} = \mu(p_1)\frac{\varphi(p_1^{\alpha_1})}{\varphi(p_1)} = (-1) \cdot \frac{p_1^{\alpha_1-1}(p_1 - 1)}{p_1 - 1} = -p_1^{\alpha_1-1} = S_m(\Phi_n);$$

  - Dans le cas restant, m contient au plus $\alpha_1 - 2$ fois p_1 dans sa factorisation et $d\frac{n}{(n,m)} = \frac{p_1^{\alpha_1}}{(n,m)}$ est un multiple de p_1^2. Ainsi, $\mu(d) = 0$, ce qui implique aussi :

$$\mu(d)\frac{\varphi(n)}{\varphi(d)} = 0 = S_m(\Phi_n);$$

Les trois possibilités ci-dessus impliquent à la fois que la formule (1.20) est vraie pour tout n, $\omega(n) = 1$;

- Supposons qu'elle soit vraie pour tout n, $\omega(n) \leq k-1$. Si $\omega(n) = k$ nous écrivons $n = p \cdot q$ avec $p = (p_1^{\alpha_1} \dots p_{k-1}^{\alpha_{k-1}})$, $q = p_k^{\alpha_k}$. Comme $(p,q) = 1$ on a donc :

$$\begin{cases} (m,pq) \mid m \\ (m,pq) \mid pq \end{cases} \overset{(p,q)=1}{\Longrightarrow} \begin{cases} (m,pq) \mid m \\ \left[\begin{array}{l} (m,pq) \mid p \\ (m,pq) \mid q \end{array}\right. \end{cases} \Rightarrow \left[\begin{array}{l} (m,pq) \mid (m,p) \\ (m,pq) \mid (m,q) \end{array}\right. \Rightarrow (m,pq) \mid (m,p)\cdot(m,q);$$

Inversement, lorsque $(p,q) = 1$, $\Big((m,p),(m,q)\Big) = 1$ et $(m,p) \mid (m,pq)$, $(m,q) \mid (m,pq)$ si bien que $(m,p)\cdot(m,q) \mid (m,pq)$, ce qui donne :

$$(m,pq) = (m,p)\cdot(m,q), \text{ en plus } \Big(\tfrac{p}{(m,p)}, \tfrac{q}{(m,q)}\Big) = 1.$$

On sait que les trois fonctions φ, μ et $n \mapsto S_m(\Phi_n)$ sont toutes multiplicatives, ce qui entraîne :

$$\begin{aligned} S_m(\Phi_n) = S_m(\Phi_p)\cdot S_m(\Phi_q) &= \mu(\tfrac{p}{(m,p)})\frac{\varphi(p)}{\varphi(\frac{p}{(m,p)})}\cdot\mu(\tfrac{q}{(m,q)})\frac{\varphi(q)}{\varphi(\frac{q}{(m,q)})} \\ &= \mu\Big(\frac{pq}{(m,p)\cdot(m,q)}\Big)\frac{\varphi(pq)}{\varphi(\frac{pq}{(m,p)\cdot(m,q)})} \\ &= \mu\Big(\frac{pq}{(m,pq)}\Big)\frac{\varphi(pq)}{\varphi(\frac{pq}{(m,pq)})} = \mu\Big(\frac{n}{(m,n)}\Big)\frac{\varphi(n)}{\varphi(\frac{n}{(m,n)})}, \end{aligned}$$

et donc, (1.20) est vraie pour tout n, $\omega(n) = k$.

□

Remarque 1.19. Comme $\varphi(m) \mid \varphi(n)$ si $m \mid n$ (remarque 1.10), $S_m(\Phi_n) \in \mathbb{Z}$ est entier pour tout $m = 1, \dots, \varphi(n)$, et de plus :

$$S_m(\Phi_n)|\varphi(n) \Rightarrow |S_m(\Phi_n)| < n \Rightarrow \tau(S_m(\Phi_n)) \leq \log_2 n \text{ [2]} \tag{1.23}$$

Lorsque $m = 1$, (1.20) devient :

$$S_1(\Phi_n) = \sum_{\substack{1\leq k\leq n \\ (k,n)=1}} \exp\tfrac{k.2\pi\imath}{n} = \mu(n); \tag{1.24}$$

D'autres propriétés de Φ_n sont utiles pour le calcul. Elles seront expliquées dans une autre partie où l'on présente le calcul du polynôme cyclotomique (sous-section 3.1).

1.3 L'application Doublage

La définition de T_n nous conduit à l'identité (1.7c) : $T_k(x+\frac{1}{x}) = x^k + \frac{1}{x^k}$. Il est naturel de considérer l'application *doublage* :

Définition 1.20. $\mathcal{D}$ est la transformation de $\mathbb{Q}[x]$ dans l'espace des polynômes à coefficients rationnels, de degré pair et réversibles, déterminée par :

$$\mathcal{D}(P) = x^{\deg P}\cdot P(x+\tfrac{1}{x}).$$

2. τ est la taille binaire, qui sera mentionnée dans la Définition 2.1

Remarque 1.21. Si $P \in \mathbb{Q}[x]$ est donné dans la base des polynômes unitaires de Chebyshev :

$$P(x) = p_0 + \sum_{j=1}^{n} p_j T_j,$$

on déduit immédiatement :

$$\mathcal{D}(P) = p_0 x^n + \sum_{j=1}^{n} p_j (x^{n+j} + x^{n-j}).$$

Examinons quelques propriétés utiles de la transformation $\mathcal{D}$.

Lemme 1.22. *Pour tout $f, g \in \mathbb{Q}[x]$, on a*

$$\mathcal{D}(f) \cdot \mathcal{D}(g) = \mathcal{D}(f \cdot g); \tag{1.25}$$

Si l'on a en plus $\deg f = \deg g$ *alors :*

$$\mathcal{D}(f + g) = \mathcal{D}(f) + \mathcal{D}(g). \tag{1.26}$$

Démonstration. Les deux polynômes $\mathcal{D}(f) \cdot \mathcal{D}(g)$ et $\mathcal{D}(f \cdot g)$ sont de même degré $2(\deg f + \deg g)$; ils prennent la même valeur pour tout $x \neq 0$ puisque :

$$\begin{aligned} \mathcal{D}(f)(x) \cdot \mathcal{D}(g)(x) &= x^{\deg f} \cdot x^{\deg g} \cdot f(x + \tfrac{1}{x}) \cdot g(x + \tfrac{1}{x}) \\ &= x^{\deg f + \deg g} \cdot (f \cdot g)(x + \tfrac{1}{x}) = \mathcal{D}(f \cdot g)(x). \end{aligned}$$

Ils sont donc identiques.

L'identité (1.26) est une conséquence directe de la définition de $\mathcal{D}$, dès lors que $\deg f = \deg g$. □

Supposons que $g \neq 0$, $\deg g \leq \deg f$, q, r sont le quotient (noté $\mathrm{Quo}(f, g)$) et le reste (noté $\mathrm{Rem}(f, g)$) de la division euclidienne f par g, c'est-à-dire que $q, r \in \mathbb{Q}[x]$, $\deg r < \deg g$ et :

$$f = g \cdot q + r. \tag{1.27}$$

Substituons x par $x + \frac{1}{x}$ dans de l'équation (1.27) puis multiplions des deux côtes par $x^{\deg f}$ on obtient :

$$\begin{aligned} x^{\deg f} \cdot f\left(x + \tfrac{1}{x}\right) = \left(x^{\deg g} \cdot g\left(x + \tfrac{1}{x}\right)\right) \left(x^{\deg f - \deg g} \cdot q\left(x + \tfrac{1}{x}\right)\right) \\ + x^{\deg f - \deg r} \cdot \left(x^{\deg r} \cdot r\left(x + \tfrac{1}{x}\right)\right), \end{aligned}$$

ce qui implique :

$$\mathcal{D}(f) = \mathcal{D}(g) \cdot \mathcal{D}(q) + x^{\deg f - \deg r} \cdot \mathcal{D}(r). \tag{1.28}$$

Comme $\deg r < \deg g \Rightarrow \deg f - \deg r \geq \deg f - \deg g + 1$, alors :

$$\mathcal{D}(f) \equiv \mathcal{D}(g) \cdot \mathcal{D}(q) \pmod{x^{\deg f - \deg g + 1}}. \tag{1.29}$$

Lemme 1.23. *Soit f, g deux polynômes dans $\mathbb{Q}[x]$ avec $\deg g \geq 1$, alors $g \mid f$ si et seulement si $\mathcal{D}(g) \mid \mathcal{D}(f)$.*

Démonstration. Si $f = g \cdot q$ alors $\mathcal{D}(f) = \mathcal{D}(g) \cdot \mathcal{D}(q)$ et $\mathcal{D}(g) \mid \mathcal{D}(q)$;

Inversement, si $\mathcal{D}(g)|\mathcal{D}(f)$, en considérant $r = \mathrm{Rem}(f, g)$, l'identité (1.28) fournit :

$$\frac{\mathcal{D}(f)}{\mathcal{D}(g)} - \mathcal{D}(q) = \frac{x^{\deg f - \deg r} \cdot \mathcal{D}(r)}{\mathcal{D}(g)} \in \mathbb{Q}[x].$$

D'après la remarque 1.21, le coefficient constant de $\mathcal{D}(g)$ est le coefficient dominant de g qui n'est pas nul, donc $(\mathcal{D}(g), x^{\deg f - \deg r}) = 1$. Ainsi $\mathcal{D}(g)$ divise $\mathcal{D}(r)$ dans $\mathbb{Q}[x]$, mais comme $\deg \mathcal{D}(r) < \deg \mathcal{D}(g)$, alors $\mathcal{D}(r) = 0$ et $r = 0$. □

Terminons cette section par une conséquence importante des résultats précédents :

Corollaire 1.24. *Soit f et g deux polynômes dans $\mathbb{Z}[x]$. Alors on a :*

$$\mathcal{D}((f, g)) = (\mathcal{D}(f), \mathcal{D}(g)); \tag{1.30}$$

Si plus u, v sont leurs coefficients de Bézout, i.e. $f \cdot u + g \cdot v = (f, g)$ alors

$$\mathcal{D}(f) \cdot \mathcal{D}(u) + \mathcal{D}(g) \cdot \mathcal{D}(v) = \mathcal{D}((f, g)). \tag{1.31}$$

Démonstration. (1.30) est un corollaire direct du lemme 1.23 ;

L'expression (1.31) est une conséquence de la formule (1.26). En effet, $(f, g) = f \cdot u + g \cdot v$, et comme $\deg(f, g) \leq \deg f \leq \deg(f \cdot u)$, alors $\deg(f, g) \leq \deg(g \cdot v)$, ainsi $\deg(f \cdot u) = \deg(g \cdot v)$. □

1.4 Le polynôme minimal de $2 \cos \frac{\pi}{n}$

Le polynôme minimal de $\cos \frac{2\pi}{n}$ apparait régulièrement dans la littérature : dans son livre [Riv90, Chapitre 5], Rivlin considère le polynôme minimal de $\cos \frac{2\pi}{n}$ comme un facteur du polynôme de Chebyshev de seconde espèce. Watkin et Zeitlin [WZ93] l'ont identifié comme facteur de $\mathbf{T}_{\lceil (n+1)/2 \rceil} - \mathbf{T}_{\lfloor (n-1)/2 \rfloor}$; Récemment, Bayard & Cangul [BC12] ont publié une version modifiée de cette factorisation et ont exprimé ce polynôme minimal à l'aide de l'inversion de Möbius.

1.4.1 Le polynôme minimal de $2 \cos \frac{\pi}{n}$ et le polynôme cyclotomique

Définition 1.25. Pour $n \in \mathbb{N}_{>1}$, on définit M_n comme étant le polynôme minimal de $2 \cos \frac{\pi}{n}$ dans $\mathbb{Q}[x]$. On posera, par convention, $M_1(x) = 1$.

L'application $\mathcal{D}$, va nous permettre d'associer Φ_n et M_n et va nous permettre, entre autres, de factoriser complètement les polynômes unitaires de Chebyshev. Ceci complètera une suite de résultats connus : en 1984, Hsiao a donné la factorisation de $\mathbf{T}_n$ dans [Hsi84] ; en 1990, Rivlin [Riv90] qui a donné celle de $\mathbf{U}_n$.

Il est très intéressant que tous les coefficients de M_n dans la base des polynômes unitaires de Chebyshev est exactement ceux de Φ_{2n} dans la base des monômes, ce qui suivra le fait que :

Lemme 1.26. *Pour tout $n \in \mathbb{N}$, $n \geq 3$, on a $\Phi_{2n} = \mathcal{D}(M_n)$.*

Démonstration. D'après la remarque 1.15, $\Phi_{2n} \in \mathbb{Z}[x]$ est un polynôme unitaire réversible ; De plus, la remarque 1.10 indique que $\deg \Phi_{2n}$ est un entier pair si bien que l'on peut poser $\deg \Phi_{2n} = 2K$ $(K \in \mathbb{N})$.

Supposons que

$$\Phi_{2n}(x) = 1 + x^{2K} + \sum_{j=1}^{K-1} c_j \left(x^j + x^{2K-j}\right) + c_K x^K, c_j \in \mathbb{Z}.$$

Considérons le polynôme dans $\mathbb{Z}[x]$, $m_n = c_K + \sum_{j=1}^{K-1} c_j T_{K-j} + T_K$. Par la remarque 1.21 :

$$\mathcal{D}(m_n) = \Phi_{2n}.$$

Φ_{2n} est irréductible. D'après le lemme 1.22, m_n est également irréductible.

Enfin, en utilisant l'identité (1.7c), on a :

$$m_n(2\cos\tfrac{\pi}{n}) = m_n\Big(\exp(\tfrac{\pi\imath}{n}) + \exp(\tfrac{-\pi\imath}{n})\Big) = \exp(-K\tfrac{\pi\imath}{n}) \cdot \Phi_{2n}\Big(\exp(\tfrac{2\pi\imath}{2n})\Big) = 0.$$

Ainsi m_n est un polynôme unitaire irréductible dans $\mathbb{Z}[x]$ qui s'annule en $2\cos\frac{\pi}{n}$. Il est donc nécessairement identique à M_n. □

Remarque 1.27. L'idée d'utiliser la relation $\Phi_{2n} = \mathcal{D}(M_n)$ peut être trouvée dans [Riv90, Sec. 5.3].

Le lemme 1.26 nous permet de décrire immédiatement l'ensemble de racines de M_n :

Corollaire 1.28. *Pour tout $n \geq 3$, M_n est un polynôme unitaire dans $\mathbb{Z}[x]$ de degré $\frac{\varphi(2n)}{2}$; Ses racines sont $2\cos\frac{k\pi}{n}$, $(k, 2n) = 1$, autrement dit*

$$M_n = \prod_{\substack{1\leq k\leq n \\ (k,2n)=1}} (x - 2\cos\tfrac{k\pi}{n}) \tag{1.32}$$

1.4.2 La factorisation des polynômes unitaires de Chebyshev

Pour la factorisation, nous montrons d'abord qu'il sera suffisant de considérer les M_n d'indices impairs :

Lemme 1.29 (M_n avec n pair)**.** *Soit n un nombre naturel non nul que l'on écrit $n = 2^k n_0$ où $k \geq 1$, n_0 impair. Alors*

$$M_n = \begin{cases} T_{2^{k-1}} & \text{si } n_0 = 1 \\ M_{n_0} \circ T_{2^k} & \text{si } n_0 > 1 \end{cases}.$$

Démonstration. Nous utilisons le même argument pour les deux cas : si un polynôme unitaire dans $\mathbb{Z}[x]$ de degré $\frac{\varphi(2n)}{2}$, s'annule en $2\cos\frac{\pi}{n}$, alors il est égal à M_n.

Par exemple pour cas où $n_0 = 1$ (c'est-à-dire $n = 2^k$) :

$T_{2^{k-1}}(2\cos\frac{\pi}{n}) = 2\cos\frac{\pi}{2} = 0$;

$\deg T_{2^{k-1}} = 2^{k-1} = \frac{\varphi(2^k)}{2}$;

$\mathrm{lc}(M_n) = \mathrm{lc}(T_{2^{n-1}}) = 1$; □

La proposition qui suit reprend les résultats de Hsiao [Hsi84] et de Rivlin, [Riv90, Chapitre 5] pour les polynômes de Chebyshev de première espèce $\mathbf{T}_n$ et de seconde espèce $\mathbf{U}_n$:

Proposition 1.30. *Pour tout $n \in \mathbb{N}^*$ si on écrit $n = 2^k n_0$ avec n_0 impair, alors T_n et U_n se factorisent comme suit :*

$$T_n(x) = \prod_{d|n_0} M_{d2^{k+1}}(x); \tag{1.33a}$$

$$U_n(x) = (-1)^{\frac{n_0-1}{2}} \prod_{d|n_0} \Big[M_d(-x) \prod_{j=0}^{k} M_{d2^j}(x) \Big] \tag{1.33b}$$

Démonstration. Il est nécessaire d'utiliser la formule d'Euler $n = \sum_{d|n} \varphi(d)$ (1.11) plusieurs fois.

- Factorisation de T_n : Pour tout $d|n_0$:

$$T_n(2\cos\tfrac{\pi}{d2^{k+1}}) = 2\cos\tfrac{n\pi}{d2^{k+1}} = 2\cos(\tfrac{n_0}{d})\tfrac{\pi}{2} = 0 \Rightarrow M_{d.2^{k+1}} \mid T_n.$$

La famille $\{M_{d.2^{k+1}}, d|n_0\}$ ne contient que des polynômes irréductibles distincts qui sont tous unitaires et diviseurs de T_n.
La somme des degrés de ces facteurs est

$$\begin{aligned}\sum_{d|n_0} \deg M_{d.2^{k+1}} &= \sum_{d|n_0} \tfrac{\varphi(d.2^{k+2})}{2} \\ &= \sum_{d|n_0} \varphi(d)\tfrac{\varphi(2^{k+2})}{2} = 2^k \sum_{d|n_0} \varphi(d) \\ &= 2^k n_0 = \deg T_n,\end{aligned}$$

ce qui implique la formule (1.33a).

- Factorisation de U_n : Pour tout $d|n_0$ et $j = 1, \dots, k$:

$$U_n(2\cos\tfrac{\pi}{d2^j}) = \frac{\sin n\frac{\pi}{d2^j}}{\sin\frac{\pi}{d2^j}} = \frac{1}{\sin\frac{\pi}{d2^j}} \sin(\tfrac{n_0}{d}2^{k-j}\pi) = 0, \text{ donc } M_{d.2^j}|U_n;$$

Pour tout $1 < d|n_0$, on définit $\overline{M}_d(x) = M_d(-x)$, le polynôme minimal de $-2\cos\frac{\pi}{d} = 2\cos\frac{(d-1)\pi}{d}$. La valeur de U_n en ce point est :

$$U_n(2\cos\tfrac{(d-1)\pi}{d}) = \frac{\sin n\frac{(d-1)\pi}{d}}{\sin\frac{(d-1)\pi}{d}} = \frac{1}{\sin\frac{(d-1)\pi}{d}} \sin(\tfrac{n_0}{d}(d-1)2^k\pi) = 0, \text{ donc } \overline{M}_d|U_n;$$

$2\cos\frac{(d-1)\pi}{d}$ n'est pas racine de M_d et donc $(M_d, \overline{M}_d) = 1$. La famille $\{\overline{M}_d, M_d, M_{2^j.d}, d \mid n_0, j = 1, \dots, k\}$ ne comprend que des polynômes unitaires irréductibles distincts qui sont tous diviseurs de U_n.
La somme de degrés des polynômes de cette famille est :

$$\begin{aligned}\sum_{d|n_0}^{k}[\deg \overline{M}_d + \sum_{j=0}^{k} \deg M_{d.2^j}] &= -1 + \sum_{d|n_0}(\tfrac{\varphi(2d)}{2} + \sum_{j=0}^{k} 2^j\varphi(d)) = \\ &= -1 + \sum_{d|n_0}(\tfrac{\varphi(d)}{2} + \sum_{j=0}^{k}\varphi(d)2^{j-1}) = -1 + 2^k \sum_{d|n_0}\varphi(d) = \\ &= n - 1 = \deg U_n\end{aligned}$$

Enfin calculons le signe du produit : pour chaque $1 < d \mid n_0$, afin de rendre $\overline{M}(x) = M_d(-x)$ unitaire, il faut changer son signe $\deg M_d = \frac{\varphi(2d)}{2} = \frac{\varphi(d)}{2}$ fois. Le dernier signe sera donc :

$$(-1)^{\sum_{1<d|n_0}\frac{\varphi(d)}{2}} = (-1)^{\frac{n_0-1}{2}}.$$

□

Pour $x = 2\cos t$, en posant

$$V_n(x) = \frac{\cos(n+\frac{1}{2})t}{\cos(t/2)},$$

alors V_n ressemble au polynôme de Chebyshev original de troisième espèce (voir [MH03, Chap. 1]). Comme

$$\begin{aligned} T_{2n+1}(x) &= 2\cos(2n+1)t = 2\cos t \cdot \frac{\cos(n+\frac{1}{2})2t}{\cos t} \\ &= x \cdot V_n(2\cos 2t) = x \cdot V_n(T_2(x)), \end{aligned}$$

alors, en substituant n par $2n+1$ dans la formule (1.33a) on obtient :

$$\begin{aligned} x \cdot V_n(T_2(x)) = T_{2n+1}(x) &= \prod_{d|2n+1} M_{2d}(x) \\ &= x \cdot \prod_{\substack{d|2n+1 \\ d>1}} M_d(T_2(x)). \end{aligned}$$

Ainsi V_n et $\prod_{d|2n+1} M_d$ sont de même degré et prennent la même valeur pour tout $x = 2\cos(2t)$, $t \in (0, \frac{\pi}{2})$. Ces deux polynômes sont donc identiques. Enfin, pour tout $n \geq 3$:

$$V_n(x) = \prod_{d|2n+1} M_d(x). \tag{1.34}$$

Conclusion du chapitre 1

1. Les propriétés du polynôme de Chebyshev peuvent être appliquées pour le polynôme unitaire de Chebyshev.
2. Le polynôme cyclotomique est un objet classique de l'algèbre qui nous permet d'établir les propriétés les plus importantes des polynômes minimaux des nombres algébriques mentionnés dans ce travail. Nous avons rappelé des propriétés utiles de ce polynôme (section 1.2) parmi lesquelles l'expression de ses sommes de Newton.
3. En introduisant la nouvelle application $\mathcal{D}$ (section 1.3), nous pouvons faire immédiatement la transformation aller-retour des opérations arithmétiques entre la base des monômes et la base des polynômes unitaires de Chebyshev à l'aide de la remarque clé 1.21.
4. Le polynôme minimal M_n de $2\cos\frac{\pi}{n}$ est l'antécédent de Φ_{2n} par $\mathcal{D}$. Cette relation, connue de Rivlin [Riv90], permet à l'auteur de retrouver les propriétés du polynôme minimal M_n (section 1.4) et également les factorisations (1.33) de T_n et U_n, donnant une nouvelle forme aux résultats de Hsiao (1984) et de Rivlin (1990),

Chapitre 2

Opérations rapides avec les formes de Chebyshev

Sommaire

2.1 Résultats utiles ou classiques **34**
2.1.1 Les calculs efficaces dans la base des monômes 35
2.1.2 Calculer T_n dans la base des monômes 36
2.2 La multiplication et la division de formes de Chebyshev **39**
2.2.1 Multiplier deux formes de Chebyshev 39
2.2.2 La division euclidienne de formes de Chebyshev 41
2.2.3 Le cas de la division exacte 43
2.3 Stratégie "Diviser pour régner" **44**
2.3.1 Produit de plusieurs de polynômes 46
2.3.2 Composition dans la base de monômes 47
2.4 Changement de base **49**
2.4.1 Calculer la forme de Chebyshev d'un polynôme 50
2.4.2 Développer une forme de Chebyshev 53

Résumé : Dans ce chapitre, nous construisons des algorithmes rapides dans $\mathbb{Z}[x]$ dans la base des polynômes unitaires de Chebyshev. Le terme *rapide* est compris comme *aussi rapide que dans la base des monômes* $\mathcal{X}$.

Nous donnons d'abord une méthode pour calculer T_n dans la base des monômes en $\widetilde{\mathcal{O}}(n)$ opérations arithmétiques, $\widetilde{\mathcal{O}}(n^2)$ opérations binaires.

Ensuite, nous considérons la multiplication et la division dans la base des polynômes unitaires de Chebyshev. L'application $\mathcal{D} : P \mapsto x^{\deg P} P(x + 1/x)$ y joue un rôle très important : en composant avec $\mathcal{D}$, la multiplication, la division, le calcul du pgcd, sont équivalents aux calculs dans la base des monômes.

Nous rappelons la stratégie *Diviser pour Régner* dans la section 2.3 pour détailler deux situations qui seront utilisées plusieurs fois : (1) La multiplication de k polynômes de degrés inférieurs à d, présentés dans la base des monômes par les coefficients de taille binaires au plus τ se fait en $\widetilde{\mathcal{O}}(k^2 n\tau)$ opérations binaires ; (2) La composition des polynômes f et g peut être exécutée en $\widetilde{\mathcal{O}}(dd'(\tau(f) + d\tau(g)))$ opérations binaires.

En appliquant la stratégie *Diviser pour Régner*, nous développons des algorithmes pour effectuer rapidement les changements de bases entre la base des monômes et la base des polynômes unitaires de Chebyshev en $\widetilde{\mathcal{O}}(d^2 + d\tau)$ opérations binaires, où d et τ sont le degré et la taille binaire des entrées.

2.1 Résultats utiles ou classiques

Pour commencer, nous rappelons quelques notions importantes de Calcul Formel et de complexité.

Définition 2.1 (Taille binaire d'un nombre rationnel)**.** Soit $n \in \mathbb{Z}^*$, on appelle la *taille binaire* de n, dénoté $\tau(n)$, le nombre de bits nécessaires pour écrire n, i.e. $\tau(n) = 1 + \lfloor \log_2 |n| \rfloor$; Par convention $\tau(0) = \tau(1) = 1$;

Soit $r = \frac{u}{v}$ un nombre rationnel, $u \in \mathbb{Z}, v \in \mathbb{Z}_{>0}$, $(u, v) = 1$. La taille binaire de r est définie par $\tau(r) = \max\{\tau(u), \tau(v)\}$.

Dans ce travail, quand on parle de *la taille binaire*, sans explication supplémentaire, il est sous-entendu que les données sont entières.

Notation 2.2. Soit K un anneau, $\mathcal{B} = (B_j, j \in \mathbb{N})$ une base de $K[x]$, c'est-à-dire que chaque polynôme $f \in K[x]$ s'écrit de façon unique sous la forme :

$$f = f_0 + \sum_{j=1}^{d} f_j B_j \quad (f_d \neq 0).$$

Par abus de notation, on notera $[\mathcal{B}]f$, la décomposition de f dans la base $\mathcal{B}$. Quand $\mathcal{B}$ est la base des polynômes unitaires de Chebyshev $\mathcal{T} = (1, T_k, k \in \mathbb{Z}_{>0})$, on appelle $[\mathcal{T}]f$ sa *forme de Chebyshev*.

Étant donné $[\mathcal{B}]f$, nous définissons :

- *Le coefficient dominant* de f est f_d, dénoté $\mathrm{lc}([\mathcal{B}]f)$ [1] ;
- *Le j-ième coefficient* de f est f_j, dénoté $[B_j]f$;
- Plusieurs normes peuvent être définies, lorsque $K = \mathbb{Z}, \mathbb{Q}, \mathbb{R}$.
  - † *La norme infinie* $\|.\|_\infty$est la valeur absolue la plus grande des coefficients,
  $$\|f\|_\infty = \max_j(|f_j|);$$
  - † *La norme Manhattan* $\|.\|_1$ est la somme de valeurs absolues des coefficients,
  $$\|f\|_1 = \sum_{j=0}^{d} |f_j|;$$
  - † *La "norme Chebyshev"* $\|.\|_T$ est déterminée sur l'anneau des formes de Chebyshev dans $\mathbb{R}[x]$: soit $f = f_0 + \sum_{j=1}^{d} f_j T_j$ alors on définit
  $$\|f\|_T = |f_0| + 2\sum_{j=1}^{d} |f_j|.$$
- *La taille binaire des coefficients* est le nombre minimal de bits permettant d'écrire un coefficient quelconque de f :
  $$\tau_{\mathcal{B}}(f) = \max\{\tau(f_j), j = 0, \ldots, d\};$$
  Si $K = \mathbb{Z}$ on considère :
  $$\tau_{\mathcal{B}}(f) = \lceil \log_2 \|[\mathcal{B}]f\|_\infty \rceil.$$

1. lc : Leading Coefficient en anglais

Notation 2.3 (Les notations $\mathcal{O}$ et $\widetilde{\mathcal{O}}$). [GG13, Sec. 25.7]

- Une fonction partielle[2] $f : \mathbb{N} \to \mathbb{R}$ est dite *asymptotiquement positive* lorsqu'il existe $N \in \mathbb{N}$ tel que $f(n) > 0$ pour tout $n > N$;

- Si f et g sont asymptotiquement positives, on dit que $f = \mathcal{O}(g)$ s'il existe une constante $c > 0$, un nombre entier positif N tel que $f(n) < cg(n)$ pour tout $n > N$;

Par exemple, $f = \mathcal{O}(1)$ implique que la fonction f est bornée.

- Si f et g sont asymptotiquement positives, alors on dit que $f = \widetilde{\mathcal{O}}(g)$ s'il existe deux constantes $c > 0$ et $N \in \mathbb{N}$ telles que $f(n) \leq g(n)(\log_2(3 + g(n)))^c$[3], pour tout $n > N$.

L'ajout du symbole $\widetilde{\ }$ indique que les facteurs logarithmiques sont négligés. Par exemple, $\mathcal{O}(n \log\log\log n)$ pourra être simplifié par $\widetilde{\mathcal{O}}(n)$.

Les propriétés utiles de base pour $\mathcal{O}$ et $\widetilde{\mathcal{O}}$ pour les calculs de complexité sont :

Lemme 2.4. *[GG13, page 720] Soit f_1, f_2, g_1, g_2 des fonctions asymptotiquement positives telles que $f_1 = \mathcal{O}(g_1)$ et $f_2 = \mathcal{O}(g_2)$. Alors $f_1 + f_2 = \mathcal{O}(g_1 + g_2)$, $f_1 \cdot f_2 = \mathcal{O}(g_1 \cdot g_2)$.*

De la même façon, si $f_1 = \widetilde{\mathcal{O}}(g_1)$ et $f_2 = \widetilde{\mathcal{O}}(g_2)$ alors $f_1 + f_2 = \widetilde{\mathcal{O}}(g_1 + g_2)$, $f_1 \cdot f_2 = \widetilde{\mathcal{O}}(g_1 \cdot g_2)$.

Pour évaluer la complexité d'un calcul, on peut considérer sa *complexité arithmétique*, c'est à dire l'ordre de grandeur du nombre d'opérations arithmétiques utilisées (dans un anneau de base devant être précisé) ou sa *complexité binaire*, c'est à dire l'ordre de grandeur du nombre d'opérations machine réalisées (opérations de base - comparaison, addition, soustraction, multiplication, division - entre entiers de taille fixe ou nombre flottants).

Dans ce travail nous utilisons régulièrement deux symboles C_A pour indiquer une complexité arithmétique et C_B pour une complexité binaire.

La complexité $\widetilde{\mathcal{O}}(d)$ s'appelle *quasi-linaire* ; $\widetilde{\mathcal{O}}(d^2)$ s'appelle *quasi-quadratique* .

2.1.1 Les calculs efficaces dans la base des monômes

Multiplication

Le calcul du produit de polynômes et d'entiers est au cœur de l'algorithmique efficace en Calcul Formel. Pour multiplier deux polynômes de degrés d à coefficients dans un anneau K, la méthode naïve consistant à calculer successivement $(\sum_{j=0}^{d} a_j x^j)(\sum_{j=0}^{d} b_j x^j) = \sum_{0 \leq i,j \leq d} a_i b_j x^{i+j}$, requiert $\mathcal{O}(d^2)$ opérations dans K. De même, l'algorithme naïf de multiplication de deux entiers à d chiffres demande $\mathcal{O}(d^2)$ opérations.

Ces algorithmes ont été améliorés. Il est apparu plusieurs algorithmes de multiplication rapide, dont celui de Karatsuba, de complexité $\mathcal{O}(d^{1,59})$, ainsi que ceux utilisant la transformation de Fourier rapide, et qui ont des complexités quasi-linaires en d [GG13, Chap. 8].

Définition 2.5. [GG13, Déf. 8.26] - Soit R un anneau commutatif avec unité. On appelle la fonction $\mathcal{M} : \mathbb{N}_{>0} \to \mathbb{R}_{>0}$ un temps de multiplication pour $R[x]$ si deux polynômes dans $R[x]$ de degrés inférieurs à n peuvent être multipliés en utilisant au plus $\mathcal{M}(n)$ opérations dans R.

- De la même façon, une fonction $\mathcal{M}$ comme ci-dessus s'appelle un temps de multiplication pour $\mathbb{Z}$ si deux nombres entiers de taille binaire au plus n peuvent être multipliés en utilisant $\mathcal{M}(n)$ opérations binaires.

2. Il n'est pas obligé que f est définie pour tout $n \in \mathbb{N}$.
3. Ici le terme 3 est ajouté pour que $\log_2[3 + g(n)]$ soit asymptotiquement pus grand que 1.

Proposition 2.6 (Multiplication rapide)**.** *[GG13, Théo. 8.23, Théo 8.24]*

1. *Soit R un anneau commutatif, alors deux polynômes de degrés inférieurs à n peuvent être multipliés en $\mathcal{M}(n) = \mathcal{O}(n \log_2 \log_2 n) = \widetilde{\mathcal{O}}(n)$ opérations arithmétiques dans R ;*
2. *Deux nombres entiers de tailles binaires au plus n peuvent être multipliés en*

$$\mathcal{O}(n \log_2 n \log_2 \log_2 n) = \widetilde{\mathcal{O}}(n)$$

opérations binaires.

Corollaire 2.7. *[GG13, Coro. 8.27] Deux polynômes dans $\mathbb{Z}[x]$ de degré au plus n, avec les coefficients de taille binaire inférieure à τ peuvent être multipliés en $\mathcal{M}(n, \tau) = \mathcal{O}(\mathcal{M}(n\tau + n \log_2 n)) = \widetilde{\mathcal{O}}(n\tau)$ opérations binaires.*

Division

Plusieurs opérations arithmétiques dans $\mathbb{Z}[x]$ dans la base des monômes utilisent la multiplication rapide, c'est le cas, par exemple, des différentes divisions (avec ou sans reste) :

Proposition 2.8 (Complexité de la division)**.** *[GG13, Théo. 9.6, Théo. 9.8]*

1. *Soit R un anneau commutatif avec l'unité, $f, g \in R[x]$, $\deg f = n + m$, $\deg g = n$, g est unitaire. Alors la division avec reste de f par g peut être faite en utilisant $\widetilde{\mathcal{O}}(n + m)$ opérations dans R ;*
2. *Soit a, b deux entiers de taille binaire inférieur à n. Alors la division avec reste de a par b peut être faite en $\widetilde{\mathcal{O}}(n)$ opérations binaires.*

Proposition 2.9 (Sur la division exacte)**.** *[GG13, Théo. 9.6 et commentaires - page 261] Soit f et g dans $\mathbb{Z}[x]$ de degrés au plus d avec les coefficients de taille binaire inférieure à τ alors :*

1. *Il est possible de tester si f est divisible par g en $\widetilde{\mathcal{O}}(d^2 + d\tau)$ opérations binaires.*
2. *En cas où $g \mid f$, on peut obtenir le quotient $\frac{f}{g}$ en $\widetilde{\mathcal{O}}(d^2 + d\tau)$ opérations binaires.*

2.1.2 Calculer T_n dans la base des monômes

Le calcul T_n dans la base des monômes est un cas particulier du changement de base que nous verrons plus tard.

Le polynôme de Chebyshev de première espèce vérifie de nombreuses relations, dont beaucoup sont rappelées dans [Czi12]. A partir de la récurrence $\mathbf{T}_{n+1} = 2x\mathbf{T}_n - \mathbf{T}_{n-1}$, on obtient par exemple :

- $\mathbf{T}_n = \frac{1}{2}[(x + \sqrt{x^2 - 1})^n + (x - \sqrt{x^2 - 1})^n]$,
- $\mathbf{T}_n$ est le déterminant de la matrice tri-diagonale $\begin{pmatrix} x & -1 & 0 & 0 & \dots & 0 \\ -1 & 2x & -1 & 0 & \dots & 0 \\ 0 & -1 & 2x & -1 & \dots & 0 \\ \vdots & \vdots & \ddots & \ddots & \ddots & \vdots \\ 0 & 0 & \dots & -1 & 2x & -1 \\ 0 & 0 & \dots & 0 & -1 & 2x \end{pmatrix}$.
- La famille $(\mathbf{T}_n)$ est orthonormée pour la mesure $\sqrt{1 - x^2}\mathrm{d}x$ sur $[-1, 1]$. On obtient $\mathbf{T}_n = \frac{(-2)^n n!}{(2n)!}\sqrt{1 - x^2}\frac{d^n}{dx^n}(1 - x^2)^{n - \frac{1}{2}}$

- Ils peuvent aussi être générés par l'expansion de Taylor de la fonction

$$F(z) = \frac{1}{2}\Big(1 + \frac{1-z^2}{1-2xz+z^2}\Big);$$

- On peut les trouver comme solution de l'équation différentielle :

$$(1-x^2)\mathbf{T}_n''(x) - x\mathbf{T}_n'(x) + n^2\mathbf{T}_n(x) = 0,$$

à la condition initiale $\mathbf{T}_n'(0) = \begin{cases} 0 & \text{si } 2 \mid n \\ (-1)^{\frac{n-1}{2}} & \text{sinon} \end{cases}$, $\mathbf{T}_n(0) = \begin{cases} 0 & \text{si } (2,n)=1 \\ (-1)^{\frac{n}{2}} & \text{sinon} \end{cases}$;

- Enfin, il y a également la récurrence matricielle : $\begin{pmatrix} \mathbf{T}_n(x) \\ \mathbf{T}_{n-1}(x) \end{pmatrix} = \begin{pmatrix} 2x & -1 \\ 1 & 0 \end{pmatrix}^{n-1} \begin{pmatrix} x \\ 1 \end{pmatrix}$.

Ces techniques sont toujours applicables au polynôme unitaire de Chebyshev T_n. Nous proposons une méthode différente permettant de calculer T_n avec une complexité arithmétique quasi-linaire, en temps de calcul quasi-quadratique.

Un algorithme quasi-optimal pour calculer $[\mathcal{X}]T_n$

Posons $\ell = \lfloor \log_2 n \rfloor$. On suppose que tous les $T_{2^j}, j = 1, \dots \ell+1$ sont pré-calculés. L'algorithme 2.10 suivant utilise la relation

$$[\mathcal{X}]T_{(h+k)/2} = [\mathcal{X}]\Big(\frac{T_h + T_k}{T_{(k-h)/2}}\Big),$$

il ne demande que $\ell+1$ multiplications et au plus $\ell+1$ divisions exactes de polynômes dans $\mathbb{Z}[x]$ exprimés dans la base des monômes :

Algorithme 2.10 : Calculer T_n

Entrées : n
Sorties : $[\mathcal{X}]T_n$

```
begin
    ℓ ← ⌊log₂ n⌋ ; T₁ ← x;
    pour i ← 0 a ℓ faire
        T_{2^{i+1}}(x) = T²_{2^i}(x) − 2
    h ← 2^ℓ ; k ← 2^{ℓ+1};
    répéter
        j ← (h + k)/2;
        Calculer T_j ← [X] (T_h + T_k)/T_{(k−h)/2};
        si n > j alors
            h ← j
        sinon
            k ← j
    jusqu'à ce que (n − h)(n − k) = 0;
retourner T_n
```

Preuve de correction. Posons $\ell = \lfloor \log_2 n \rfloor$, $h_0 = 2^\ell$, $k_0 = 2 \cdot 2^\ell$ et l'on dénote l'intervalle $i-$ième apparaissant dans le processus par $[h_i, k_i]$.

Par la détermination décrite et l'induction, nous avons

$$h_i = \left\lfloor \frac{n}{2^{\ell-i}} \right\rfloor \cdot 2^{\ell-i} \quad ; \quad k_i = \left\lceil \frac{n}{2^{\ell-i}} \right\rceil \cdot 2^{\ell-i}.$$

Donc, si n est factorisé par $n = 2^\alpha \cdot (2n'+1)$, l'intervalle $(\ell-\alpha-1)-$ième sera déterminé par deux points :

$$h_{\ell-\alpha-1} = \left\lfloor \frac{2^\alpha(2n'+1)}{2^{\alpha+1}} \right\rfloor \cdot 2^{\alpha+1} = n - 2^\alpha \quad ; \quad h_{\ell-\alpha-1} = \left\lceil \frac{2^\alpha(2n'+1)}{2^{\alpha+1}} \right\rceil \cdot 2^{\alpha+1} = n + 2^\alpha,$$

en ce point là, la moyenne arithmétique de ces deux valeurs est bien n ce qui termine l'algorithme. □

Exemple 2.11. $n = 37$, $\ell = \lfloor \log_2 n \rfloor = 5$.

- D'abord on calcule $T_1, T_2, T_4, T_8, T_{16}, T_{32}, T_{64}$;

- Partant de l'intervalle $[32, 64]$, on calcule $T_{48} = \dfrac{T_{32}+T_{64}}{T_{16}}$, $T_{40} = \dfrac{T_{32}+T_{48}}{T_8}$, $T_{36} = \dfrac{T_{32}+T_{40}}{T_4}$, $T_{38} = \dfrac{T_{36}+T_{40}}{T_2}$, $T_{37} = \dfrac{T_{36}+T_{38}}{T_1}$ successivement.

Au total nous avons dû effectuer $\ell+1 = 6$ multiplications de polynômes $(T_1^2, \ldots, T_{32}^2)$ et $\ell = 5$ divisions exactes de polynômes

Lemme 2.12. *Soit $n > 0$ un entier naturel, alors $\tau(T_n) \leq n-1$. En posant $\ell = \lfloor \log_2 n \rfloor$, on peut calculer tous les $T_{2^k}, k = 0, \ldots, \ell+1$ en $\widetilde{\mathcal{O}}(n)$ opérations arithmétiques dans $\mathbb{Z}$ et en $\widetilde{\mathcal{O}}(n^2)$ opérations binaires.*

Démonstration.
1. Pour la taille binaire des coefficients de T_n, signalons d'abord que $T_3(x) = x^2 - 3x$, $\|T_3\|_\infty = 3 < 2^2$; Supposons que $\|T_k\|_\infty < 2^{k-1}$ pour tout k, $3 \leq k \leq n$. Comme $T_{n+1} = x.T_n - T_{n-1}$, alors

$$\|T_{n+1}\|_\infty \leq \|T_n\|_\infty + \|T_{n-1}\|_\infty < 2^{n-1} + 2^{n-2} < 2^n$$

et on en déduit que $\|T_{n+1}\|_\infty < 2^n$. Par induction, $\|T_n\|_\infty \leq 2^{n-1}$ pour tout $n \in \mathbb{N}_{\geq 3}$.
2. D'après la formule (1.7b), on a $T_{2k}(x) = T_k(x)^2 - 2$ où $\deg T_k = k$, $\tau(T_k) \leq k$. Par le corollaire 2.7, on calcule $T_{2^{k+1}}$ à partir du T_{2^k} en $\widetilde{\mathcal{O}}(2^k)$ opérations arithmétiques, $\widetilde{\mathcal{O}}(4^k)$ opérations binaires.

Au total, pour obtenir tous les $T_{2^k}, k = 0, \ldots, \ell+1$, nous utilisons :

$$\sum_{k=0}^{\ell} \widetilde{\mathcal{O}}(2^k) = \widetilde{\mathcal{O}}(2^{\ell+1}) = \widetilde{\mathcal{O}}(n)$$

opérations arithmétiques, et

$$\sum_{k=0}^{\ell} \widetilde{\mathcal{O}}(4^k) = \widetilde{\mathcal{O}}(4^{\ell+1}) = \widetilde{\mathcal{O}}(n^2)$$

opérations binaires. □

Proposition 2.13. *Soit $n > 2$ un nombre entier. On peut calculer T_n dans la base des monômes en utilisant $\widetilde{\mathcal{O}}(n)$ opérations arithmétiques, $\widetilde{\mathcal{O}}(n^2)$ opérations binaires.*

Démonstration. 1. Pour ce calcul de complexité, nous utilisons l'algorithme 2.10.

1. On calcule d'abord tous les T_{2^j}, $j = 0, \ldots, \ell+1$ en $C_A^{(1)} = \widetilde{\mathcal{O}}(n)$ opérations arithmétiques ou $C_B^{(1)} = \widetilde{\mathcal{O}}(n^2)$ opérations binaires d'après le lemme 2.12.

2. Quand les $T_{2^j}, j = 0, \ldots, \ell+1$ sont calculés, nous effectuons au plus $\ell+1$ divisions exactes $\frac{T_h+T_k}{T_{(k-h)/2}}$, où $h, k < 2^{\ell+1} \leq 2n$.

On remarque que $\deg(T_h + T_k) \leq \max(h, k) < 2n$; $\|T_h + T_k\|_\infty \leq \|T_h\|_\infty + \|T_k\|_\infty < 2^{\ell+1} - 1 < 2^{2n}$ donc $\tau(T_h + T_k) < 2n$. $\deg T_{(k-h)/2} = (k-h)/2 < n$, $\tau(T_{(k-h)/2}) \leq (k-h)/2 < n$. D'après les propositions 2.8 et 2.9, les quotients $\frac{T_h+T_k}{T_{(k-h)/2}}$ peuvent donc être calculés en $C_A^{(2)} = (\ell+1)\widetilde{\mathcal{O}}(n) = \widetilde{\mathcal{O}}(n)$ opérations arithmétiques ou $C_B^{(2)} = (\ell+1)\widetilde{\mathcal{O}}(n^2) = \widetilde{\mathcal{O}}(n^2)$ opérations binaires, car $\ell + 1 = \lceil \log_2 n \rceil$.

En résumé, la complexité du calcul de T_n est : $C_A^{(1)} + C_A^{(2)} = \widetilde{\mathcal{O}}(n)$ opérations arithmétiques et $C_B^{(1)} + C_B^{(2)} = \widetilde{\mathcal{O}}(n^2)$ opérations binaires. □

2.2 La multiplication et la division de formes de Chebyshev

L'ensemble de formes de Chebyshev dans $\mathbb{Z}[x]$ forme un anneau euclidien. Nous proposons dans la suite des méthodes pour effectuer rapidement les opérations arithmétiques dans cet anneau.

2.2.1 Multiplier deux formes de Chebyshev

Étant données deux formes de Chebyshev f et g dans $\mathbb{Z}[x]$ où $\deg f = \deg g \leq d$ dans la base $\mathcal{T}$:

$$[\mathcal{T}]f = f_0 + \sum_{j=1}^{d} f_j T_j, \quad [\mathcal{T}]g = g_0 + \sum_{j=1}^{d} g_j T_j,$$

supposons que $\tau = \max\{\tau_{\mathcal{T}}(f), \tau_{\mathcal{T}}(g)\}$. Nous calculons $[\mathcal{T}](f \cdot g)$.

Idée de l'algorithme

Dans son travail [Gio12] en 2012, Giorgi propose un algorithme permettant de multiplier deux formes de Chebyshev à l'aide de la convolution de Baszenski [Bas97]. Cette méthode a été décrite à nouveau par Benoît dans sa thèse [Ben12] soutenue la même année.

Il est possible d'adapter cette idée pour notre famille des polynômes unitaires de Chebyshev. Le résultat suivant est une modification de [Bas97, Prop. 2.1] :

Lemme 2.14. *Soit $f = f_0 + \sum_{j=1}^{d} f_j T_j$, $g = g_0 + \sum_{j=1}^{d} g_j T_j$ deux polynômes dans $\mathbb{Z}[x]$. Alors en posant :*

$$(\sum_{j=0}^{d} f_j x^j) \cdot (\sum_{j=0}^{d} g_j x^j) = \sum_{i=0}^{2d} a_i x^i \quad ; \quad (\sum_{j=0}^{d} f_j x^j) \cdot (\sum_{j=0}^{d} g_{d-j} x^j) = \sum_{i=0}^{2d} b_i x^i,$$

on a $f \cdot g = h_0 + \sum_{j=1}^{2d} h_j T_j$ avec :

$$h_i = \begin{cases} 2b_d - f_0 g_0, & i = 0; \\ a_i + b_{d-i} + b_{d+i} - (f_0 g_i + f_i g_0), & i = 1, \ldots, d-1; \\ a_i, & i = d, \ldots, 2d. \end{cases} \tag{2.1}$$

Démonstration. Il suffit de regrouper les coefficients de chaque T_k, $k = 1, \dots, 2d$ dans l'expression :

$$f \cdot g = f_0\Big(g_0 + \sum_{j=1}^{d} g_j T_j\Big) + g_0\Big(f_0 + \sum_{j=1}^{d} f_j T_j\Big) + \sum_{1 \le i,j \le d} f_i g_j (T_{i+j} + T_{|i-j|}).$$

□

Ainsi, pour multiplier $[\mathcal{T}]f$ avec $[\mathcal{T}]g$, on peut calculer les deux produits $(\sum_{j=0}^{d} f_j x^j) \cdot (\sum_{j=0}^{d} g_j x^j)$ et $(\sum_{j=0}^{d} f_j x^j) \cdot (\sum_{j=0}^{d} g_{d-j} x^j)$ dans la base des monômes, puis utiliser l'identité (2.1) pour construire $[\mathcal{T}](f \cdot g)$.

D'une autre façon, le lemme 2.14 peut être expliqué d'une manière beaucoup plus simple. D'après l'identité (1.25) $\mathcal{D}(f \cdot g) = \mathcal{D}(f) \cdot \mathcal{D}(g)$ et par la remarque 1.21, les coefficients de $[\mathcal{T}](f \cdot g)$ sont exactement ceux de $[\mathcal{X}]\mathcal{D}(f) \cdot \mathcal{D}(g)$, ce qui induit l'algorithme 2.15 :

Algorithme 2.15 : Multiplication de formes de Chebyshev par le Doublage

Entrées : $[\mathcal{T}]f = f_0 + \sum_{j=1}^{d} f_j T_j$, $[\mathcal{T}]g = g_0 + \sum_{j=1}^{d'} g_j T_j$, $f, g \in \mathbb{Z}[x]$
Sorties : $[\mathcal{T}](f \cdot g)$

```
1 begin
2     Calculer H(x) = Σ_{j=0}^{d+d'} h_{d+d'-j} x^j = D(f).D(g) (mod x^{d+d'+1});
3     retourner h_0 + Σ_{j=1}^{d+d'} h_j T_j
```

Preuve de correction. La correction de l'algorithme est évidemment corollaire de l'identité (1.25). □

Complexité de la multiplication

Rappelons d'abord des évaluations des coefficients après les opérations arithmétiques dans la base des monômes de l'anneau polynomial.

Remarque 2.16.

1. Pour tout $f, g \in \mathbb{R}[x]$, pour tout k on a : $\Big|[x^k](f+g)\Big| = \Big|[x^k]f + [x^k]g\Big| \le \|f\|_\infty + \|g\|_\infty$ et $\Big|[x^k](f \cdot g)\Big| = \Big|\sum_{\substack{0 \le i \le \deg f \\ 0 \le j \le \deg g \\ i+j=k}} f_i g_j\Big| \le (1 + \min\{\deg f, \deg g\})\|f\|_\infty \|g\|_\infty$, alors il est clair que :
$$\|f + g\|_\infty \le \|f\|_\infty + \|g\|_\infty; \tag{2.2a}$$
$$\|f \cdot g\|_\infty \le (1 + \min(\deg f, \deg g))\|f\|_\infty \|g\|_\infty. \tag{2.2b}$$
2. En remplaçant $\mathbb{R}$ par $\mathbb{Z}$ où $\tau(f) = \lceil \log_2(\|f\|_\infty) \rceil$, pour tout $f, g \in \mathbb{Z}[x]$ on a :
$$\tau(f + g) \le \max\{\tau(f), \tau(g)\} + 1; \tag{2.3a}$$
$$\tau(f \cdot g) \le \tau(f) + \tau(g) + \log_2(1 + \min\{\deg f, \deg g\}). \tag{2.3b}$$

Proposition 2.17 (Multiplication de formes de Chebyshev)**.** *Soit f et g deux polynômes dans $\mathbb{Z}[x]$ de degré au plus d, tels que $\tau_{\mathcal{T}}(f), \tau_{\mathcal{T}}(g) \le \tau$. Alors on a :*

$$\tau_{\mathcal{T}}(f \cdot g) \le \tau_{\mathcal{T}}(f) + \tau_{\mathcal{T}}(g) + \log_2 \min(\deg f, \deg g) + 1; \tag{2.4}$$

La forme de Chebyshev $[\mathcal{T}](f \cdot g)$ peut être calculée en $\widetilde{\mathcal{O}}(d)$ opérations arithmétiques, $\widetilde{\mathcal{O}}(d\tau)$ opérations binaires.

Démonstration. 1. Appliquons l'algorithme 2.15. Vu le fait que $\tau_{\mathcal{T}}(f \cdot g) = \tau(\mathcal{D}(f \cdot g)) = \tau(\mathcal{D}(f) \cdot \mathcal{D}(g))$ (remarque 1.21, identité (1.25)), alors l'inégalité (2.3b) nous dit que :

$$\begin{aligned} \tau_{\mathcal{T}}(f \cdot g) &\leq \tau(\mathcal{D}(f)) + \tau(\mathcal{D}(g)) + \log_2(1 + \min\{\deg \mathcal{D}(f), \deg \mathcal{D}(g)\}) \\ &\leq \tau_{\mathcal{T}}(f) + \tau_{\mathcal{T}}(g) + \log_2 \min\{\deg f, \deg g\} + 1; \end{aligned} \tag{2.5}$$

Maintenant étudions la complexité. Le calcul comporte deux phases :

(1) Calcul du produit $\mathcal{D}(f) \cdot \mathcal{D}(g)$ dans la base des monômes. On a $\deg \mathcal{D}(f), \deg \mathcal{D}(g) \leq 2d$ et $\tau(\mathcal{D}(f)), \tau(\mathcal{D}(g)) \leq \tau$. On utilise $C_A^{(1)} = \widetilde{\mathcal{O}}(d)$ opérations arithmétiques (d'après la proposition 2.6) et $C_B^{(1)} = \widetilde{\mathcal{O}}(d\tau)$ opérations binaires, d'après le corollaire 2.7.

(2) Collecter les coefficients de $[\mathcal{T}](f \cdot g)$ dans $\mathcal{D}(f) \cdot \mathcal{D}(g)$. Cela utilise clairement au plus $C_A^{(2)} = d$ opérations arithmétiques et au plus $C_B^{(2)} = d\tau$ opérations binaires.

La complexité de l'algorithme 2.15 est donc finalement donnée par $C_A^{(1)} + C_A^{(2)} = \widetilde{\mathcal{O}}(d)$ opérations arithmétiques, et $C_B^{(1)} + C_B^{(2)} = \widetilde{\mathcal{O}}(d\tau)$ opérations binaires. □

2.2.2 La division euclidienne de formes de Chebyshev

Étant donnés deux polynômes f et g de $\mathbb{Z}[x]$ tels que g soit unitaire[4], $\deg f = d$, $\deg g = d'$. Alors il existe un unique couple (q, r) dans $\mathbb{Z}[x] \times \mathbb{Z}[x]$, avec $\deg r < d'$ et qui vérifient $f = g \cdot q + r$. On appelle q le quotient et r le reste de la division euclidienne de f par g. On les notera $q = \mathrm{Quo}(f, g)$ et $r = \mathrm{Rem}(f, g)$.

Si f et g sont donnés par leurs formes de Chebyshev :

$$[\mathcal{T}]f = f_0 + \sum_{j=1}^{d} f_j T_j, \quad [\mathcal{T}]g = g_0 + \sum_{j=1}^{d'} g_j T_j,$$

$\tau = \max\{\tau_{\mathcal{T}}(f), \tau_{\mathcal{T}}(g)\}$, alors on calcule les deux formes de Chebyshev $[\mathcal{T}]q$, $[\mathcal{T}]r$.

Idée et Algorithme

Il suffit de considérer le cas où $d' \leq d$. L'idée principale de la division est d'utiliser l'application $\mathcal{D}$ pour revenir à une situation dans la base des monômes que l'on sait résoudre.

D'après la formule (1.29), signalons que si $f = g \cdot q + r$ alors

$$\mathcal{D}(f) \equiv \mathcal{D}(g) \cdot \mathcal{D}(q) \pmod{x^{d-d'+1}}.$$

Puisque g est unitaire, alors $D(g)$ est également unitaire (remarque 1.21) et réversible. C'est donc un élément inversible dans $\mathbb{Z}[x]/\langle x^{d-d'+1} \rangle$. On a donc :

$$\mathcal{D}(q) = \mathcal{D}(f) \cdot \mathcal{D}(g)^{-1} \pmod{x^{d-d'+1}}. \tag{2.6}$$

La relation (2.6) fournit les $d - d' + 1$ premiers coefficients de $\mathcal{D}(q)$. Mais on a la remarque 1.21 confirmant que :

$$P(x) = p_0 + \sum_{j=1}^{n} p_j T_j \;\Rightarrow\; \mathcal{D}(P) = p_0 x^n + \sum_{j=1}^{n} p_j (x^{n+j} + x^{n-j}),$$

4. On n'a plus besoin de cette condition si $K \in \{\mathbb{Q}, \mathbb{R}\}$. Pourtant, afin de mieux expliquer avec la notion de la taille binaire des coefficients, on restreint $K = \mathbb{Z}$ où la condition est nécessaire.

de plus $\deg q = d - d'$, alors ces coefficients sont suffisants pour reconstruire $[\mathcal{T}](q)$.

Le problème du calcul de l'inversion $\mathcal{D}(g)^{-1} \pmod{x^{d-d'+1}}$ a été déjà résolu dans la base des monômes $\mathcal{X}$. Par exemple avec la méthode de Newton expliquée dans [GG13, Chapter 9], on calcule successivement :

$$w_0 = 1, w_{k+1} = w_k(2 - w_k \cdot g) \pmod{x^{2^{k+1}}}$$

jusqu'à ce que 2^{k+1} dépasse $d - d' + 1$. Signalons :

Lemme 2.18. *[GG13, Théo. 9.4, Théo. 9.6 et ses commentaires] Soit g un polynôme dans $K[x]$, $\ell \in \mathbb{N}_{>0}$. Alors le polynôme $w = g^{-1} \pmod{x^\ell}$ étant dans $K[x]_{\deg<\ell}$ peut être calculé en $\widetilde{\mathcal{O}}(\ell)$ opérations arithmétiques dans K* [5].

Quand $K = \mathbb{Z}$, $g(0) = 1$, $\tau(g) = \tau$ alors $\tau(w) = \mathcal{O}(\ell\tau)$. Il est possible de calculer w en $\widetilde{\mathcal{O}}(\ell^2\tau)$ opérations binaires.

Nous proposons l'algorithme 2.19 permettant d'exécuter la division euclidienne dans la base des polynômes unitaires de Chebyshev :

Algorithme 2.19 : Division euclidienne de formes de Chebyshev

Entrées : $[\mathcal{T}]f$, $[\mathcal{T}]g$ des $f, g \in \mathbb{Z}[x]$, $0 \leq \deg g = d'$, $\deg f = d$, g est unitaire
Sorties : $([\mathcal{T}]q, [\mathcal{T}]r)$ où $q = \mathrm{Quo}(f, r)$, $r = \mathrm{Rem}(f, g)$

```
1 begin
2   si d < d' alors
3     retourner (q = 0, r = [T]f)
4   Construire D(f), D(g)/* vu la remarque 1.21 */
5   Calculer w ← D(g)^{-1} (mod x^{d-d'+1});
6   Calculer q̃ ← D(f) · w (mod x^{d-d'+1}) ; /* Dénotons q̃ = Σ_{i=0}^{d-d'} λ_i x^i */
7   q ← λ_{d-d'} + Σ_{i=1}^{d-d'} λ_{d-d'-i} T_i;
8   Calculer [T]r ← [T](f − g · q);
9   retourner ([T]q, [T]r)
```

Preuve de correction. La correction de l'algorithme est confirmé par l'identité (2.6), la remarque 1.21 et le fait que $\deg q = d - d'$. □

Exemple 2.20. $f = -10 - 75T_1 - 17T_2 + 71T_3 - 44T_4 + 80T_5 - 82T_6 + 62T_7$,
$g = 72 + 74T_1 + 6T_2 - 92T_3 + 75T_4 + T_5$;

1. $\deg f = 7 > \deg g = 5$;
2. On construit
   (a) $\mathcal{D}(f) = 62 - 82x + 80x^2 - 44x^3 + 71x^4 - 17x^5 - 75x^6 - 10x^7 - 75x^8 - 17x^9 + 71x^{10} - 44x^{11} + 80x^{12} - 82x^{13} + 62x^{14}$;
   (b) $\mathcal{D}(g) = 1 + 75x - 92x^2 + 6x^3 + 74x^4 + 72x^5 + 74x^6 + 6x^7 - 92x^8 + 75x^9 + x^{10}$;
3. Faire l'inversion de Newton de $\mathcal{D}(g)$, $\ell = d - d' + 1 = 3$:
   (a) $w_0 \leftarrow 1$;
   (b) $w_1 \leftarrow 1(2 - \mathcal{D}(g) \cdot 1) \pmod{x^{2^{0+1}}} = 1 - 75x$;
   (c) $w_2 \leftarrow (1 - 75x(2 - \mathcal{D}(g) \cdot (1 - 75x))) \pmod{x^{2^2}} = 1 - 75x + 5717x^2 - 435681x^3$;

5. Si $\ell = 2^k, k \in \mathbb{N}$ on a $C_A = 3\mathcal{M}(\ell) + \ell$ [GG13, Theo. 9.4] ; sinon $C_A = 5\mathcal{M}(\ell) + \ell$ [CC12].

   (d) Alors $w = \mathrm{Rem}(w_2, x^\ell) = 1 - 75x + 5717x^2$;
4. Le quotient :
   (a) $\tilde{q} \leftarrow \mathcal{D}(f) \cdot w \pmod{x^3} = 360684x^2 - 4732x + 62$;
   (b) on en déduit $[\mathcal{T}]q = 36068 - 4732T_1 + 62T_2$;
5. Le reste : $[\mathcal{T}]r = [\mathcal{T}](f - q \cdot g) = -25269666 - 26320479T_1 - 2258411T_2 + 33561641T_3 - 27482328T_4$;

Proposition 2.21. *Soit f et g deux polynômes dans $\mathbb{Z}[x]$ donnés par leurs formes de Chebyshev $[\mathcal{T}]f$ et $[\mathcal{T}]g$ tels que $\deg f = d$, $\deg g = d' \leq d$, $\max(\tau_{\mathcal{T}}(f), \tau_{\mathcal{T}}(g)) = \tau$, g est unitaire. Soit q le quotient et r, le reste de la division euclidienne de f par g. Alors on a*

1. *$\tau_{\mathcal{T}}(q)$, $\tau_{\mathcal{T}}(r)$ sont les deux dans la classe $\mathcal{O}(d\tau)$;*
2. *Il est possible de calculer $([\mathcal{T}]q, [\mathcal{T}]r)$ en $\widetilde{\mathcal{O}}(d)$ opérations arithmétiques, $\widetilde{\mathcal{O}}(d^2\tau)$ opérations binaires.*

Démonstration. Suivons les différentes étapes de l'algorithme. Commençons par calculer $\mathcal{D}(f)$ et $\mathcal{D}(g)$ en $\widetilde{\mathcal{O}}(d\tau)$ opérations binaires. Nous avons $\tau_{\mathcal{X}}(\mathcal{D}(f)) = \tau_{\mathcal{T}}(f) \leq \tau$ et $\tau_{\mathcal{X}}(\mathcal{D}(g)) = \tau_{\mathcal{T}}(g) \leq \tau$.

Nous calculons ensuite $w = \mathcal{D}(g)^{-1} \pmod{x^{d-d'+1}}$ en $\widetilde{\mathcal{O}}(d)$ opérations arithmétiques dans $\mathbb{Z}[x]$ et $\widetilde{\mathcal{O}}(d^2\tau)$ opérations binaires d'après le lemme 2.18. Nous savons que $\tau_{\mathcal{X}}(w) = \widetilde{\mathcal{O}}(d\tau)$.

La multiplication de $\mathcal{D}(f)$ et de w dans la base des monômes requiert $\widetilde{\mathcal{O}}(d)$ opérations arithmétiques et $\widetilde{\mathcal{O}}(d^2\tau)$ opérations binaires d'après le corollaire 2.7. La taille binaire de $\tilde{q} = \mathcal{D}(f) \cdot w \pmod{x^{d-d'+1}}$ est en $\widetilde{\mathcal{O}}(d\tau)$ d'après l'inégalité (2.3b) [6] et le fait qu'il suffit de prendre les restes modulo $x^{d-d'+1}$.

Ensuite nous déduisons q à partir de $\mathcal{D}(q) = \tilde{q}$ en $\mathcal{O}(d)$ opérations arithmétiques (recopiage des coefficients de taille binaire $\widetilde{\mathcal{O}}(d\tau)$) et donc en $\widetilde{\mathcal{O}}(d^2\tau)$ opérations binaires.

Pour terminer nous obtenons $[\mathcal{T}]r = [\mathcal{T}](f - g \cdot q)$ en $\widetilde{\mathcal{O}}(d)$ opérations arithmétiques et $\widetilde{\mathcal{O}}(d^2\tau)$ opérations binaires, en utilisant l'estimation (2.4) [7] on obtient $\tau_{\mathcal{T}}(r) = \mathcal{O}(d\tau)$; □

2.2.3 Le cas de la division exacte

L'algorithme 2.19 de division euclidienne montre qu'on peut faire la division avec le reste par une méthode similaire à celle utilisée dans la base des monômes.

Il est souvent des cas (par exemple, ici pour le calcul des polynômes T_n par le premier algorithme 2.10, ou plus tard, pour le calcul du polynôme minimal M_n par l'algorithme 3.14) où nous savons que le reste de la division est nul et nous n'avons besoin que du quotient. Nous utilisons la remarque 1.21 permettant de passer de f à $\mathcal{D}(f)$:

$$\mathcal{D}(\sum_{i=0}^{n} p_i T_i) = x^n + \sum_{i=1}^{n} p_i (x^{n+i} + x^{n-i}).$$

Le corollaire 1.24 nous indique que $\mathcal{D}((f,g)) = (\mathcal{D}(f), \mathcal{D}(g))$ et $\mathcal{D}(f) \cdot \mathcal{D}(u) + \mathcal{D}(g) \cdot \mathcal{D}(v) = \mathcal{D}((f,g))$ où u et v sont des coefficients de Bézout : $u \cdot f + v \cdot g = (f,g)$. En utilisant la proposition

Proposition 2.22. *[GG13, Corollaire 11.14] Soit f et g dans $\mathbb{Z}[x]$ donnés dans la base des monômes, $\deg g, \deg g \leq d$, $\tau(f), \tau(g) \leq \tau$ alors on peut calculer le $\mathrm{pgcd}(f,g)$et les coefficients de Bézout u, v satisfont $u \cdot g + v \cdot f = \gcd(f,g)$ en $\widetilde{\mathcal{O}}(d^2\tau)$ opérations binaires.*

6. $\tau(f \cdot g) \leq \tau(f) + \tau(g) + \log_2(1 + \min\{\deg f, \deg g\}$
7. $\tau_{\mathcal{T}}(f \cdot g) \leq \tau_{\mathcal{T}}(f) + \tau_{\mathcal{T}}(g) + \log_2 \min(\deg f, \deg g) + 1$

Nous en déduisons

Corollaire 2.23. *Soit f et g dans $\mathbb{Z}[x]$ donnés dans la base de Chebyshev* $\deg g, \deg g \leq d$, $\tau_{\mathcal{T}}(f), \tau_{\mathcal{T}}(g) \leq \tau$ *alors on peut calculer les formes de Chebyshev du* $\operatorname{pgcd}(f,g)$ *et des coefficients de Bézout u, v qui satisfont $u \cdot g + v \cdot f = \gcd(f,g)$ en $\widetilde{\mathcal{O}}(d^2\tau)$ opérations binaires.*

En ce qui concerne la taille binaire des diviseurs, nous utilisons

Lemme 2.24 (Borne de Mignotte)**.** *[GG13, Theo. 6.33] Si $f, h \in \mathbb{Z}[x]$, tels que $h \mid f$ alors $\|h\|_\infty \leq 2^{\deg h}\|f\|_\infty\sqrt{\deg f + 1}$, ou bien*

$$\tau(h) \leq \deg h + \tfrac{1}{2}\log_2(\deg f + 1) + \tau(f). \tag{2.7}$$

qui entraîne

Corollaire 2.25. *Soit f un polynôme dans $\mathbb{Z}[x]$ de degré d, donné par sa forme de Chebyshev $[\mathcal{T}]f$ alors, pour tout diviseur h de f on a :*

$$\tau_{\mathcal{T}}(h) \leq \tau_{\mathcal{T}}(f) + 2\deg h + \tfrac{1}{2}\log_2(2\deg f + 1).$$

De l'identité (1.25) : $\mathcal{D}(f) \cdot \mathcal{D}(g) = \mathcal{D}(f \cdot g)$ et de la proposition 2.9, on déduit

Corollaire 2.26. *Soit f et g deux polynômes dans $\mathbb{Z}[x]$ donnés par leurs formes de Chebyshev $[\mathcal{T}]f$, $[\mathcal{T}]g$ tels que $\deg g \leq \deg f = d$, $\max(\tau_{\mathcal{T}}(f), \tau_{\mathcal{T}}(g)) = \tau$ alors :*

1. *On peut décider si $g \mid f$ en utilisant $\widetilde{\mathcal{O}}(d)$ opérations arithmétiques ou $\widetilde{\mathcal{O}}(d^2 + d\tau)$ opérations binaires.*
2. *Si $g \mid f$ alors le quotient $[\mathcal{T}](\frac{f}{g})$ peut être calculé en $\widetilde{\mathcal{O}}(d)$ opérations arithmétiques ou $\widetilde{\mathcal{O}}(d^2 + d\tau)$ opérations binaires ;*

Démonstration. Vu le fait que $\tau_{\mathcal{T}}(P) = \tau(\mathcal{D}(P))$ pour tout $P \in \mathbb{Z}[x]$, nos conclusions sont corollaires directs de la proposition 2.9 et l'identité (1.25). □

2.3 Stratégie "Diviser pour régner"

Un principe important dans la conception d'algorithmes efficaces est le paradigme "Diviser pour Régner" . Il consiste à résoudre un problème de taille n en le réduisant à un certain nombre m de sous problèmes de tailles $\frac{n}{m}$ (le plus souvent $m = 2$) puis à combiner les résultats efficacement. Nous allons utiliser cette stratégie pour d'une part calculer le produit de plusieurs polynômes et d'autre part pour évaluer un polynôme.

Produit de plusieurs polynômes

Nous cherchons à calculer $f = \prod_{j=0}^{k-1} f_j$ et nous supposons toujours que $k = 2^\ell$ (sinon on considère $\ell = \lceil \log_2(k+1) \rceil$ puis on pose $f_j = 1$ pour tout $j = k, \ldots, 2^\ell - 1$). La stratégie "Diviser pour Régner" nous amène à calculer les nœuds sur l'arbre illustré par le schéma

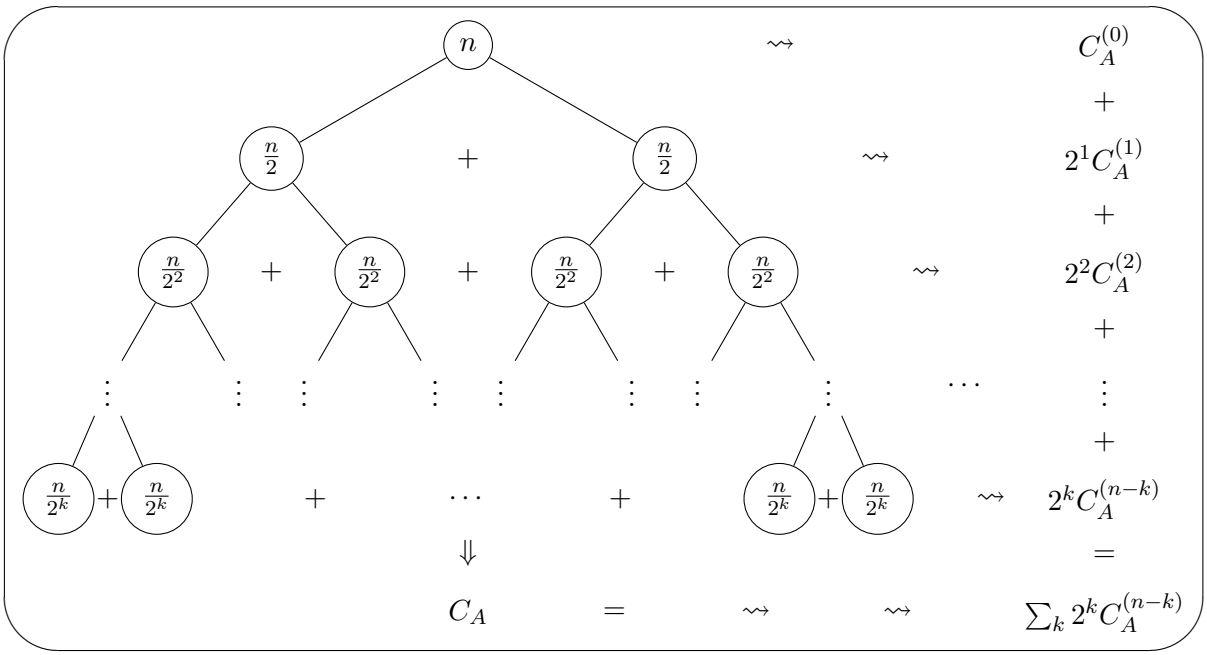
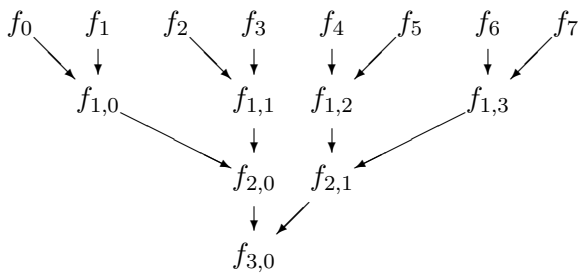


Figure 2.1 – Stratégie "diviser pour régner"

(2.8) où l'on "divise" par $m = 2$:



(2.8)

On calcule alors successivement :

$$\begin{aligned} f_{0,j} &= f_j, \quad && j = 0, \dots, 2^\ell - 1 \\ f_{c,j} &= f_{c-1,2j} \cdot f_{c-1,2j+1}, \quad && j = 0, \dots, 2^{\ell-c} - 1 \\ & && \text{pour } c = 1, \dots, \ell, \\ \text{enfin :} & && f = f_{\ell,0}. \end{aligned} \tag{2.9}$$

Remarquons que chaque $f_{c,j}$ est produit de 2^c polynômes :

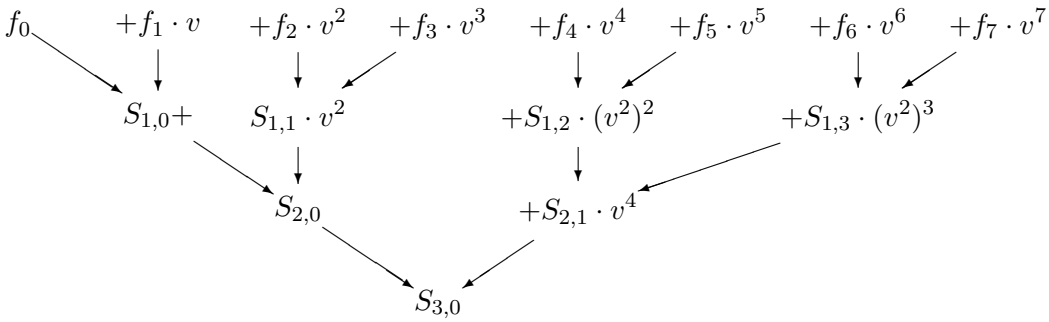
$$f_{c,j} = \prod_{i=0}^{2^c-1} f_{j \cdot 2^c + i}. \tag{2.10}$$

Evaluer un polynôme

Nous cherchons à évaluer

$$f = \sum_{j=0}^{k-1} f_j v^j.$$

Il est aussi possible de considérer $k = 2^\ell$ (sinon on pose $\ell = \lceil \log_2(k+1) \rceil$ puis on ajoute $f_j = 0$ pour tout $j = k, \ldots, 2^\ell - 1$). Avec la stratégie "Diviser par 2 pour Régner", on se ramène à calculer sur l'arbre illustré par le schéma (2.11) :



(2.11)

On calcule alors successivement :

$$\begin{aligned} S_{0,j} &= f_j, & j &= 0, \ldots, 2^\ell - 1 & v_0 &= v, \\ S_{c,j} &= S_{c-1,2j} + S_{c-1,2j+1} \cdot v_{c-1}, & j &= 0, \ldots, 2^{\ell-c} - 1 & v_c &= v_{c-1}^2 \\ & & & \text{pour } c = 1, \ldots, \ell, \\ & \text{enfin :} & f &= S_{\ell,0}. \end{aligned} \tag{2.12}$$

Remarquons que chaque $S_{c,j}$ se situe sur un segment de longueur 2^c de f et que chaque v_c est une puissance de v :

$$S_{c,j} = \sum_{i=0}^{2^c-1} f_{j.2^c+i} v^i; \tag{2.13a}$$

$$v_c = v^{2^c}; \tag{2.13b}$$

2.3.1 Produit de plusieurs de polynômes

Avant de donner un algorithme pour le calcul du produit de polynômes, nous évaluons la taille des coefficients du produit.

Lemme 2.27. *Soit $f_0, \ldots, f_{k-1}$ des polynômes dans $\mathbb{R}[x]$ tel que $\deg f_j \leq d$, $\|f_j\|_\infty \leq M$ pour tout $j = 0, \ldots, k-1$ alors on a :*

$$\| \prod_{j=0}^{k-1} f_j \|_\infty \leq (d+1)^{k-1} M^k. \tag{2.14}$$

Preuve du lemme 2.27. L'inégalité (2.14) est clairement vraie pour $k = 1$; Supposons qu'elle soit vraie jusqu'à k :

$$\|f_0 \ldots f_{k-1}\|_\infty \leq (d+1)^{k-1} M^k,$$

En appliquant l'inégalité (2.2b), on a :

$$\|(f_0 \ldots f_{k-1}) f_k\|_\infty \leq (d+1) \times (d+1)^{k-1} M^k \times M,$$

ce qui montre la propriété (2.14) pour $k+1$. □

En appliquant la stratégie "Diviser pour Régner" pour le produit de plusieurs polynômes dans la base des monômes, nous obtenons alors :

Proposition 2.28 (Coût de la multiplication de plusieurs polynômes)**.** *Soit $f_0, \ldots, f_{k-1}$ des polynômes de degré au plus d avec les coefficients entiers de taille binaire inférieure à τ, $f = \prod_{j=0}^{k-1} f_j$. Alors :*

$$\tau(f) < k[\tau + \log_2(d+1)]; \tag{2.15}$$

On peut calculer f en $\widetilde{\mathcal{O}}(kd)$ opérations arithmétiques, $\widetilde{\mathcal{O}}(k^2 d\tau)$ opérations binaires.

Démonstration. Dans notre cas où $K = \mathbb{Z}$, la condition $\tau(f_j) \leq \tau$ équivaut à $\|f_j\|_\infty \leq 2^\tau$ pour tout $j = 1, \ldots, k-1$. L'inégalité (2.14) nous donne :

$$\|f\|_\infty \leq (d+1)^{k-1} 2^{k\tau} < [(d+1)2^\tau]^k,$$

ce qui prouve l'inégalité (2.15).

La suite de $f_{c,j}$ est déterminée par les formules de (2.9), pour chaque $c = 1, \ldots, \ell$, et chaque j dans $[0, \ldots, 2^{\ell-c} - 1]$, nous effectuons la multiplication $f_{c-1,2j} \cdot f_{c-1,2j+1}$ de polynômes de degrés inférieurs à $d \cdot 2^{c-1}$ et de taille binaire inférieure à $2^{c-1}[\tau + \log_2(d+1)]$(lemme 2.27). Chaque produit s'effectue en $\mathcal{M}(2^{c-1}d)$ opérations arithmétiques et $\widetilde{\mathcal{O}}\Big(2^{c-1}d \cdot 2^{c-1}[\tau + \log_2(d+1)]\Big) = \widetilde{\mathcal{O}}(2^{2c}d\tau)$ opérations binaires. Le résultat du produit est de taille binaire inférieure à $2^c[\tau + \log_2(d+1)]$, en utilisant le lemme 2.27.

Finalement, à l'étape c nous effectuons $2^{\ell-c}\mathcal{M}(2^{c-1}d) = \widetilde{\mathcal{O}}(2^{\ell-1}d)$ opérations arithmétiques, ou $2^{\ell-c}\widetilde{\mathcal{O}}(2^{2c}d\tau) = \widetilde{\mathcal{O}}(2^{\ell+c}d\tau)$ opérations binaires.

Nous obtenons alors notre produit en $\sum_{c=1}^{\ell} \widetilde{\mathcal{O}}(2^{\ell-1}d) = \widetilde{\mathcal{O}}(kd)$ opérations arithmétiques et $\sum_{c=1}^{\ell} \widetilde{\mathcal{O}}(2^{\ell+c}d\tau) = \widetilde{\mathcal{O}}(k^2 d\tau)$ opérations binaires, car $k = 2^\ell$. □

2.3.2 Composition dans la base de monômes

Une autre application du principe *Diviser pour Régner* est la composition de deux polynômes.

Étant donnés deux polynômes f et g dans $\mathbb{Z}[x]$ dans la base des monômes, plusieurs méthodes ont été recommandées afin d'évaluer précisément l'expression de $f(g(x))$: La méthode de Horner [Knu97] ; La composition via la base de polynômes de Berstein [dB87] ; L'algorithme basé sur l'évaluation des points engendrés par interpolation [Pan01, page 81] etc. Le travail le plus récent [HN11] de Hart et Novocin était publié en 2011.

Commençons par évaluer la taille de nos objets.

Lemme 2.29. *Soit P et g deux éléments de $\mathbb{R}[x]$, avec $\deg P = k$, $\deg g = d'$, $\|g\|_\infty \geq 1$, alors*

$$\|P(g(x))\|_\infty \leq \|P\|_\infty \|g\|_\infty^{k+1} (d'+1)^k. \tag{2.16}$$

Démonstration. Posons $P(x) = p_0 + \ldots + p_k x^k$. Pour tout $j = 1, \ldots, k$, en appliquant l'inégalité (2.14) pour $f_1 = \ldots = f_j = g$ on a $\|g^j\|_\infty \leq (d'+1)^{j-1}\|g\|_\infty^j$. On en déduit que

$$\|P(g(x))\|_\infty \leq \sum_{j=0}^{k} \|p_j g^j\|_\infty \leq \|P\|_\infty \sum_{j=0}^{k} \|g^j\|_\infty.$$

En utilsant $(d'+1)\|g\|_\infty - 1 \geq 1$, on déduit que $\|g^j\|_\infty \leq (d'+1)^j \|g\|_\infty^{j+1} - (d'+1)^{j-1}\|g\|_\infty^j$ et donc

$$\sum_{j=0}^{k} \|g^j\|_\infty \leq \sum_{j=0}^{k} \Big((d'+1)^j \|g\|_\infty^{j+1} - (d'+1)^{j-1}\|g\|_\infty^j\Big) \leq (d'+1)^k \|g\|_\infty^{k+1}.$$

et le résultat s'en déduit. □

Dans [HN11], les auteurs ont choisi de diviser par $m = 4$. En supposant que $\tau(f), \tau(g) = \mathcal{O}(d')$, ils ont obtenu [HN11, Théo. 2] affirmant que :

Le calcul de $f \circ g$ en $\widetilde{\mathcal{O}}(dd')$ opérations arithmétiques, $\mathcal{O}(d^2 d'^2 \log_2 d') = \widetilde{\mathcal{O}}(d^2 d'^2)$ opérations binaires.

Ici la condition $\tau(f), \tau(g) = \mathcal{O}(d')$ n'est plus nécessaire. Nous obtenons :

Proposition 2.30. *Soit f et g deux polynômes dans $\mathbb{Z}[x]$, $\deg f = d$, $\deg g = d'$, $\tau(f) = \tau$, $\tau(g) = \tau'$ alors*

$$\tau(f(g)) < \tau + (d+1)[\tau' + \log_2(d'+1)] = \tau + \mathcal{O}(d(\tau' + \log_2 d')); \tag{2.17}$$

On peut écrire $f(g(x))$ dans la base des monômes en $\widetilde{\mathcal{O}}(dd')$ opérations arithmétiques, $\widetilde{\mathcal{O}}(dd'(\tau + d\tau'))$ opérations binaires.

Démonstration.
1. La taille binaire $\tau(f(g))$ se déduit du lemme 2.29, en appliquant $\tau(f(g)) = \lceil \log_2 \|f(g)\|_\infty \rceil$.
2. Étudions maintenant la complexité. En utilisant le calcul selon le schéma (2.11) et en substituant $v = g$, nous devons calculer d'une part, les g^{2^c}, pour $c = 1, \ldots, \ell$, où $\ell = \lceil \log_2 k \rceil$ et les $S_{c,j} = S_{c-1,2j} + S_{c,2j+1} \cdot g^{2^{c-1}}$ d'autre part. Avec un indice c, dénotons $\tau_c = \max\{\tau(S_{c,j}), j\}$.

2.1. Calculons d'abord les $g^{2^c}, c = 1, \ldots, \ell$. Pour chaque c nous déduisons g^{2^c} de $g^{2^{c-1}}$ en $\mathcal{M}(2^{c-1}d', 2^c\tau') = \widetilde{\mathcal{O}}(2^{2c}d'\tau')$opérations binaires ou en $\mathcal{M}(2^{c-1}d')$ opérations arithmétiques.

D'autre part nous avons :

† $\deg g^{2^{c-1}} = 2^{c-1}d'$;

† Appliquons le lemme 2.27 pour $f_0 = \ldots = f_{2^{c-1}-1} = g$, on obtient

$$\tau(g^{2^{c-1}}) \leq 2^{c-1}[\tau' + \log_2(d'+1)]. \tag{2.18}$$

Le calcul de tous les $g^{2^c}, c = 1, \ldots, \ell$ utilise donc :

$$\begin{aligned} C_A^{(1)} &= \sum_{c=0}^{\ell-1} \mathcal{M}(2^c d') = \sum_{c=0}^{\ell-1} \widetilde{\mathcal{O}}(2^c d') = \widetilde{\mathcal{O}}(2^\ell d') = \widetilde{\mathcal{O}}(dd') \\ &\quad \text{opérations arithmétiques;} \\ C_B^{(1)} &= \sum_{c=0}^{\ell-1} \mathcal{O}(\mathcal{M}(2^c d', \tau(g^{2^c}))) = \sum_{c=0}^{\ell-1} \widetilde{\mathcal{O}}(2^c d'(2^c(\tau' + \log_2(d'+1))) \\ &= \sum_{c=0}^{\ell-1} \widetilde{\mathcal{O}}(4^c d'\tau') = \widetilde{\mathcal{O}}(4^\ell d'\tau') = \widetilde{\mathcal{O}}(d^2 d'\tau') \\ &\quad \text{opérations binaires.} \end{aligned}$$

2.2. Nous devons calculer ensuite, quand c étant fixé, les $2^{\ell-c}$ produits

$$S_{c,j} = S_{c-1,2j} + S_{c-1,2j+1} g^{2^{c-1}}.$$

Pour chacun de ces produits, on a $\deg S_{c-1,j} \leq 2^{c-1}d'$ et comme $S_{c-1,j}$ est déterminé par la formule (2.13a), on peut appliquer le lemme 2.29 pour obtenir :

$$\begin{aligned} \tau_{c-1} &= \max\{\tau(S_{c-1,j}), j\} \\ &\leq \tau + (2^{c-1}+1)(\tau' + \log_2(d'+1)) = \tau + \mathcal{O}\Big(2^{c-1}(\tau' + \log_2 d')\Big) ; \end{aligned} \tag{2.19}$$

En résumé, chaque calcul de $S_{c,j}$ contient une multiplication et une addition polynomiale dont la taille d'entrées est expliquée ci-dessous :

† $\deg S_{c-1,2j}, \deg S_{c-1,2j+1} \leq (2^{c-1}-1)d'$, vu la formule (2.13a) avec $v = g$; τ_{c-1} est borné comme expliquée dans l'inégalité (2.19) ;

† $\deg g^{2^{c-1}} = 2^{c-1}d'$, $\tau(g^{2^{c-1}})$ est bornée par $2^{c-1}[\tau' + \log_2(d'+1)]$ comme confirmée par l'inégalité (2.18) ;

† La taille binaire du produit est évaluée par l'inégalité (2.3b). Avec la taille de $S_{c-1,2j+1}$ et de $g^{2^{c-1}}$ qui vient d'être évaluée on obtient

$$\tau(S_{c-1,2j+1}g^{2^{c-1}}) \leq \tau_{c-1} + \tau(g^{2^{c-1}}) + \log_2(2^c d') = \tau + \widetilde{\mathcal{O}}\Big(2^c(\tau' + \log_2(d'+1)\Big). \quad (2.20)$$

Le calcul de tous les $S_{c,j}$ utilise donc :

$$C_A^{(2)} = \sum_{c=1}^{\ell} 2^{\ell-c}\Big(\mathcal{M}(2^{c-1}d') + 2^{c-1}d'\Big) = \sum_{c=1}^{\ell} 2^{\ell-c}\widetilde{\mathcal{O}}(2^{c-1}d) = \sum_{c=1}^{\ell} 2^{\ell-c}\widetilde{\mathcal{O}}(2^{c-1}d) = \widetilde{\mathcal{O}}(dd')$$

opérations arithmétiques;

$$\begin{aligned}
C_B^{(2)} &= \sum_{c=1}^{\ell} 2^{\ell-c}\Big(\underbrace{\mathcal{O}(\mathcal{M}(2^{c-1}d', \max\{\tau_{c-1}, \tau(g^{2^{c-1}})\}))}_{\text{coût de la multiplication}} + \underbrace{2^{c-1}d' \max\{\tau_{c-1}, \tau(S_{c-1,2j+1}g^{2^{c-1}})\}}_{\text{coût de l'addition}}\Big) \\
&= \widetilde{\mathcal{O}}\left(\sum_{c=1}^{\ell} 2^{\ell-1}d'\Big[\max\{\tau_{c-1}, \tau(g^{2^{c-1}})\} + \max\{\tau_{c-1}, \tau(S_{c-1,2j+1}g^{2^{c-1}})\}\Big]\right) \\
&= \widetilde{\mathcal{O}}\left(\sum_{c=1}^{\ell} dd'\Big[\tau + \widetilde{\mathcal{O}}\Big(2^c[\tau' + \log_2(d'+1)]\Big)\Big]\right), \text{ utilisons les évaluations (2.18), (2.19) et (2.20)} \\
&= \widetilde{\mathcal{O}}(dd'(\tau + d\tau')) \text{ opérations binaires.}
\end{aligned}$$

Finalement la complexité de la composition est :

$$\begin{aligned}
C_A &= C_A^{(1)} + C_A^{(2)} = \widetilde{\mathcal{O}}(dd') && \text{opérations arithmétiques,} \\
C_B &= C_B^{(1)} + C_B^{(2)} = \widetilde{\mathcal{O}}(dd'(\tau + d\tau')) && \text{opérations binaires.}
\end{aligned}$$

□

2.4 Changement de base

Une fois donnée une base $\mathcal{B}$ de $K[x]$, le problème du changement de base est naturellement demandé, ce qui comprend deux sens :

Expand$_n$ Étant donnés $\alpha_0, \ldots, \alpha_{n-1} \in K$, calculer les coefficients dans la base des monômes du $A = \sum_{j=0}^{n-1} \alpha_j B_j$;

Decomp$_n$ Réciproquement, étant donné le polynôme A dans la base des monômes, retrouver les α_j, c'est-à-dire décomposer A comme une combinaison linéaire des B_j.

L'algorithme naïf demande $\mathcal{O}(n^2)$ opérations arithmétiques dans K pour les deux sens, pourtant il existe des méthodes plus rapides.

Il y a des algorithmes rapides connus pour quelques problèmes étroitement liés qui permettent de déduire rapidement des algorithmes pour **Expand**$_n$, par exemple : la question est de calculer les valeurs $[\alpha_0, \ldots, \alpha_{n-1}] \mapsto \Big[\sum_{i=0}^{n-1} \alpha_i B_i(x_j)\Big]_{0 \leq j < n}$, où $x_j = \cos \frac{j\pi}{n-1}$

dans [PST98], ou bien le problème transposé de calculer l'application $[\alpha_0, \ldots, \alpha_{n-1}] \mapsto \left[\sum_{i=0}^{n-1} \alpha_i B_j(x_i)\right]_{0 \le j < n}$ mentionné dans [DHJR97].

En ce qui concerne le problème $\mathbf{Decomp}_n$, des travaux ont traité le cas d'une base particulière comme [Fru95] (publié en 1995, pour la base de Legendre), [Pan98] (en 1998, pour la base de Chebyshev), [LRC07] (en 2007, pour la base de Hermite) pour approcher la complexité arithmétique $\mathcal{O}(\mathcal{M}(n) \log_2 n)$.

En 2008, Bostan, Salvy et Schost, dans [BSS08] ont résolu les deux problèmes sur quelques bases classiques comme celles de Jacobi, Hermite, Laguerre, etc. à la fois, et en 2010, ces auteurs arrivent à une conclusion générale pour toutes les bases orthogonales :
Théorème [BSS10, Théo. 1] *Les deux problèmes* $\mathbf{Expand}_n$ *et* $\mathbf{Decomp}_n$ *peuvent être résolus en complexité arithmétique* $C_A = \mathcal{O}(\mathcal{M}(n) \log_2 n)$.

Parlons particulièrement de la base des polynômes de Chebyshev de type original, Benoît [Ben12, Chap. 3], en 2012, est parti de l'idée dans [BSS08] pour proposer une méthode permettant de faire ces changements de base en $\mathcal{O}(\mathcal{M}(n))$ opérations arithmétiques.

Comme notre base $\mathcal{T}$ est une base orthogonale de $\mathbb{Z}[x]$, on est dans un cas particulier du changement de base. Il est donc impossible d'améliorer la complexité arithmétique du calcul. Par ailleurs, nous nous inspirons la méthode de Bostan et al. [BSS10] afin de construire des algorithmes simples en utilisant la particularité des polynômes unitaires de Chebyshev, et d'analyser la complexité binaire du calcul.

2.4.1 Calculer la forme de Chebyshev d'un polynôme

Une fois que l'on sait multiplier rapidement dans la base des polynômes unitaires de Chebyshev, cette multiplication nous aidera à transformer de la base $\mathcal{X}$ à la base $\mathcal{T}$.

Étant donné :

$$f(x) = \sum_{j=0}^{d} a_j x^j, \quad d = 2^\ell - 1, \quad \tau(f) = \tau,$$

nous allons calculer $[\mathcal{T}]f$.

En déployant le schéma (2.11) avec $v = T_1$ et la multiplication dans la base des polynômes unitaires de Chebyshev, la suite à calculer successivement devient :

$$\begin{aligned} S_{0,j} &= a_j, \quad j = 0, \ldots, 2^\ell - 1 && v_0 = T_1, \\ S_{c,j} &= [\mathcal{T}]\Big(S_{c-1,2j} + S_{c-1,2j+1} \cdot v_{c-1}\Big), \quad j = 0, \ldots, 2^{\ell-c} - 1 && v_c = [\mathcal{T}]v_{c-1}^2 \\ & \qquad \text{pour } c = 1, \ldots, \ell, \\ \text{enfin :} \quad & [\mathcal{T}]f = S_{\ell,0}. \end{aligned} \tag{2.21}$$

Il est toujours vrai que chaque $S_{c,j}$ est la forme de Chebyshev déterminée sur un segment de longueur 2^c de f, chaque v_c est la forme de Chebyshev d'une puissance de x :

$$S_{c,j} = [\mathcal{T}] \sum_{i=0}^{2^c-1} a_{j \cdot 2^c + i} x^i; \tag{2.22a}$$

$$v_c = [\mathcal{T}] x^{2^c}; \tag{2.22b}$$

L'algorithme 2.31 réalise ce calcul permettant de trouver la forme de Chebyshev de f.

Algorithme 2.31 : Calculer la forme de Chebyshev

Entrées : $f = \sum_{i=0}^{d} a_i x^i$
Sorties : $[\mathcal{T}]f = f_0 + \sum_{i=1}^{d} f_i T_i$

```
begin
    ℓ ← ⌊d/2⌋ ; c ← 1 ; v ← T_2 + 2 ;
    pour j ← 0 a ℓ faire
        S_{c,j} ← a_{2j} + a_{2j+1} · T_1
    répéter
        c ← c + 1;
        pour j ← 0 a ⌊ℓ/2⌋ faire
            S_{c,j} ← [T](S_{c-1,2j} + S_{c-1,2j+1} · v)
        v ← [T](v^2);
        ℓ ← ⌊ℓ/2⌋
    jusqu'à ce que ℓ = 0;
    retourner S_{c,0}
```

Preuve de correction. Il est évidence que l'algorithme va sortie la forme de Chebyshev de f après $\lfloor \log_2 n \rfloor$ étapes. □

Lemme 2.32. *Soit n un entier positif alors on a :*

$$\|[\mathcal{T}]x^n\|_\infty < 2^n \quad ; \quad \tau_{\mathcal{T}}(x^n) < n. \tag{2.23}$$

Démonstration. Rappelons l'identité (1.2a) :

$$x^n = \sum_{k=0}^{\lfloor n/2 \rfloor - 1} \binom{n}{k} T_{n-2k}(x) + \frac{1+(-1)^n}{2} \binom{n}{\lfloor n/2 \rfloor}.$$

Pour tout k on a $\binom{n}{k} < 2^n$ donc $\|\tau_{\mathcal{T}}(x^n)\|_\infty < 2^n$ et $\tau_{\mathcal{T}}(x^n) < n$. □

Proposition 2.33 (Changement de $\mathcal{X}$ à $\mathcal{T}$)**.** *Soit f un polynôme à coefficients entiers dans la base de monômes,* $\deg f = d$ *alors :*

$$\tau_{\mathcal{T}}(f) \leq \tau(f) + d + 1; \tag{2.24}$$

On peut calculer la forme de Chebyshev $[\mathcal{T}]f$ en $\widetilde{\mathcal{O}}(d)$ opérations arithmétiques, $\widetilde{\mathcal{O}}(d^2 + d\tau)$ opérations binaires.

Démonstration. 1. En ayant $f = \sum_{j=0}^{d} a_j x^j$, $|a_j| < 2^\tau \, \forall j = 0, \ldots, d$, nous regroupons les coefficients de T_k (k=1,...,d) : dans $a_j x^j$ il y en a au plus

$$a_j \|[\mathcal{T}]x^j\|_\infty < a_j \cdot 2^j < 2^{\tau j},$$

alors le coefficient $[T_k]f$ est borné : $|[T_k]f| < \left(\sum_{j=0}^{d} 2^j\right) 2^\tau < 2^{d+1} 2^\tau$. On en déduit :

$$\|[\mathcal{T}]f\|_\infty < 2^{\tau+d+1},$$

ce qui prouve (2.24)

2. Afin d'évaluer la complexité, nous utilisons l'algorithme 2.31 en considérant que $d = 2^\ell - 1$, le calcul peut être partagé en deux parties :

2.1. Calculer tous les $[\mathcal{T}]x^{2^c}, c = 1, \ldots, \ell$ est fait successivement par $[\mathcal{T}]x^{2^{c+1}} = [\mathcal{T}](x^{2^c})^2, c = 0, \ldots, \ell - 1$. Appliquons la majoration de la taille du produit (proposition 2.17) avec $\deg x^{2^c} = 2^c$, $\tau_{\mathcal{T}}(x^{2^c}) < 2^c$ (lemme 2.32), sous l'aide de la proposition 2.6 et du corollaire 2.7, ce calcul utilise :

$$C_A^{(1)} = \sum_{c=0}^{\ell-1} \mathcal{M}(2^c) = \sum_{c=0}^{\ell-1} \widetilde{\mathcal{O}}(2^c) = \widetilde{\mathcal{O}}(2^\ell) = \widetilde{\mathcal{O}}(d) \qquad \text{opérations arithmétiques,}$$

$$C_B^{(1)} = \sum_{c=0}^{\ell-1} \mathcal{M}(2^c, 2^c) = \sum_{c=0}^{\ell-1} \widetilde{\mathcal{O}}(4^c) = \widetilde{\mathcal{O}}(d^2) \qquad \text{opérations binaires ;}$$

2.2. Calculer tous les $S_{c,j}$: pour chaque $c = 1, \ldots, \ell$ nous faisons $2^{\ell-c}$ fois le calcul $S_{c,j} = [\mathcal{T}](S_{c-1,2j} + S_{c-1,2j+1}x^{2^{c-1}})$ i.e. une addition et une multiplication dans $\mathbb{Z}[x]$ en une fois, dans la base $\mathcal{T}$, dont les termes sont de taille comme suit :

† $[\mathcal{T}]x^{2^{c-1}}$: $\deg x^{2^{c-1}} = 2^{c-1}$, et d'après le lemme 2.32 :

$$\tau_{\mathcal{T}}(x^{2^{c-1}}) \leq 2^{c-1} \tag{2.25}$$

† $S_{c-1,2j}, S_{c-1,2j+1}$: vu la définition (2.22a) $\deg S_{c-1,2j}, \deg S_{c-1,2j+1} \leq 2^{c-1} - 1$, en plus avec l'évaluation (2.24) :

$$\tau_{c-1} = \max\{\tau_{\mathcal{T}}(S_{c-1,j})j = 0, 2^{\ell-c+1} - 1\} \leq \tau + 2^{c-1} + 1; \tag{2.26}$$

† Selon l'inégalité (2.4), le produit $[\mathcal{T}](S_{c-1,2j+1} \cdot x^{2^{c-1}})$ possède les coefficients de taille binaire bornée par :

$$\tau_{\mathcal{T}}(S_{c-1,2j+1} \cdot x^{2^{c-1}}) \leq \tau_{c-1} + \tau_{\mathcal{T}}(x^{2^{c-1}}) + \log_2(2^{c-1}) + 1 = \tau + \mathcal{O}(2^c) \tag{2.27}$$

Sur les opérations dans la base des polynômes unitaires de Chebyshev, utilisons la proposition 2.17 pour la complexité de la multiplication, avec les bornes des tailles des coefficients (2.25), (2.26), (2.27), alors le calcul de tous les $S_{c,j}$, $c = 1, \ldots, \ell$ utilise :

$$C_A^{(2)} = \sum_{c=1}^{\ell} 2^{\ell-c}(\mathcal{M}(2^{c-1}) + 2^{c-1} - 1) = \sum_{c=1}^{\ell} \widetilde{\mathcal{O}}(2^{\ell-1}) = \widetilde{\mathcal{O}}(d) \text{ opérations arithmétiques ;}$$

$$\begin{aligned} C_B^{(2)} &= \sum_{j=1}^{\ell-1} 2^{\ell-c}\Big(\underbrace{\widetilde{\mathcal{O}}(2^{c-1}\max\{\tau_{c-1}, \tau_{\mathcal{T}}(x^{2^{c-1}})\})}_{\text{coût de la multiplication}} + \underbrace{2^{c-1}\max\{\tau_{c-1}, \tau_{\mathcal{T}}(S_{c-1,2j+1}x^{2^{c-1}})\}}_{\text{coût de l'addition}}\Big) \\ &= \widetilde{\mathcal{O}}\Big(2^{\ell-1}\sum_{c=1}^{\ell}(\tau + 2^{c-1}) + 2^{\ell-1}\sum_{c=1}^{\ell}(\tau + \mathcal{O}(2^c))\Big) \\ &= \widetilde{\mathcal{O}}(2^{\ell-1}(\tau + \mathcal{O}(\sum_{c=1}^{\ell} 2^{c-1}))) = \widetilde{\mathcal{O}}(d^2 + d\tau) \text{ opérations binaires ;} \end{aligned}$$

En résumé, la complexité globale du calcul est :

$$C_A = C_A^{(1)} + C_A^{(2)} = \widetilde{\mathcal{O}}(d) \qquad \text{opérations arithmétiques,}$$

$$C_B = C_B^{(1)} + C_B^{(2)} = \widetilde{\mathcal{O}}(d^2 + d\tau) \qquad \text{opérations binaires.}$$

□

2.4.2 Développer une forme de Chebyshev

Étant donné la forme de Chebyshev du polynôme $f \in \mathbb{Z}[x]$:

$$f(x) = f_0 + \sum_{j=1}^{d} f_j T_j(x), \quad d = 2^\ell - 1 \tag{2.28}$$

pour obtenir son expression dans la base $\mathcal{X}$ des monômes, l'idée est d'utiliser une récurrence matricielle permettant d'appliquer le schéma de calcul (2.11).

Comme Bostan et al. ont fait dans [BSS10], la récurrence $T_n = xT_{n-1} - T_{n-2}$ implique que

$$\begin{bmatrix} T_i \\ T_{i+1} \end{bmatrix} = X \begin{bmatrix} T_{i-1} \\ T_i \end{bmatrix}, \text{ où } X = \begin{bmatrix} 0 & 1 \\ -1 & x \end{bmatrix}.$$

$$\Rightarrow \begin{bmatrix} T_{i-1} \\ T_i \end{bmatrix} = X^i \begin{bmatrix} T_0 \\ T_1 \end{bmatrix} \ \forall i \in \mathbb{N}.$$

Parce que $T_0 = 2$, la formule (2.28) devient :

$$\begin{aligned}
[f] + f_0 &= \begin{bmatrix} f_0 & f_1 \end{bmatrix} \begin{bmatrix} T_0 \\ T_1 \end{bmatrix} + \begin{bmatrix} f_2 & f_3 \end{bmatrix} \begin{bmatrix} T_2 \\ T_3 \end{bmatrix} + \ldots + \begin{bmatrix} f_{d-1} & f_d \end{bmatrix} \begin{bmatrix} T_{d-1} \\ T_d \end{bmatrix} \\
&= \begin{bmatrix} f_0 & f_1 \end{bmatrix} \begin{bmatrix} T_0 \\ T_1 \end{bmatrix} + \begin{bmatrix} f_2 & f_3 \end{bmatrix} X^2 \begin{bmatrix} T_0 \\ T_1 \end{bmatrix} + \ldots + \begin{bmatrix} f_{d-1} & f_d \end{bmatrix} (X^2)^{2^{\ell-1}-1} \begin{bmatrix} T_0 \\ T_1 \end{bmatrix} \\
&= \left[\begin{bmatrix} f_0 & f_1 \end{bmatrix} + \begin{bmatrix} f_2 & f_3 \end{bmatrix} X^2 + \ldots + \begin{bmatrix} f_{d-1} & f_d \end{bmatrix} (X^2)^{2^{\ell-1}-1} \right] \begin{bmatrix} T_0 \\ T_1 \end{bmatrix}.
\end{aligned} \tag{2.29}$$

Le problème se ramène à calculer la matrice f^* de taille 1×2 comme suit :

$$f^* = \begin{bmatrix} f_0 & f_1 \end{bmatrix} + \begin{bmatrix} f_2 & f_3 \end{bmatrix} X^2 + \ldots + \begin{bmatrix} f_{d-1} & f_d \end{bmatrix} (X^2)^{2^{\ell-1}-1}. \tag{2.30}$$

Divisons par 2 pour régner, comme expliqué par le schéma (2.11), la suite des calculs s'explique par :

$$\begin{aligned}
S_{0,j} &= \begin{bmatrix} f_{2j} & f_{2j+1} \end{bmatrix}, && j = 0, \ldots, 2^{\ell-1} - 1 && v_0 = X^2, \\
S_{c,j} &= S_{c-1,2j} + S_{c-1,2j+1} \cdot v_{c-1}, && j = 0, \ldots, 2^{\ell-c-1} - 1 && v_c = v_{c-1}^2 \\
& && \text{pour } c = 1, \ldots, \ell - 1, \\
& \text{enfin :} && f^* = S_{\ell-1,0}.
\end{aligned} \tag{2.31}$$

Signalons que

$$v_c = X^{2^{c+1}}. \tag{2.32}$$

On considère d'abord le calcul des X^m :

Lemme 2.34.

1. *Pour tout $m \in \mathbb{Z}_{>0}$ on a $\tau(X^m) \leq m$;*
2. *Soit $d = 2^\ell - 1$ alors il est possible de calculer tous les X^{2^c}, $c = 0, \ldots, \ell - 1$ en $\widetilde{\mathcal{O}}(d)$ opérations arithmétiques, $\widetilde{\mathcal{O}}(d^2)$ opérations binaires.*

Démonstration. Car $U_0 = 0, U_1 = 1$, $U_{n+1} = xU_n - U_{n-1}$:

$$X = \begin{bmatrix} 0 & 1 \\ -1 & x \end{bmatrix} = \begin{bmatrix} -U_0 & U_1 \\ -U_1 & U_2 \end{bmatrix},$$

par une simple induction nous avons :

$$X^n = \begin{bmatrix} -U_{n-1} & U_n \\ -U_n & U_{n+1} \end{bmatrix}. \tag{2.33}$$

Mais on a la formule (1.3) : $U_m = \frac{1}{m}T'_m$ et d'après la preuve du lemme 2.12 $\|T_m\|_\infty \leq 2^{m-1}$, alors pour tout m,

$$\tau(U_m) \leq m - 1, \tag{2.34}$$

ou bien

$$\tau(X^m) \leq m.$$

Pour calculer tous les X^{2^c}, remarquons que l'on a les deux formules suivantes à l'aide des transformations trigonométriques élémentaires :

$$2U_{n+1}(x) = xU_n(x) + T_n(x); \tag{2.35a}$$

$$T_n(x) = U_{n+1}(x) + U_{n-1}(x). \tag{2.35b}$$

Donc, il est possible d'obtenir tous les X^{2^c}, $c = 0, \ldots \ell$ avec l'algorithme 2.35 :

Algorithme 2.35 : Calculer X^m

Entrées : $\ell \in \mathbb{Z}_{>0}$
Sorties : $\{X^{2^c}, c = 0, \ldots, \ell\}$

```
1 begin
2     Calculer tous les T_{2^c}, c = 0, ..., ℓ;
3     pour m ∈ {2^c, c = 1, ..., ℓ} faire
4         U_m ← (1/m) T'_m ; /* Vu (la relation 1.3)                    */
5         U_{m+1} ← (1/2)(x U_m + T_m) ;/* Vu la relation (2.35a)       */
6         U_{m-1} ← T_m − U_{m+1} ; /* Vu la relation (2.35b)           */
7         retourner X^m = [ −U_{m-1}  U_m ; −U_m  U_{m+1} ]
```

On sait que tous les $T_{2^c}, c = 0, \ldots, \ell$ peuvent être calculés en $\widetilde{\mathcal{O}}(d)$ opérations arithmétiques, $\widetilde{\mathcal{O}}(d^2)$ opérations binaires (lemme 2.12), ce qui prouve le lemme 2.34. □

L'algorithme 2.36 suivant nous permet de faire le changement de $\mathcal{T}$ à $\mathcal{X}$:

Proposition 2.37 (Changement de $\mathcal{T}$ à $\mathcal{X}$)**.** *Soit* $[\mathcal{T}]f$ *une forme de Chebyshev dans* $\mathbb{Z}[x]$ *tel que* $\deg f = d$, $\tau_{\mathcal{T}}(f) = \tau$. *Alors*

$$\tau(f) \leq \tau_{\mathcal{T}}(f) + d + 1; \tag{2.36}$$

Il est possible d'écrire f *dans la base des monômes en* $\widetilde{\mathcal{O}}(d)$ *opérations arithmétiques,* $\widetilde{\mathcal{O}}(d^2 + d\tau)$ *opérations binaires.*

Algorithme 2.36 : Développer une forme de Chebyshev

Entrées : $f = f_0 + \sum_{i=1}^{d} f_i T_i$
Sorties : $[\mathcal{X}]f = \sum_{i=0}^{d} a_i x^i$

```
begin
    ℓ ← ⌊d/2⌋ ; c ← 0 ; V ← X^2 ;
    pour j ← 0 a ℓ faire
        S_{c,j} ← [f_{2j}  f_{2j+1}]
    répéter
        ℓ ← ⌊ℓ/2⌋;
        pour j ← 0 a ℓ faire
            S_{c+1,j} ← S_{c,2j} + S_{c,2j+1} · V
        c ← c + 1 ; V ← [X]V^2;
    jusqu'à ce que ℓ = 0;
    retourner S_{c,0} · [2; x] − f_0
```

Démonstration.

1. Pour la taille des coefficients de $[\mathcal{X}]f$, remarquons qu'on a $\tau(T_n) \leq n-1$ (vu la preuve du lemme 2.12). Dans la formule (2.28), pour tout $1 \leq k \leq d$, $[x^k]T_j \leq 2^j$ $\forall j = 0, \ldots, d$. En collectant les coefficient de x^k on a :

$$[x^k]f \leq \sum_{j=0}^{d} |f_j| \|[\mathcal{X}]T_j\|_\infty \leq \|[\mathcal{T}]f\|_\infty \sum_{j=0}^{d} 2^j \leq 2^\tau 2^{d+1},$$

on en déduit donc l'évaluation (2.36).

2. Afin d'évaluer la complexité, il y a des données intermédiaires à pré-évaluer :

2.1. D'après le lemme 2.34, le calcul de tous les $X^{2^c}, c = 0, \ldots, \ell-1$ utilise $C_A^{(1)} = \widetilde{\mathcal{O}}(d)$ opérations arithmétiques, $C_B^{(1)} = \widetilde{\mathcal{O}}(d^2)$ opérations binaires.

2.2. En ce qui concerne la taille des $S_{c,j}$, $c = 0, \ldots, \ell$, $j = 0, \ldots, 2^{\ell-c-1}$: $S_{c,j}$ est une matrice de taille 1×2, s'écrit $S_{c,j} = \begin{bmatrix} P_{c,j} & Q_{c,j} \end{bmatrix}$. Vu deux formules (2.31) et (2.33) :

$$\begin{aligned} \begin{bmatrix} P_{c,j} & Q_{c,j} \end{bmatrix} &= \begin{bmatrix} P_{c-1,2j} & Q_{c-1,2j} \end{bmatrix} + \begin{bmatrix} P_{c-1,2j+1} & Q_{c-1,2j+1} \end{bmatrix} X^{2^c} \\ &\overset{(2.33)}{=} \begin{bmatrix} P_{c-1,2j} - P_{c-1,2j+1}U_{2^c-1} - Q_{c-1,2j+1}U_{2^c} & P_{c-1,2j} + P_{c-1,2j+1}U_{2^c} - Q_{c-1,2j+1}U_{2^c+1} \end{bmatrix} \end{aligned} \tag{2.37}$$

Considérons l'identité (2.37) on peut remarquer deux informations concernent $S_{c,j}$:

(i) Car $\deg P_{0,j} = \deg Q_{0,j} = 0 (j = 1, \ldots 2^{\ell-1} - 1)$, par l'induction sur l'identité (2.37) nous arrivons :

$$\deg P_{c,j} \leq 2^c - 3, \deg Q_{c,j} \leq 2^c - 2 \quad (j = 0, \ldots, 2^{\ell-c-1}). \tag{2.38}$$

(ii) Ensuite évaluons $\tau_c = \max\{\tau(S_c, j), j = 0, \ldots, 2^{\ell-c-1}\}$ pour $c = 0, \ldots, \ell-1$. En appliquant la remarque 2.16 pour l'identité (2.37) on a :

$$\tau_c \leq \tau_{c-1} + \tau(U_{2^c+1}) + \log_2 \deg U_{2^c+1} + 2 = \tau_{c-1} + 2^c + c + 3 \tag{2.39}$$

Ecrivons l'inégalité (2.39) successivement jusqu'au $\tau_0 = \tau$ on obtient :

$$\tau_c \leq \tau_0 + \sum_{j=0}^{c-1} 2^j + \sum_{j=0}^{c-1}(j+3) = \tau + 2^c + \tfrac{c^2+5c-2}{2} = \tau + \mathcal{O}(2^c). \tag{2.40}$$

2.3. Brièvement, pour chaque $c = 1, \ldots, \ell$, nous faisons $2^{\ell-c-1}$ calculs expliqués par la formule (2.37) dont leurs entrées sont de taille comme suit :

† $\deg U_m = m-1$, $\tau(U_m) \leq m$;

† $\deg P_{c,j} \leq 2^c - 3, \deg Q_{c,j} \leq 2^c - 2$;

† $\tau_{c-1} = \max(\tau(P_{c-1,j}), \tau(Q_{c-1,j}), j = 1, \ldots, 2^{\ell-c})$ est majorée par $\tau + \mathcal{O}(2^c)$ comme indiqué dans l'inégalité (2.40) ;

† La taille binaire des produits présentés dans la formule (2.37) peut être bornée grâce à l'évaluation (2.3b) :

$$\max\{\tau(P_{c-1,2j+1}U_{2^c-1}), \tau(Q_{c-1,2j+1}U_{2^c}), \tau(P_{c-1,2j+1}U_{2^c}), \tau(Q_{c-1,2j+1}U_{2^c+1})\} \leq \tau_{c-1} + 2^c + \log_2(2^c+1) = \tau + \mathcal{O}(2^c). \tag{2.41}$$

La complexité de la multiplication et de l'addition dans la base des monômes est mentionnée par la proposition 2.6 et le corollaire 2.7. Nous appliquons deux résultats avec les données de degré au plus 2^c+1 possédant les coefficients de taille binaire $\tau + \mathcal{O}(2^c)$. Alors la complexité du calcul de tous les $S_{c,j}$ est :

$$\begin{aligned}
C_A^{(2)} &= \sum_{c=1}^{\ell}\Big(2^{\ell-c-1}.(4\mathcal{M}(2^{c-1}) + 2.2^{c-1})\Big) \\
&= \ell 2^{\ell-1} + \ell\widetilde{\mathcal{O}}(2^\ell) = \widetilde{\mathcal{O}}(d) \text{ opérations arithmétiques;} \\
C_B^{(2)} &= \sum_{c=1}^{\ell} 2^{\ell-c-1}.(4\mathcal{M}(2^{c-1}, \tau_{c-1}) + 2^c(\tau + \mathcal{O}(2^c))), \text{ par l'évaluation (2.41)} \\
&= 2^\ell\widetilde{\mathcal{O}}\Big(\sum_{c=1}^{\ell}(\tau + \mathcal{O}(2^c)\Big) + \ell 2^{\ell-1}\tau + 2^\ell\mathcal{O}\Big(\sum_{c=1}^{\ell} 2^{c-1}\Big) \\
&= \widetilde{\mathcal{O}}(\ell 2^\ell\tau + 2^\ell(2^{\ell+1})\tau + \ell 2^{\ell-1}\tau + 2^\ell\mathcal{O}(2^{\ell+1})) = \widetilde{\mathcal{O}}(d^2 + d\tau) \text{ opérations binaires ;}
\end{aligned}$$

Finalement, la dernière complexité du calcul est :

$$\begin{aligned}
C_A &= C_A^{(1)} + C_A^{(2)} = \widetilde{\mathcal{O}}(d) && \text{opérations arithmétiques,} \\
C_B &= C_B^{(1)} + C_B^{(2)} = \widetilde{\mathcal{O}}(d^2 + d\tau) && \text{opérations binaires.}
\end{aligned}$$

□

Remarque 2.38. Considérons la base des polynômes de Chebyshev classique $\overline{\mathcal{T}} \overset{\text{déf}}{=} \{\mathbf{T}_n, n \in \mathbb{N}\}$, notre méthode permet de faire le changement entre $\mathcal{X}$ et $\overline{\mathcal{T}}$ en même complexité :

1. Etant donné $f = \sum_{j=1}^{d} a_j x^j$, $\tau(f) = \tau$:
   (i) En écrivant d'abord $2^d f(x) = \overline{f}(2x)$ gratuitement, on a $\tau' \overset{\text{déf}}{=} \tau(\overline{f}) = \tau + \mathcal{O}(d)$;
   (ii) Ensuite on applique l'algorithme 2.31 avec l'entrée $[\mathcal{X}]\overline{f}$ pour obtenir $[\mathcal{T}]\overline{f}$ en $C_B = \widetilde{\mathcal{O}}(d^2 + d\tau') = \widetilde{\mathcal{O}}(d^2 + d\tau)$ opérations binaires, cela nous donne :

$$2^d f(x) = \overline{f}(2x) = \overline{f}_0 + \sum_{j=1}^{d} \overline{f}_j T_j(2x)$$

(iii) Finalement, grâce à la relation $2\mathbf{T}_k(x) = T_k(2x)$ on obtient $[\overline{\mathcal{T}}]f$ par :

$$f(x) = \frac{\overline{f}_0}{2^d} + \sum_{j=1}^{d} \frac{\overline{f}_j}{2^{d-1}} \mathbf{T}_j(x).$$

2. Inversement, avec une forme $[\overline{\mathcal{T}}]f = f_0 + \sum_{j=1}^{d} f_j \mathbf{T}_j$ ($\tau_{\overline{\mathcal{T}}}(f) = \tau$) donnée, on peut faire successivement :

   (i) Réécrire

$$2f(x) = 2f_0 + \sum_{j=1}^{d} f_j \cdot 2\mathbf{T}_j(x) = 2f_0 + \sum_{j=1}^{d} f_j T_j(2x);$$

   (ii) Appliquer l'algorithme 2.36 pour la forme $2f_0 + \sum_{j=1}^{d} f_j T_j$ dont les coefficients sont de taille binaire au plus $\tau+1$, on utilise $C_B = \widetilde{\mathcal{O}}(d^2+d(\tau+1)) = \widetilde{\mathcal{O}}(d^2+d\tau)$ opérations binaires pour obtenir :

$$2f(x) = \sum_{j=0}^{d} a'_j (2x)^j;$$

   (iii) Enfin, renvoyer gratuitement :

$$f(x) = \sum_{j=0}^{d} a'_j \cdot 2^{j-1} x^j.$$

Conclusion du chapitre 2

En utilisant l'application $\mathcal{D}$ et à l'aide de la stratégie *Diviser pour Régner*, nous avons proposé des algorithmes et analysé leur complexité :

1. T_n, U_n peuvent être calculés dans la base des monômes en complexité $C_A = \widetilde{\mathcal{O}}(n)$, $C_B = \widetilde{\mathcal{O}}(n^2)$, (algorithme 2.10) ;
2. Les opérations arithmétiques de $\mathbb{Z}[x]$ (multiplication, division euclidienne, test de divisibilité, calcul de pgcd, calcul des coefficients de Bézout) dans la base des polynômes unitaires de Chebyshev peuvent être exécutées aussi rapidement que dans la base des monômes (algorithmes 2.15, 2.19) ;
3. Le produit de plusieurs polynômes et la composition dans la base des monômes peuvent être calculées assez rapidement (propositions 2.28, 2.30) ;
4. Avec l'entrée de taille (d, τ), le changement de base entre $\mathcal{X}$ et $\mathcal{T}$ peut être fait en complexité $C_A = \widetilde{\mathcal{O}}(d), C_B = \widetilde{\mathcal{O}}(d^2 + d\tau)$ (algorithmes 2.31, 2.36).

Chapitre 3

Calcul des polynômes minimaux

Sommaire

3.1 Calcul du polynôme cyclotomique 60

3.1.1 Méthode d'Arnold et Monagan 61

3.1.2 Calcul de $\Phi_n(x)$ avec une complexité arithmétique quasi-linaire . 62

3.2 Calcul du polynôme minimal de $2\cos\frac{\pi}{n}$ 63

3.2.1 Idée principale et algorithme 63

3.2.2 Analyse de complexité 65

Résumé : Nous étudions le calcul du polynôme cyclotomique Φ_n et du polynôme minimal M_n de $2\cos\frac{\pi}{n}$. Nous démontrons que Φ_n peut être obtenu en $\widetilde{\mathcal{O}}(n)$ opérations arithmétiques, par rapport à $\mathcal{O}(2^{\omega(n)} \cdot n)$ évaluée par Arnold et Monagan en 2010.

Pour M_n, la représentation dans la base des polynômes unitaires de Chebyshev peut être calculée en complexité binaire $\widetilde{\mathcal{O}}(n^2)$.

Nous en déduisons le calcul de Φ_n en $\widetilde{\mathcal{O}}(n^2)$ opérations binaires.

Dénotons $\mathcal{P}$ l'ensemble de nombres premiers. Soit n un entier positif, nous considérons toujours sa factorisation standard :

$$n = 2^h \cdot p_1^{\alpha_1} p_2^{\alpha_2} \dots p_k^{\alpha_k}, \quad (2 < p_1 < p_2 < \dots < p_k \in \mathcal{P}, h \geq 0, \alpha_j > 0),$$

Le nombre $2^{\min\{h,1\}} p_1 \dots p_k$ s'appelle *le radical* de n. On dénote aussi $n_0 = p_1 \dots p_k$ le radical de la partie impaire de n.

k est le nombre de diviseurs premiers impairs de n. D'autre part, le nombre de tous les diviseurs premiers de n est expliqué par une fonction arithmétique connue dénotée par $\omega(n)$ [1].

Le nombre de diviseurs de n est dénoté $d(n)$. Il est classique que $d(n) = (h+1)(\alpha_1 + 1) \dots (\alpha_k + 1)$; Évidemment $d(n) < n$.

Calculer la valeur de la fonction d'Euler

Nous présentons d'abord quelques résultats concernant la factorisation de nombres entiers et le calcul de la fonction d'Euler qui sont très proches du calcul de $\Phi_n(x)$.

Le coût de la factorisation d'un nombre entier est un problème classique du Calcul Numérique. On a :

1. alors on a : $\omega(n) = k + \frac{1+(-1)^n}{2}$.

Lemme 3.1. *[Mil76, Théo. 3], [GG13, Coro. 19.4, Table 19.1] Soit n un entier positif, alors on peut factoriser n et calculer $\varphi(n)$ en $\mathcal{O}(n^{1/4})$ opérations arithmétiques, $\widetilde{\mathcal{O}}(n^{1/4})$ opérations binaires.*

Remarque 3.2. Pour tout diviseur d de n, vu la remarque 1.10, $\varphi(d) \mid \varphi(n) < n$, alors

$$\tau(\varphi(d)) < \tau(n) \leq \log_2 n.$$

On en déduit

Corollaire 3.3. *Soit $n = p_1^{\alpha_1} \dots p_k^{\alpha_k}$ alors on peut calculer tous les $\varphi(d)$, et les $\mu(d)$, $d \mid n$ en $\mathcal{O}(n)$ opérations arithmétiques, $\widetilde{\mathcal{O}}(n)$ opérations binaires.*

Démonstration. Partons de la factorisation $n = p_1^{\alpha_1} \dots p_k^{\alpha_k}$ que nous obtenons en $\widetilde{\mathcal{O}}(n^{1/4}$ opérations binaires. Pour chaque p_i, nous avons $\varphi(p_i)$ en $\mathcal{O}(\log_2 n)$ opérations. Calculons tous les diviseurs $d = p_1^{\beta_1} \dots p_k^{\beta_k}$ de n dans un tableau, en gardant l'ordre lexicographique sur les exposants. Leur nombre est inférieur à n. Chaque nouveau diviseur est obtenu en multipliant le précédant par un p_i, ce qui coûte $\mathcal{O}(\log_2 n)$ opérations binaires. Pour ce diviseur, nous obtenons $\varphi(d)$, soit en multipliant par $\varphi(p_i) = p_i - 1$, soit en multipliant par p_i, en $\mathcal{O}(\log_2 n)$ opérations binaires. À la fin, nous obtenons tous les $(d, \varphi(d), \mu(d)$ en $\widetilde{\mathcal{O}}(n \log_2 n) = \widetilde{\mathcal{O}}(n)$ opérations binaires. Le nombre d'opérations arithmétiques utilisées est donc $C_A = d(n) < n$. Vu le coût de la multiplication de nombres entiers énoncé dans le corollaire 2.7, le calcul utilise moins que $C_B = d(n)\widetilde{\mathcal{O}}(\log_2 n) = \widetilde{\mathcal{O}}(n)$ opérations binaires. □

3.1 Calcul du polynôme cyclotomique

Il y a eu plusieurs travaux sur la description de $\Phi_n(x)$ en détail où les auteurs avaient donné des évaluations théoriques des coefficients du Φ_n.

En 1883, Migotti a prouvé que $\|\Phi_n\|_\infty = 1$ pour tout n tel que $\omega(n) = 2$ [Mig83]. Mais pour le cas où $\omega(n) = 3$, l'auteur a donné un simple exemple Φ_{105} qui admet -2 comme coefficient ($[x^7]\Phi_{105} = -2$). Ensuit, Bang a montré en 1895 que $\|\Phi_n\|_\infty \leq p - 1$ où p est le diviseur premier le plus petit de n [Ban95]. En 1931, Schur a prouvé un théorème important :

Proposition 3.4. *Il existe des polynômes cyclotomiques de n'importe quelle taille de coefficients [Sch, 1931].*

En allant plus loin que Schur, Lehmer (énonciation 3.1a, [Leh36, 1936]) et Erdos (énonciation 3.1b, [Erd46, 1946]) ont montré qu'on ne peut pas borner les coefficients de Φ_n par une fonction polynomiale de n :

$$\exists c > 0 : |\{n : \|\Phi_n\|_\infty > cn^{\frac{1}{3}}\}| = +\infty, \tag{3.1a}$$

$$\forall k \in \mathbb{N}, |\{n : \|\Phi_n\|_\infty > n^k\}| = +\infty. \tag{3.1b}$$

Quelques travaux sur les coefficients de Φ_n peuvent listés : [Bos95, LL96, Bac03, GT91, Kos00].

L'essai le plus récent a été publié par Arnold et Monagan en 2010 [AM10].

3.1.1 Méthode d'Arnold et Monagan

Rappelons quelques propriétés de Φ_n :

Lemme 3.5. *[AM10, Lem. 2] Soit n un entier positif alors :*

$$\Phi_{2n}(x) = \begin{cases} \Phi_n(-x) & \text{si } n \text{ est impair} \\ \Phi_n(x^2) & \text{sinon.} \end{cases}$$

Démonstration. On a $x^{2n} - 1 = (x^n - 1)(x^n + 1)$.

∘ Si n est impair, considérons ξ une racine primitive $(2n)$-ième de 1 : $\mathrm{ord}(\xi) = 2n$, $\xi^n - 1 \neq 0 \Rightarrow 1 - (-\xi)^n = 0$ donc $-\xi$ est une racine n-ième de 1.
En fait, si $\mathrm{ord}(-\xi) = k|n$, $k < n$ alors $\xi^{2k} = 1$, cela contredit $\mathrm{ord}(\xi) = 2n$.
Ainsi $\mathrm{ord}(-\xi) = n$, $-\xi$ est racine primitive n-ième de 1, autrement dit ξ est racine du polynôme $\Phi_n(-x)$. Parce que $\deg \Phi_{2n} = \deg \Phi_n = \varphi(n)$, les deux sont irréductibles donc ils doivent être identiques : $\Phi_{2n}(x) = \Phi_n(-x)$.

∘ Si n est pair, $n = 2^h n'$, $h \geq 1$, n' impair donc $\varphi(n) = \varphi(2^k)\varphi(n') = 2^{k-1}\varphi(n')$. De même, $\varphi(2n) = 2^k\varphi(n') \Rightarrow \deg \Phi_{2n} = 2 \deg \Phi_n$. D'autre part pour toute racine ξ de Φ_{2n}, $(\xi^2)^n = 1$, $\mathrm{ord}(\xi^2) = \frac{2n}{(2,2n)} = n$, alors ξ^2 est racine de Φ_n. Ainsi $\Phi_{2n} \mid \Phi_n(x^2)$, les deux sont unitaires de même degré alors ils sont identiques : $\Phi_{2n}(x) = \Phi_n(x^2)$. □

La partie complémentaire de Φ_n dans $x^n - 1$ est aussi un polynôme unitaire de $\mathbb{Z}[x]$ (voir la preuve du lemme 1.12). Comme cela est fait dans [AM10], on le dénotera $\Psi_n(x)$:

$$\Psi_n(x) = \frac{x^n - 1}{\Phi_n(x)} = \prod_{\substack{1 \leq d \leq n \\ (d,n)>1}} (x - \xi_d).$$

Lemme 3.6. *[AM10, Lemme 1] Soit p, q deux nombres premiers tels que $p \nmid n, q \mid n$ alors :*

$$\Phi_{np}(x) = \frac{\Phi_n(x^p)}{\Phi_n(x)}; \quad \Phi_{nq}(x) = \Phi_n(x^q); \tag{3.2a}$$

$$\Psi_{np}(x) = \Psi_n(x^p) \cdot \Phi_n(x); \quad \Psi_{nq}(x) = \Psi_n(x^q). \tag{3.2b}$$

Démonstration. Parce que $\Phi_n(x) \cdot \Psi_n(x) = x^n - 1$, (3.2a)⇔(3.2b). Il faut alors démontrer deux identités dans (3.2a).

Nous aurons besoin d'une propriété venant de la remarque 1.8 : Si ξ est une racine primitive n-ième, alors

$$\mathrm{ord}(\xi^k) = \tfrac{n}{(n,k)}.$$

1. Avec p qui n'est pas diviseur de n :

Prenons ξ un zéro arbitraire de Φ_n qui est racine primitive n-ième de 1 : Car $\xi^n = 1$, $(\xi^p)^n = 1$. Mais $\mathrm{ord}(\xi^p) = \frac{n}{(n,p)} = n$ donc ξ^p est zéro de Φ_n, c'est-à-dire que ξ est zéro de $\Phi_n(x^p)$. On a donc

$$\Phi_n \mid \Phi_n(x^p); \tag{3.3}$$

Un zéro arbitraire ξ' de Φ_{np} est racine primitive (np)-ième de 1. Parce que $(\xi')^{np} = 1$ alors $(\xi'^p)^n = 1$. En plus $\mathrm{ord}(\xi'^p) = \frac{np}{(np,p)} = n$ donc ξ'^p est zéro de Φ_n, ce qui implique que ξ' est racine de $\Phi_n(x^p)$, par la suite :

$$\Phi_{np} \mid \Phi_n(x^p); \tag{3.4}$$

En fin, la somme de degrés de Φ_n et Φ_{np} est :

$$\deg \Phi_{np} + \deg \Phi_n = \varphi(np) + \varphi(n) = p\varphi(n) = \deg \Phi_n(x^p); \tag{3.5}$$

Les trois polynômes $\Phi_n, \Phi_{np}, \Phi_n(x^p)$ sont tous unitaires, donc les relations (3.3), (3.4) et (3.5) indiquent que $\Phi_{np}\Phi_n = \Phi_n(x^p)$.

2. Avec q l'un des diviseurs premiers de n, prenons un zéro λ arbitraire de Φ_{nq} : $\lambda^{nq} = (\lambda^q)^n = 1 \Rightarrow \lambda^q$ est racine n-ième de 1. En plus $\mathrm{ord}(\lambda^q) = \frac{nq}{(nq,q)} = n$, donc λ^q est zéro de Φ_n, ou λ est zéro de $\Phi_n(x^q)$. On obtient

$$\Phi_{nq} \mid \Phi_n(x^q). \tag{3.6}$$

Posons $n = q^k n'$, $k \geq 1$, $(n', q) = 1$ alors :

$$\begin{aligned}\varphi(nq) &= \varphi(n'q^{k+1}) = \varphi(n')\varphi(q^{k+1}) \\ &= \varphi(n')q^{k+1}(1 - \tfrac{1}{q}) = q\varphi(n')\varphi(q^k) = q\varphi(n) = \deg \Phi_n(x^q),\end{aligned} \tag{3.7}$$

grâce à la relation (3.6) et l'identité (3.7), $\Phi_{nq}(x) = \Phi_n(x^q)$.

□

Basés sur le lemme 3.6, Arnold et Monagan ont proposé quatre algorithmes. Ils ont aussi démontré la complexité arithmétique nécessaire pour obtenir Φ_n dans la base des monômes :

Proposition 3.7 ([AM10]). *Supposons que $n = p_1 \dots p_k$ avec $p_1, \dots, p_k$ premiers distincts, alors on peut calculer Φ_n par $\mathcal{O}(2^k n)$ opérations arithmétiques de $\mathbb{Z}$.*

Remarque 3.8. Dans cette complexité, $k = \omega(n)$. Sa valeur moyenne est assez petite selon [HW08, Sec. 22.11] :

$$\omega(n) \sim \log_2 \log_2 n.$$

Pourtant, théoriquement, le facteur 2^k ne peut pas être enlevé en utilisant $\widetilde{\mathcal{O}}$ à la place de $\mathcal{O}$.

Si l'on ne s'intéresse qu'à la complexité arithmétique, effectivement il existe une autre méthode décrite ci-dessous utilisant $\widetilde{\mathcal{O}}(n)$ opérations arithmétiques de $\mathbb{Z}$ permettant d'obtenir Φ_n.

3.1.2 Calcul de $\Phi_n(x)$ avec une complexité arithmétique quasi-linaire

Supposons que P soit unitaire de degré d, les $S_i(P), i = 1, \dots, d$ soient calculées. Si l'on applique directement les formules de Newton (théorème 1.16) :

$$[x^{d-i}]P = -\tfrac{1}{i}\Big[S_1(P)A_{i-1} + \dots + S_{i-1}(P)A_1 + S_i(P)\Big], i = 1, \dots, d,$$

alors, il est possible de calculer tous les $A_i, i = 1, \dots, d$ successivement en utilisant

$$\sum_{j=1}^{d-1} 2(j-1) = \mathcal{O}(d^2)$$

opérations arithmétiques. On va utiliser un autre outil afin d'approcher la complexité quasi-linaire.

On sait que Φ_n est réversible alors $\Phi_n = \mathrm{REV}(\Phi_n)$. Par le lemme 1.17, en posant $N = \deg \Phi_n$, on peut écrire :

$$\Phi_n(x) = \exp\Big(S_1(\Phi_n)x + \tfrac{1}{2}S_2(\Phi_n)x^2 + \dots + \tfrac{1}{N}S_N(\Phi_n)\Big) \pmod{x}^{N+1}.$$

Lemme 3.9. *[Bre76] Soit $f = \sum_{j\geq 0} f_j x^j$ une série formelle. Si les d premiers coefficients de f sont donnés, on peut calculer les d premiers coefficients de $\exp(f)$ en $\mathcal{O}(\mathcal{M}(d)\log_2 d) = \widetilde{\mathcal{O}}(d)$ opérations arithmétiques.*

Basé sur cette idée, dans [BFSS05], les auteurs ont construit un algorithme qui réalise la transformation $\{S_i(P)\} \mapsto \{A_i\}$ permettant d'énoncer :

Lemme 3.10 (`TransNewton`)**.** *[Pan00, Théo. 10.1],[BFSS05, Coro. 1] Soit P un polynôme unitaire de degré d dans $K[x]$ donnés par ses d premières sommes de Newton, où la caractéristique de K soit 0 ou plus grand que d. Alors, on peut calculer tous les coefficients de P en $\mathcal{O}(\mathcal{M}(d)) = \widetilde{\mathcal{O}}(d)$ opérations arithmétiques de K.*

En appliquant les lemmes 3.9 et 3.10, nous avons :

Proposition 3.11. *Soit n un entier positif, alors le polynôme cyclotomique Φ_n peut être calculé en utilisant $\widetilde{\mathcal{O}}(n)$ opérations arithmétiques de $\mathbb{Z}$.*

Démonstration. D'après le corollaire 3.3, nous pouvons obtenir d'abord tous les $\varphi(d)$ et $\mu(d)$, $d \mid n$ en $C_A^{(1)} = \widetilde{\mathcal{O}}(n)$ opérations arithmétiques de $\mathbb{Z}$;

Ensuite, nous calculons tous les $S_m(\Phi_n)$, $m = 1, \ldots, \varphi(n)$. Pour ce faire, exécutons l'identité (1.20)

$$S_m(\Phi_n) = \mu\Big(\tfrac{n}{(n,m)}\Big) \cdot \frac{\varphi(n)}{\varphi(\frac{n}{(n,m)})} \text{ (théorème de Hölder 1.18)}$$

pour $\varphi(n)$ fois, ce qui utilise $C_A^{(2)} = 2\varphi(n) = \mathcal{O}(n)$ opérations arithmétiques ;

Finalement, avec les $S_m(\Phi_n)$ calculés, nous appliquons la boîte noire `TransNewton` pour trouver Φ_n. Cette étape utilise $C_A^{(3)} = \widetilde{\mathcal{O}}(n)$ opérations arithmétiques, vu la proposition 3.10.

Évidemment $C_A^{(1)} + C_A^{(2)} + C_A^{(3)} = \widetilde{\mathcal{O}}(n)$ ce qui termine la preuve. □

Nous ne mentionnons pas la complexité binaire de cette méthode parce qu'elle demande d'entrer profondément à la boîte noire `TransNewton`, ce qui serait embarrassant.

3.2 Calcul du polynôme minimal de $2\cos\frac{\pi}{n}$

En 2006, A. Valibouze [Val06] a proposé une méthode pour le calcul de M_n basé sur le calcul de résultants mais aucune complexité binaire n'a été confirmée.

Ici nous proposons d'adapter la méthode d'Anold et Monagan avec les opérations dans la base des polynômes unitaires de Chebyshev.

3.2.1 Idée principale et algorithme

Nous utilisons une version modifiée du lemme 3.6 concernant Φ_n :

Lemme 3.12. *Soit $p = 2q + 1$ un nombre premier, n un entier positif. On a :*

$$M_p = (-1)^q + (-1)^{q-1}T_1 + \ldots + T_q; \tag{3.8a}$$

$$M_{np} = \begin{cases} M_n(T_p) & \text{si } p|n \\ \frac{M_n(T_p)}{M_n} & \text{sinon} \end{cases}. \tag{3.8b}$$

Démonstration. 1. Appliquons la formule d'Euler $\exp(\imath x) = \cos x + \imath \sin x$ et le théorème d'Hölder

$$S_m(\Phi_n) = \sum_{(k,n)=1} \exp\Big(m \cdot \tfrac{2k\pi\imath}{n}\Big) = \mu\Big(\frac{n}{(n,m)}\Big)\frac{\varphi(n)}{\varphi\Big(\frac{n}{(n,m)}\Big)}$$

pour $n = 2p$, $m = 1$ on obtient :

$$\sum_{(k,2p)=1} \Big(\cos \tfrac{k\pi}{p} + \imath \sin \tfrac{k\pi}{p}\Big) = \mu(2p) = 1,$$

à condition que $p = 2q + 1 \in \mathcal{P}$, ce qui nous donne :

$$\sum_{i=1}^{q} (-1)^i \cos \tfrac{i\pi}{p} = -\tfrac{1}{2},$$

on en déduit

$$\Big((-1)^q + (-1)^{q-1}T_1 + (-1)^{q-2}T_2 + \ldots + T_q\Big)(2\cos \tfrac{\pi}{p}) = 0,$$

$(-1)^q + (-1)^{q-1}T_1 + (-1)^{q-2}T_2 + \ldots + T_q$ étant polynôme unitaire de degré $q = \deg M_p$ qui s'annule en $2\cos\frac{\pi}{p}$, doit être identique M_p.
2. Rappelons que $M_n = \prod_{\substack{1\le k\le n\\(k,2n)=1}} (x - 2\cos\frac{k\pi}{n})$ (corollaire 1.28). De plus, les trois polynômes M_n, M_{np}, $M_n(T_p)$ sont unitaires.

On a d'abord : $M_n(T_p)(2\cos\frac{\pi}{np}) = M_n(2\cos\frac{\pi}{n}) = 0$, donc $M_{np}|M_n(T_p)$;
∘ Si $p|n$ alors :

$$\begin{aligned}\deg M_n(T_p) &= \tfrac{\varphi(n)}{2}\cdot p = \tfrac{\varphi(np)}{2} = \tfrac{\varphi(2np)}{2})\\ &= \deg M_{np}\end{aligned}$$

donc $M_{np} = M_n(T_p)$;
∘ Si $(p,n) = 1$, $2\cos\frac{p\pi}{n}$ est racine de M_n, $M_n(T_p)(2\cos\frac{\pi}{n}) = 0$, ceci implique que $M_{np}|M_n(T_p)$. D'autre part :

$$\begin{aligned}\deg M_n(T_p) &= \tfrac{\varphi(n).p}{2} = \tfrac{\varphi(n)\varphi(p)}{2} + \tfrac{\varphi(n)}{2} = \tfrac{\varphi(2np)}{2} + \tfrac{\varphi(2n)}{2} =\\ &= \deg M_{np} + \deg M_n.\end{aligned}$$

Évidemment M_n et M_{np} sont premiers entre-eux qui sont tous les deux facteurs irréductibles de $M_n(T_p)$, donc on a $M_n(T_p) = M_n M_{np}$. □

Dans le chapitre 1, le lemme 1.29 confirme que M_n équivaut à $T_{2^{k-1}}$ si $n_0 = 1$, à $M_{n_0} \circ T_{2^k}$ si $n_0 > 1$. En combinant ce résultat avec la proposition 3.12 nous avons :

Corollaire 3.13. *Soit n un entier positif, alors $M_n = 1$ si $n = 1$; $M_n = T_{\frac{n}{2}}$ si n est une puissance de 2. Pour les cas restants, $M_n = M_{n_0}(T_{n/n_0})$ où n_0 est le radical de la partie impaire de n.*

Nous construisons donc l'algorithme 3.14 où toutes les opérations arithmétiques de $\mathbb{Z}[x]$ se sont faites dans la base des polynômes unitaires de Chebyshev :

Algorithme 3.14 : Calculer M_n dans la base des polynômes unitaires de Chebyshev

Entrées : $n \in \mathbb{N}$
Sorties : La forme de Chebyshev de M_n, le polynôme minimal de $2\cos\frac{\pi}{n}$

```
1  begin
2      Factoriser n = 2^h p_1^{α_1} ... p_k^{α_k};
3      n_0 ← p_1 ... p_k;
4      si k = 0 alors
5          si h = 0 alors
6              retourner 1
7          sinon
8              retourner T_{n/2}
9      m ← (−1)^{(p_1−1)/2} + (−1)^{(p_1−3)/2} T_1 + ... − T_{(p_1−3)/2} + T_{(p_1−1)/2};
10     pour i ← 2 a k faire
11         m ← [T](m(T_{p_i}) / m)
12     retourner m(T_{n/n_0})
```

Preuve de correction. Grâce au lemme 3.12 et au corollaire 3.13, l'algorithme est bien correct pour calculer $[\mathcal{T}]M_n$. □

3.2.2 Analyse de complexité

- À la ligne du départ, grâce au lemme 3.12, $[\mathcal{T}]M_p, p \in \mathcal{P}$ ne demande aucun calcul : on écrit simplement ses coefficients qui sont tous ± 1 sur $\frac{p-1}{2}$ places convenables en $\widetilde{\mathcal{O}}(p)$ opérations binaires.

- Les compositions nécessaires sont aussi faciles à obtenir, grâce au fait que :

$$\text{Si } f(x) = f_0 + \sum_{j=1}^{d} f_j T_j(x) \text{ alors } f(T_k(x)) = f_0 + \sum_{j=1}^{d} f_j T_{kj}(x), \tag{3.9}$$

tous les coefficients sont gardés, c'est-à-dire qu'on doit justement copier, trouver une place à coller. Ces copies utilisent au plus $d\tau_{\mathcal{T}}(f)$ opérations binaires.

- Avec le corollaire 2.26, on sait faire rapidement la division exacte dans la base des polynômes unitaires de Chebyshev : Étant donnés $[\mathcal{T}]f, [\mathcal{T}] \in \mathbb{Z}[x]$, $\deg g \leq \deg f = d$, $\max(\tau_{\mathcal{T}}(f), \tau_{\mathcal{T}}(g)) = \tau$, $g \mid f$ alors le quotient $[\mathcal{T}](\frac{f}{g})$ peut être calculé en $\widetilde{\mathcal{O}}(d)$ opérations arithmétiques, $\widetilde{\mathcal{O}}(d^2 + d\tau)$ opérations binaires.

Ce qui manquant est la taille des coefficients de M_n dans la base des polynômes unitaires de Chebyshev. En fait on peut aller un peu plus loin en évaluant les deux tailles $\tau(M_n)$ et $\tau_{\mathcal{T}}(M_n)$:

Lemme 3.15. *Soit n un entier positif, alors on a : $\tau(M_n) = \mathcal{O}(n)$, $\tau_{\mathcal{T}}(M_n) = \mathcal{O}(n_0)$.*

Démonstration. 1. Par la définition $U_n(2\cos t) = \frac{\sin nt}{\sin t}$, il est clair que $U_n(2\cos\frac{\pi}{n}) = 0$ ce qui implique que $M_n | U_n$.

Mais $\deg U_n = n - 1$ et d'après l'évaluation (2.34), $\tau(U_n) \leq n - 1$. En appliquant la borne de Mignotte venant du lemme 2.24, on a :

$$\tau(M_n) \leq \deg M_n + \tfrac{1}{2}\log_2 n + \tau(U_n) \leq \tfrac{\varphi(2n)}{2} + \tfrac{1}{2}\log_2 n + n - 1 = \mathcal{O}(n).$$

2. Appliquons de nouveau la borne de Mignotte pour $\mathcal{D}(M_{n_0}) = \Phi_{2n_0} \mid x^{2n_0} - 1$:

$$\begin{aligned}\tau_{\mathcal{T}}(M_n) = \tau_{\mathcal{T}}(M_{n_0}) = \tau(\Phi_{2n_0}) &\leq \deg \Phi_{2n_0} + \tfrac{1}{2}\log_2(2n_0+1) + \tau(x^{2n_0}-1) \\ &\leq \varphi(2n_0) + \log_2 n_0 + 2 = \mathcal{O}(n_0).\end{aligned}$$

□

Proposition 3.16. *Soit n un entier positif, n_0 le radical de sa partie impaire. Alors, on peut calculer la forme de Chebyshev de M_n en $\widetilde{\mathcal{O}}(n_0)$ opérations arithmétiques, $\widetilde{\mathcal{O}}(n_0^2)$ opérations binaires.*

Démonstration. Afin de calculer $[\mathcal{T}]M_n$, nous utilisons l'algorithme 3.14.

Posons $q_j = p_1 \dots p_j$, il y a au total $k-1$ divisions exactes en base des polynômes unitaires de Chebyshev à exécuter :

$$M_{q_j} = [\mathcal{T}]\Big(\tfrac{M_{q_{j-1}}(T_{p_j})}{M_{q_{j-1}}}\Big), j = 2, \dots, k.$$

1. On a $\deg M_{q_{j-1}}(T_{p_j}) < q_{j-1}p_j = q_j$, $\deg M_{q_{j-1}} < q_{j-1}$. Vu le corollaire 2.26, cette division exacte utilise $C_A^{(j)} = \widetilde{\mathcal{O}}(q_j)$ opérations arithmétiques.

Parce que $3 \leq p_1 < \dots < p_k$ alors $k < \log_3(n_0)$, la complexité arithmétique de l'algorithme est donc :

$$C_A = \sum_{j=2}^{k} C_A^{(j)} = \sum_{j=2}^{k} \widetilde{\mathcal{O}}(q_j) \leq \widetilde{\mathcal{O}}(kn_0) = \widetilde{\mathcal{O}}(n_0).$$

2. Maintenant évaluons la complexité binaire : Le lemme 3.15 nous permet d'obtenir :

$$\tau_{\mathcal{T}}(M_{q_{j-1}}(T_{p_j})) = \tau_{\mathcal{T}}(M_{q_{j-1}}) = \mathcal{O}(q_{j-1}).$$

Selon le corollaire 2.26, le quotient $[\mathcal{T}]\Big(\tfrac{M_{q_{j-1}}(T_{p_j})}{M_{q_{j-1}}}\Big)$ utilise $C_B^{(j)} = \widetilde{\mathcal{O}}(q_j^2 + q_j q_{j-1}) = \widetilde{\mathcal{O}}(n_0^2)$ opérations binaires à trouver.

Enfin la complexité binaire de l'algorithme est synthétisée par :

$$C_B = \sum_{j=1}^{k} C_B^{(j)} = \sum_{j=2}^{k} \widetilde{\mathcal{O}}(n_0^2) = \widetilde{\mathcal{O}}(kn_0^2) = \widetilde{\mathcal{O}}(n_0^2).$$

□

Corollaire 3.17. *Pour tout $n > 3$, les deux polynômes Φ_n et M_n peuvent être calculés (en base de monômes ou des polynômes unitaires de Chebyshev) en $\widetilde{\mathcal{O}}(n)$ opérations arithmétiques, $\widetilde{\mathcal{O}}(n^2)$ opérations binaires.*

Démonstration. Grâce au lemme 1.26, $\Phi_{2n} = \mathcal{D}(M_n)$, de plus :

- $\Phi_{2n} = \begin{cases} \Phi_n(-x) & \text{si}(2,n) = 1 \\ \Phi_n(x^2) & \text{sinon} \end{cases}$ alors on peut reconstruire $[\mathcal{X}]\Phi_n$ à partir de $[\mathcal{X}]\Phi_{2n}$ sans calcul, d'autre part $\tau(\Phi_{2n}) = \tau(\Phi_n)$;
- On sait la complexité du changement de base
  - pour calculer la forme de Chebyshev de f, on utilise $C_A = \widetilde{\mathcal{O}}(\deg f)$, $C_B = \widetilde{\mathcal{O}}(\deg^2 f + \deg f \tau(f))$, vu la proposition 2.33 ;

— réciproquement, pour calculer la représentation dans la base des monômes d'une forme de Chebyshev $[\mathcal{T}]f$, on utilise $C_A = \widetilde{\mathcal{O}}(\deg f)$, $C_B = \widetilde{\mathcal{O}}(\deg^2 f + \deg f \tau_{\mathcal{T}}(f))$, selon la proposition 2.37 ;

∘ $\tau(M_n) = \widetilde{\mathcal{O}}(n)$, $\tau_{\mathcal{T}}(M_n) = \tau(\Phi_{2n}) = \tau(\Phi_n) = \mathcal{O}(n_0)$ (lemme 3.15) et $\tau_{\mathcal{T}}(\Phi_n) \leq \tau(\Phi_n) + \deg \Phi_n + 1 = \mathcal{O}(n)$ (proposition 2.33).

Ce qui induit le corollaire. □

À titre de comparaison, nous avons essayé d'évaluer les autres méthodes pour calculer M_n :

- Soit on utilise les sommes de Ramanujan et les formules de Newton pour calculer tous les coefficients de $[\mathcal{X}]M_n$ en complexité $C_A = \mathcal{O}(n^2)$, $C_B = \widetilde{\mathcal{O}}(n^3)$;

- Soit on utilise l'identité (1.34) : $V_n(x) = \prod_{d|2n+1} M_d(x)$ où M_n est le dernier facteur qui peut être trouvé récursivement : $C_A = \widetilde{\mathcal{O}}(n + n_0 3^k)$, $C_B = \widetilde{\mathcal{O}}(n^2 + n_0^2(5^k + n))$;

- Soit, en répétant la méthode d'Arnold - Monagan avec les opérations en base de monômes, on calcule successivement sur la suite $p_1, p_1p_2, \ldots, p_1 \ldots p_k$: $C_A = \widetilde{\mathcal{O}}(n)$, $C_B = \widetilde{\mathcal{O}}(n^2 + nn_0^2)$.

Remarque 3.18. Il semble difficile de faire mieux que $\mathcal{O}(n)$ en majorant $\tau(M_n)$, $\tau_{\mathcal{T}}(M_n)$ parce que $\tau_{\mathcal{T}}(M_n) = \tau(\Phi_{2n})$ mais d'après (3.1b) :

$$|\{n : \tau(\Phi_n) > k \log_2 n\}| = +\infty.$$

Cependant les deux familles des coefficients sont très différentes, voyons l'exemple 3.19 suivant.

Exemple 3.19. $n = 2^2 \cdot 3^2 \cdot 5 \cdot 7$, $n_0 = 3 \cdot 5 \cdot 7 = 105$

1. $n = 1260 = 2^2 \cdot 3^2 \cdot 5 \cdot 7$, $n_0 = 3 \cdot 5 \cdot 7$;
2. (a) $[\mathcal{T}]M_3 = T_1 - 1$;
   (b) $[\mathcal{T}]M_{15} = \frac{M_3(T_5)}{M_3} = \frac{T_5 - 1}{T_1 - 1} = T_4 + T_3 - T_1 - 1$;
   (c) $[\mathcal{T}]M_{105} = \frac{M_{15}(T_7)}{M_{15}} = \frac{T_{28} + T_{21} - T_7 - 1}{T_4 + T_3 - T_1 - 1} = T_{24} - T_{23} + T_{22} + T_{19} - T_{18} + 2T_{17} - T_{16} + T_{15} + T_{12} - T_{11} + T_{10} - T_9 + T_8 - T_7 - T_4 - T_2 - 1$;
3. $[\mathcal{T}]M_{1260} = M_{105}(T_{12}) = T_{288} - T_{276} + T_{264} + T_{228} - T_{216} + 2T_{204} - T_{192} + T_{180} + T_{144} - T_{132} + T_{120} - T_{108} + T_{96} - T_{84} - T_{48} - T_{24} - 1$;

Pour donner une comparaison numérique entre $[\mathcal{T}]M_n$ et $[\mathcal{X}]M_n$, nous avons calculé la taille binaire des coefficients, le nombre de bits utilisés pour écrire M_{1260}, M_{936}, M_{8640} (fait en Maple 18, voir la figure A.4, l'annexe A.1) :

∘ Pour $n = 1260$ on a

$$\begin{aligned} M_{1260} = M_{105}(T_{12}) &= T_{288} - T_{276} + T_{264} + T_{228} - T_{216} + 2T_{204} \\ &\quad - T_{192} + T_{180} + T_{144} - T_{132} + T_{120} - T_{108} + T_{96} - T_{84} - T_{48} - T_{24} - 1. \end{aligned}$$

$\tau_{\mathcal{T}}(M_{1260}) = 2$ pendant que $\tau(M_{1260}) = 197$; pour présenter tous ses coefficients, on utilise 20475 bits.

∘ Pour $n = 936 = 2^3 \cdot 3^2 \cdot 13$ on a

$$M_{936} = T_{288} + T_{264} - T_{216} - T_{192} + T_{144} + T_{120} - T_{72} - T_{48} + 1.$$

$\tau_{\mathcal{T}}(M_{936}) = 1$ pendant que $\tau(M_{936}) = 197$; pour présenter tous ses coefficients, on utilise 20484 bits.

- Pour $n = 8640 = 2^6 \cdot 3^3 \cdot 5$ on a

$$M_{8640} = -1 - T_{576} + T_{1728} + T_{2304},$$

$\tau_{\mathcal{T}}(M_{8640}) = 1$ pendant que $\tau(M_{8640}) = 1595$; pour présenter tous ses coefficients, on utilise 1323350 bits.

Conclusion du chapitre 3

Nous avons proposé l'algorithme 3.14 accompagné par une analyse complète de complexité, permettant d'affirmer qu'il est possible d'obtenir Φ_n et M_n dans une complexité arithmétique quasi-linéaire et dans une complexité binaire quasi-quadratique.

Chapitre 4

Évaluation des expressions trigonométriques

Sommaire

4.1 Évaluation d'une expression trigonométrique 69
4.1.1 Les fonctions élémentaires et l'évaluation de F 70
4.1.2 Méthode d'isolation dans la base des monômes 71
4.1.3 Détermination du signe de F par une méthode combinée algébrique-numérique 74
4.2 Calcul dans l'anneau $\mathbb{Z}[x]/\langle M_n \rangle$ 78
4.2.1 Abréger un élément de $\mathbb{Z}[x]/\langle M_n \rangle$ 78
4.2.2 Les opérations dans $\mathbb{Z}[x]/\langle M_n \rangle$ 79
4.3 Le polynôme minimal d'un élément de $\mathbb{Z}[2\cos\frac{\pi}{n}]$ 81
4.3.1 Définitions 81
4.3.2 Calcul de polynôme annulateur P_f 82
4.3.3 Complexité du calcul de M_F 85

Nous utilisons le calcul sur les polynômes unitaires de Chebyshev pour étudier une famille particulière de nombres algébriques, ceux qui sont éléments de $\mathbb{Z}[2\cos\frac{\pi}{n}]$, étant sous la forme $F = f(2\cos\frac{\pi}{n})$, où $f \in \mathbb{Z}[x]$ est représenté dans la base des polynômes unitaires de Chebyshev avec les coefficients de taille binaire au plus τ.
On peut supposer que f est toujours de degré d inférieur à $\lfloor \frac{n-1}{2} \rfloor$. Si $\deg f = D$, il est possible de se ramener à la situation en $D\tau_{\mathcal{T}}(f)$ opérations binaires.
L'avantage de la base des polynômes unitaires de Chebyshev nous permet d'améliorer notablement la complexité pour quelques problèmes, si l'on compare avec les méthodes algébriques usuelles utilisant les opérations dans la base des monômes. Par exemple, quand la méthode d'isolation décide le signe de F en exécutant $\widetilde{\mathcal{O}}(n^3+n^2\tau)$ opérations binaires, notre méthode d'approximation n'en utilise que $\widetilde{\mathcal{O}}(n^2\tau)$;
Nous étudions aussi le calcul du polynôme minimal de M_F de F. En utilisant des sommes de Newton, nous pouvons compléter ce calcul en exécutant $\widetilde{\mathcal{O}}(n^3\tau)$ opérations binaires, par rapport à la complexité binaire $\widetilde{\mathcal{O}}(n^4 + n^3\tau)$ si l'on utilise les sous-résultants.

4.1 Évaluation d'une expression trigonométrique

Étant donné

$$f = f_0 + \sum_{j=1}^{d} f_j T_j \in \mathbb{Z}[x],$$

avec $\tau_{\mathcal{T}}(f) = \tau$, nous voulons déterminer le signe de $F = f(2\cos\frac{\pi}{n})$: est-ce qu'il est nul, positif ou négatif ?

Effectivement, l'évaluation de F est un problème classique du calcul numérique qu'on sait résoudre depuis très longtemps. Nous commençons par cette évaluation :

4.1.1 Les fonctions élémentaires et l'évaluation de F

Pour évaluer la valeur de F, l'une des idées les plus naturelles est d'évaluer approximativement tous les $\cos\frac{k\pi}{n}$, $k = 1, \ldots, d$ puis combiner ces valeurs. Retournons au problème le plus classique du Calcul Numérique : comment peut-on calculer la valeur des fonctions élémentaires ?

Sur l'entrée présentée par des nombres flottants de n chiffres binaires, les complexités dans la table[1] 4.1 [Mul06, page 93] sont bien connues.

Fonction / Opération	Complexité
Addition	$\mathcal{O}(n)$
Multiplication	$\mathcal{O}(n \log_2 n \log_2 \log_2 n)$
Division et sqrt	$\mathcal{O}(\mathcal{M}(n))$
$\log_2, \exp$	$\mathcal{O}(\mathcal{M}(n) \log_2 n)$
$\sin, \cos, \arcsin, \arccos, \arctan$	$\mathcal{O}(\mathcal{M}(n) \log_2 n)$

TABLE 4.1 – *Complexité du calcul des fonctions / opérations élémentaires.*

Parlons un peu plus de la fonction cosinus que l'on va utiliser régulièrement. Nous aurons besoin des valeurs approximatives de $\cos\frac{k\pi}{n}, k = 1, \ldots n-1$.

Supposons que x soit un nombre flottant en binaire de n chiffres. L'une des évaluations les plus importantes pour notre travail, citée par Muller [Mul06, 2006] a été proposée par Brent dans les années 70.

Lemme 4.1. *[Bre75, Sec. 8] On peut calculer la valeur de π à la précision $2^{-\ell}$ en $\mathcal{O}(\mathcal{M}(\ell)\log_2 \ell) = \widetilde{\mathcal{O}}(\ell)$ opérations binaires.*

Lemme 4.2. *[Bre76, Sec.7] Soit x un nombre positif flottant de ℓ chiffres binaires, alors on peut calculer la valeur de $\cos x$ à la précision $2^{-\ell}$ par $\mathcal{O}(\mathcal{M}(\ell)\log_2 \ell) = \widetilde{\mathcal{O}}(\ell)$ opérations binaires.*

Partant du résultat de Brent, nous arrivons à une évaluation de $\cos\frac{k\pi}{n}$ pour $k = 1, \ldots, n-1$:

Lemme 4.3. *Pour chaque k donné $0 < k < n$, posons $\gamma_k = 2\cos\frac{k\pi}{n}$. Soit $\ell \in \mathbb{Z}_{>0}$. On peut calculer $c \in \mathbb{Q}$ de taille binaire $\tau(c) \leq \ell$ tel que $|c - \gamma_k| \leq 2^{-\ell}$ par $\widetilde{\mathcal{O}}(\ell + \log_2 n)$ opérations binaires.*

Démonstration. Nous pouvons calculer successivement :

- Calculer $r \in \mathbb{Q}$ avec $\tau(r) = 2(\ell + 2\log_2 n)$ tel que $|r - \frac{k\pi}{n}| \leq 2^{-\ell-1}$, ce qui demande $C_B^{(1)} = \widetilde{\mathcal{O}}(\ell + \log_2 n)$ opérations binaires (lemme 4.1) ;
- Calculer le nombre rationnel c, $\tau(c) = \ell$ tel que $|c - \cos r| \leq 2^{-\ell-1}$, ce qui utilise $C_B^{(2)} = \widetilde{\mathcal{O}}(\ell + \log_2 n)$ opérations binaires (lemme 4.2) ;

1. $\mathcal{M}$ est un temps de multiplication, voir la définition 2.5, [GG13, Déf. 8.26]

En conclusion, on utilise $C_B^{(1)} + C_B^{(2)} = \widetilde{\mathcal{O}}(\ell + \log_2 n)$ opérations binaires pour calculer c. □

Corollaire 4.4. *Soit $f = f_0 + \sum_{j=1}^{n-1} f_j T_j \in \mathbb{Z}[x]$, avec $\tau_{\mathcal{T}}(f) = \tau$, et ℓ un nombre entier positif. Alors :*

1. *On peut calculer $\overline{F} \in \mathbb{Q}$ de taille binaire $\mathcal{O}(n\tau + \ell)$ tel que $|\overline{F} - f(2\cos\frac{\pi}{n})| \leq 2^{-\ell}$ en $\widetilde{\mathcal{O}}(n\ell + n\tau)$ opérations binaires.*
2. *On peut obtenir les deux valeurs F^-, F^+ de tailles binaires $\mathcal{O}(n\tau + \ell)$ tels que $F^- \leq F \leq F^+$ et $F^+ - F^- \leq 2^{-\ell}$ en exécutant $\widetilde{\mathcal{O}}(n\ell + n\tau)$ opérations binaires.*

Démonstration. 1. En utilisant le lemme 4.3, on peut calculer les nombres rationnels $c_j, j = 1, \ldots, n-1$ tels que :

$$\left|c_j - \cos\tfrac{j\pi}{n}\right| \leq \tfrac{1}{n} \cdot 2^{-\ell-\tau}.$$

Ils sont de taille binaire $\mathcal{O}(\tau + \ell + \log_2 n)$, obtenus par $\widetilde{\mathcal{O}}(n\tau + n\ell)$ opérations binaires.

Posons $\overline{F} = f_0 + 2\sum_{i=1}^{n-1} f_i c_i$ alors

$$|\overline{F} - F| \leq 2\sum_{i=1}^{n-1} |f_i||c_i - \cos i\tfrac{\pi}{n}| \leq 2^{-\ell}.$$

Ici, les $f_i c_i$ ont une taille binaire $\mathcal{O}(\tau + \ell + \log_2 n)$, chacun peut être calculé avec $\widetilde{\mathcal{O}}(\tau + \ell + \log_2 n)$ operations binaires, par la complexité de la multiplication rapide (voir la proposition 2.6).

La valeur de $\overline{F}$ est de taille binaire $\mathcal{O}(\tau+\ell+\log_2 n)$, elle peut être calculée en $\widetilde{\mathcal{O}}(n\tau+n\ell)$ opérations binaires.

2. De la même façon, on prouve la deuxième partie du lemme. □

Bien que l'on puisse calculer approximativement F à une précision arbitraire, pour savoir si F est nul, une méthode numérique n'est pas suffisante en général.

Exemple 4.5. Prenons le polynôme minimal $M_{121} = -1 + T_{11} - T_{22} + T_{33} - T_{44} + T_{55}$ et une forme de Chebyshev arbitraire $h = 117T_{129} - 3T_{49} - 19T_{40} + 99$ alors $F = (h \cdot M_{121})(2\cos\frac{\pi}{121}) = 0$.

Faisons deux calculs avec Maple 18 (voir la figure A.2, annexe A.1) :

- `evalf`$(F, 50)$ sort $-1 \cdot 10^{-47}$;
- Par contre, les trois commandes `Is`$(F = 0)$, `Is`$(F > 0)$, `Is`$(F > 0)$ renvoient la même réponse `FAIL`.

Le calcul de Maple nous dit que la valeur de F est très proche de zéro mais cela ne permet pas de décider si F est vraiment nul. Il faut d'autres outils algébriques.

4.1.2 Méthode d'isolation dans la base des monômes

Vu que $2\cos\frac{\pi}{n}$ est une racine de U_n, on est dans une situation habituelle : calculer le signe de $f \in \mathbb{Z}[x]$ en une racine du $U_n \in \mathbb{Z}[x]$.

Algébriquement, on peut résoudre le problème à l'aide du pgcd, de l'évaluation d'un polynôme en un point rationnel et de l'isolation des racines.

Cette méthode utilise deux calculs d'actualité très importants du Calcul Formel :

Lemme 4.6. *[BLPR15, Lem. 6] Soit $f \in \mathbb{Q}[x]$, $x_0 \in \mathbb{Q}$ tels que $\deg f = d$, $\tau(f) = \tau$, $\tau(x_0) = \tau'$ alors $f(x_0)$ peut être évalué par $\widetilde{\mathcal{O}}(d(\tau+\tau'))$ opérations binaires, et $\tau(f(x_0)) = \mathcal{O}(d(\tau+\tau'))$.*

Si $f \in \mathbb{Z}[x]$ ou le ppcm *des dénominateurs de ses coefficients a une taille binaire $\mathcal{O}(\tau)$ alors $\tau(f(x_0)) = \mathcal{O}(\tau + d\tau')$.*

Lemme 4.7 (Isolation rapide, Sagraloff et al. [MSW14])**.** *Soit f un polynôme à coefficients entiers, $\deg f = d$, $\tau(f) = \tau$. Alors l'isolation de toutes les racines de f peut être exécutée en temps de calcul $\widetilde{\mathcal{O}}(d^3 + d^2\tau)$, en plus on peut présenter tous les intervalles d'isolation par $\mathcal{O}(d\tau)$ octets.*

En utilisant deux résultats ci-dessus avec quelques autres calculs classiques on obtient :

Proposition 4.8. *Soit f une forme de Chebyshev de $\mathbb{Z}[x]$ tel que $\deg f = d < \frac{n}{2}$, $\tau_{\mathcal{T}}(f) = \tau$. Alors on peut déterminer le signe de $F = f(2\cos\frac{\pi}{n})$ en $\widetilde{\mathcal{O}}(n^3+n^2\tau)$ opérations binaires.*

Démonstration. Nous utilisons l'algorithme 4.9 suivant :

Algorithme 4.9 : Déterminer le signe de F par l'isolation

Entrées : $F = f(2\cos\frac{\pi}{n})$ où $f = f_0 + \sum_{j=1}^{n-1} f_j T_j \in \mathbb{Z}[x]$
Sorties : Signe de F

```
begin
    Calculer [X]f, [X]U_n ;
    Calculer g = (f, U_n);
    Calculer h = f.U_n/g = g · f/g · U_n/g /* dénotons deg h = d_h et τ(h) = τ_h */
    Isoler les racines de h;
    /* cela renvoie un liste d'intervalles I_γ, γ ∈ Z(h) avec les bornes
       rationnelles de taille binaire τ_γ ; on pose alors Σ = Σ_{γ∈Z(h)} τ_γ
       */
    Calculer approximativement γ̃_1 ≃ 2 cos π/n à la précision 2^{-1-Σ};
    Trouver l'intervalle d'isolation I_α qui contient γ̃_1;
    /* Dénotons u, v deux bornes de I_α */
    Calculer les signes de g aux deux bornes u, v de I_{γ_1};
    si g(u)g(v) < 0 alors
        retourner F = 0
    sinon
        Evaluer f(u);
        retourner Signe de f(u)
```

Preuve de correction. Rappelons que la notation $\mathcal{Z}(P)$ indique l'ensemble du polynôme P.

Parce que $g = (f, U_n)$ et $h = \frac{fU_n}{g}$ alors

$$\mathcal{Z}(h) = (\mathcal{Z}(f)\Delta\mathcal{Z}(U_n)) \cup \mathcal{Z}(g),$$ [2]

2. $A\Delta B = \{\, x \in A, x \notin B\} \cup \{x \in B, x \notin A\}$, où A, B deux ensembles.

cela veut dire qu'il existe un unique intervalle d'isolation I_α qui contient $2\cos\frac{\pi}{n}$, étant racine de U_n. Mais $|\widetilde{\gamma} - 2\cos\frac{\pi}{n}| \leq 2^{-1-\Sigma}$, l'intervalle I_α doit contenir $\widetilde{\gamma}$, ce qui confirme que le calcul expliqué sur la ligne numéro 7 de l'algorithme est significatif.

On a $\mathcal{Z}(f) \subset \mathcal{Z}(h)$ alors :

$$\mathcal{Z}(f) \cap I_\alpha \subset \mathcal{Z}(h) \cap I_\alpha = \left\{2\cos\frac{\pi}{n}\right\}. \qquad (\clubsuit)$$

Étant un intervalle d'isolation, $f(u) \cdot f(v) \neq 0$, il n'y a que deux cases :

i) $f(u) \cdot f(v) < 0 \Rightarrow \mathcal{Z}(f) \cap (u, v) \neq \emptyset$, d'après ($\clubsuit$), $2\cos\frac{\pi}{n} \in \mathcal{Z}(f)$ ou $f(2\cos\frac{\pi}{n}) = 0$;

ii) $f(u) \cdot f(v) > 0$, d'après ($\clubsuit$), $f(x) \neq 0$ pour tout $x \in (u, v)$ donc $\mathrm{Signe}(f(2\cos\frac{\pi}{n})) = \mathrm{Signe}(f(u))$;

Ce qui confirme la correction de l'algorithme. □

Analyse de complexité

Pour une expression concise, le symbole $\rightsquigarrow$ impliquera l'évaluation de la taille des sous-sorties. La complexité du calcul est analysée comme suit :

1. Parce que $U_n = \frac{1}{n}T'_n$, on peut l'obtenir en $\widetilde{\mathcal{O}}(n^2)$ opérations binaires (voir la proposition 2.13) ; Comme f est donné par sa forme de Chebyshev, il faut faire un changement de base, $[\mathcal{X}]f$ est calculé en $C_B^{(1)} = \widetilde{\mathcal{O}}(d^2 + d\tau) = \widetilde{\mathcal{O}}(n^2 + n\tau)$ opérations binaires (proposition 2.37) ;

$\rightsquigarrow$ $\deg f = d$, $\tau(f) \leq d + \tau + 1 = \tau + \mathcal{O}(d)$.

2. En appliquant la complexité du calcul de pgcd qui vient de la proposition 2.22, le calcul de $g = (f, U_n)$ utilise $C_B^{(2)} = \widetilde{\mathcal{O}}(n^2 \max\{\tau(U_n), \tau(f)\}) = \widetilde{\mathcal{O}}(n^3 + n^2\tau)$ opérations binaires ;

$\rightsquigarrow$ D'après la borne de Mignotte (lemme 2.24), étant diviseur de U_n, g satisfait : $\deg g \leq d$ et $\tau(g) \leq \deg(f, U_n) + \frac{1}{2}\log_2(1 + \deg U_n) + \tau(U_n) = \mathcal{O}(n)$;

3. Pour le calcul de h (ligne 3), on prend l'entrée : $\deg(f.U_n) \leq \frac{3n}{2}$, $\tau(f.U_n) \leq \tau(f) + \tau(U_n) + \log_2 d = \tau + \mathcal{O}(n)$; $\deg g \leq d$, $\tau(g) = \mathcal{O}(n)$. Grâce à la proposition 2.9 sur la complexité de la division exacte dans la base des monômes, le calcul de h utilise $C_B^{(3)} = \widetilde{\mathcal{O}}(n^2 + n\tau)$ opérations binaires.

$\rightsquigarrow$ Appliquons encore la borne de Mignotte (lemme 2.24) on a : $\deg h \leq 3n/2$ et $\tau(h) \leq \tau(U_n) + \tau(\frac{f}{g}) + \log_2 n = \tau + \mathcal{O}(n)$;

4. Vu le théorème 4.7, l'isolation des racines de h (ligne 4) utilise $C_B^{(4)} = \widetilde{\mathcal{O}}(d_h^2\tau_h + d_h^3) = \widetilde{\mathcal{O}}(n^2\tau + n^3)$ opérations binaires,

$\rightsquigarrow$ elle engendre les intervalles $I_\gamma, \gamma \in \mathcal{Z}(h)$ avec les bornes rationnelles de taille binaire τ_γ tel que $\Sigma = \sum_{\gamma \in \mathcal{Z}(h)} \tau_\gamma = \mathcal{O}(d_h\tau_h) = \widetilde{\mathcal{O}}(n\tau + n^2)$;

5. Vu le lemme 4.3, le calcul de $\widetilde{\gamma}_1 \simeq 2\cos\frac{\pi}{n}$ avec précision $1 + \Sigma = \mathcal{O}(d_h\tau_h)$ (ligne 5) utilise $C_B^{(5)} = \widetilde{\mathcal{O}}(d_h\tau_h) = \widetilde{\mathcal{O}}(n^2 + n\tau)$ opérations binaires,

$\rightsquigarrow$ $\tau(\gamma_1) = \mathcal{O}(d_h\tau_h) = \mathcal{O}(n^2 + n\tau)$;

6. Pour trouver l'intervalle d'isolation qui correspond avec γ_1 (ligne 6), nous faisons comme suit : Comparons $\widetilde{\gamma}_1$ avec l'intervalle au milieu de la liste L, i.e l'intervalle de l'indice $\lfloor\frac{|L|}{2}\rfloor$; Si cette comparaison échoue, on sait γ_1 est contenu dans la demi liste à gauche ou à droite ; prendre cette demi liste et répéter la procédure jusqu'à ce que la comparaison réussisse.

De cette manière, il nous faut $\mathcal{O}(\log_2 2n)$ comparaisons entre les bornes rationnelles des I_γ avec la valeur approximative $\widetilde{\gamma}_1$ de $2\cos\frac{\pi}{n}$. Chaque comparison utilise au plus $\widetilde{\mathcal{O}}(d_h\tau_h)$

operations binaires. Donc, l'intervalle I_α qui contient γ_1 peut être identifié en réalisant $C_B^{(6)} = \widetilde{\mathcal{O}}(d_h\tau_h) = \widetilde{\mathcal{O}}(n^2 + n\tau)$ opérations binaires ;

7. Vu la proposition 4.6, le calcul des $g(u), g(v)$ et la détermination de signe$(g(u)g(v))$ (ligne 7), utilisent $C_B^{(7)} = \widetilde{\mathcal{O}}(d_g(\tau_g + d_h\tau_h)) = \widetilde{\mathcal{O}}(n^2 + n\tau)$ opérations binaires ;

8. S'il est nécessaire, on peut faire une évaluation (ligne 10) de f en un point rationnel de taille binaire $\mathcal{O}(d_d\tau_h)$. Vu la proposition 4.6, nous utilisons en plus $C_B^{(8)} = \widetilde{\mathcal{O}}(d_f(\tau_f + d_h\tau_h)) = \widetilde{\mathcal{O}}(n^3 + n^2\tau)$ opérations binaires.

La dernière complexité de la détermination du signe est synthétisée par :

$$C_B = \sum_{i=1}^{8} C_B^{(i)} = \widetilde{\mathcal{O}}(n^3 + n^2\tau) \text{ opérations binaires.}$$

□

4.1.3 Détermination du signe de F par une méthode combinée algébrique-numérique

Par convention, dans cette section, on considère toujours des nombres $F = f(2\cos\frac{\pi}{n})$, où f est une forme de Chebyshev dans $\mathbb{Z}[x]$ de degré $d < \frac{n}{2}$.

Tester si F est nul

Afin de décider si F est nul, nous avons le résultat :

Proposition 4.10 (`TestAZéro`). *Soit $F = f(2\cos\frac{\pi}{n})$ avec $f = f_0 + \sum_{j=1}^{d} f_jT_j \in \mathbb{Z}[x]$, $\tau_{\mathcal{T}}(f) = \tau$. Alors on peut décider si F est nul par $\widetilde{\mathcal{O}}(n)$ opérations arithmétiques, $\widetilde{\mathcal{O}}(n^2 + n\tau)$ opérations binaires.*

Démonstration. M_n étant le polynôme minimal de $2\cos\frac{\pi}{n}$, algébriquement :

$$f(2\cos\tfrac{\pi}{n}) = 0 \Leftrightarrow M_n \mid f.$$

- Par la proposition 3.16, le calcul de $[\mathcal{T}]M_n$ utilise $C_B^{(1)} = \widetilde{\mathcal{O}}(n^2)$ opérations binaires. En plus $\deg M_n = \mathcal{O}(n)$, $\tau_{\mathcal{T}}(M_n) = \widetilde{\mathcal{O}}(n)$, vu le lemme 3.15 ;
- Pour décider si $M_n \mid f$ dans la base des polynômes unitaires de Chebyshev, nous utilisons le coût de la division exacte mentionné dans le corollaire 2.26. Ici $\deg f < n$, $\tau_{\mathcal{T}}(f) = \tau$ alors, on utilise $C_B^{(2)} = \widetilde{\mathcal{O}}(n^2 + n\max\{\tau_{\mathcal{T}}(M_n), \tau_{\mathcal{T}}(f)\}) = \widetilde{\mathcal{O}}(n^2 + n\tau)$ opérations binaires.

Au total, la complexité binaire du `TestAZéro` est : $C_B^{(1)} + C_B^{(2)} = \widetilde{\mathcal{O}}(n^2 + n\tau)$.

□

Traiter le cas où $F \neq 0$

Supposons que le `TestAZéro` retourne $F \neq 0$.

Pour déterminer le signe de F, la technique est très simple à l'aide d'un raffinement : Comme $F \neq 0$, il existe $\varepsilon > 0$ tel que $|F| > \varepsilon$. Alors, si $F^- \leq F \leq F^+$, $|F^- - F^+| < \varepsilon$, il faudra que F^+, F^- et F aient un même signe.

Ainsi, en calculant F approximativement, si l'on part par $\ell = 1$ puis double la précision demandée jusqu'à ce que $F^+F^- > 0$, le processus devra se terminer après au plus $-\lceil\log_2\varepsilon\rceil$ itérations.

On sait que l'algorithme doit se terminer mais pour évaluer sa complexité, il faut savoir le nombre d'itérations dans le pire cas. Cette information peut être obtenue si l'on sait minorer la valeur absolue de F. En fait on a :

Lemme 4.11. *Soit f une forme de Chebyshev, $\gamma = 2\cos\frac{k\pi}{n}$ une racine réelle de M_n tels que $f(\gamma) \neq 0$. Alors :*

$$|f(\gamma)| \geq \|f\|_T^{1-\frac{n}{2}} \text{ [3]}. \tag{4.1}$$

Nous démontrons le lemme en utilisant les résultants. Soit K un corps et $A, B \in K[x]$, $A(x) = a_k x^k + \ldots + a_0$, $B(x) = b_n x^n + \ldots + b_0 (a_k b_n \neq 0)$, la matrice de Sylvester de A, B est définie par :

$$\mathrm{Syl}(A,B) = \begin{bmatrix} a_k & a_{k-1} & \ldots & a_0 & & & \\ & a_k & a_{k-1} & \ldots & a_0 & & \\ & & \ddots & \ddots & & \ddots & \\ & & & a_k & a_{k-1} & \ldots & a_0 \\ b_n & b_{n-1} & \ldots & b_0 & & & \\ & b_n & b_{n-1} & \ldots & b_0 & & \\ & & \ddots & \ddots & & \ddots & \\ & & & b_k & b_{n-1} & \ldots & b_0 \end{bmatrix}$$

C'est la transposée de la matrice présentant l'application linéaire :

$$\begin{aligned} \varphi_{A,B} : K[x]_{\deg<n} \times K[x]_{\deg<m} &\to K[x]_{\deg<m+n} \\ (U,V) &\mapsto A.U + B.V, \end{aligned} \tag{4.2}$$

où $\deg U < n$, $\deg V < m$, et dans l'arrivée on utilise la base des monômes $1, x, \ldots, x^{m+n-1}$.

Définition 4.12. Le déterminant de $\mathrm{Syl}(A,B)$ s'appelle le *Résultant* de A, B, noté $\mathrm{Res}(A,B)$. Si l'on veut insister sur l'élimination de la variable x, on écrit $\mathrm{Res}_x(A,B)$.

Suivant est une propriété classique du Résultant :

Lemme 4.13. *[BCG+14, Chap. 6, Prop. 4]*

1. *Si les racines de A et B sont connues, $A = a_k(x-\alpha_1)\ldots(x-\alpha_k)$, $B = b_n(x-\beta_1)\ldots(x-\beta_n)$ alors*

$$\begin{aligned} \mathrm{Res}(A,B) &= a_k^n b_n^k \prod_{i,j}(\alpha_i - \beta_j) = (-1)^{mn} b_n^k \prod_{1\leq j\leq n} A(\beta_j) \\ &= a_k^n \prod_{1\leq i\leq m} B(\alpha_i) = (-1)^{mn} Res(B,A). \end{aligned} \tag{4.3}$$

2. *Considérons l'anneau $Q[x,z]$ comme $Q[z][x]$, si $A(x) = (x-x_1)\ldots(x-x_k) \in Q[x]$ alors pour tout $B \in Q[x]$,*

$$\mathrm{Res}_x(z - B(x), A(x)) = \prod_{j=1}^{k}(z - B(x_k)). \tag{4.4}$$

Corollaire 4.14. *Pour tout $A, B \in \mathbb{R}[x]$, $(A,B) = 1$ si et seulement si $\mathrm{Res}(A,B) \neq 0$.*

Preuve du lemme 4.11. Parce que $f(\gamma) \neq 0$ alors $(M_n, f) = 1$ ou $|\mathrm{Res}(M_n, f)| \neq 0$.
Mais $\mathrm{Res}(M_n, f) = \det(\mathrm{Syl}(M_n, f)) \in \mathbb{Z}$ alors $|\mathrm{Res}(M_n, f)| \geq 1$.

3. La norme $\|\|_T$ est définie par $\|f\|_T = |f_0| + 2\sum_{j=1}^{d}|f_j|$ dans le chapitre 2.

Car M_n est unitaire, l'identité (4.3) nous donne :

$$1 \leq |\mathrm{Res}(M_n, f)| = \left| \prod_{M_n(\gamma)=0} f(\gamma) \right| = |f(\gamma)| \prod_{\substack{M_n(\gamma')=0 \\ \gamma' \neq \gamma}} |f(\gamma')|.$$

Mais $\cos x \leq 1 \; \forall x \in \mathbb{R}$, pour chaque racine $\gamma' \neq \gamma$ de M_n, il est clair que :

$$|f(\gamma')| = \left| f_0 + 2 \sum_{j=1}^{d} f_j \cos t_j \right| \leq \|f\|_T.$$

Il y a au total $\deg M_n - 1 = \frac{\varphi(n)}{2} - 1 \leq \frac{n}{2} - 1$ racines $\gamma' \neq \gamma$, donc :

$$\begin{aligned} |f(\gamma)| &\geq \|f\|_T^{1-\frac{\varphi(n)}{2}} \\ &\geq \|f\|_T^{1-\frac{n}{2}}. \end{aligned} \tag{4.5}$$

□

Nous proposons l'algorithme 4.15 permettant de calculer $\mathrm{Signe}(f(\gamma))$:

Algorithme 4.15 : `SIGNE` pour déterminer le signe de $F = f(2\cos\frac{k\pi}{n})$

Entrées : $f = f_0 + \sum_{j=1}^{d} f_j T_j$, k tel que $(k, 2n) = 1$

Sorties : $\mathrm{Signe}\Big(f(2\cos k\frac{\pi}{n})\Big)$

```
1  begin
2      Calculer [T]M_n ;
3      si M_n | f alors
4          retourner 0
5      ℓ ← 1;
6      Calculer F^- et F^+ = F^- + 1 tels que F^- < F < F+;
7      répéter
8          Calculer l'intervalle [F^-, F^+] de longueur binaire au plus 2^{-ℓ} qui contient
           f(γ);
9          ℓ ← 2ℓ;
10     jusqu'à ce que F^+ · F^- > 0;
11 retourner Signe(F^+)
```

Preuve de correction. Comme M_n est le polynôme minimal de $2\cos\frac{\pi}{n}$, $F = 0 \Leftrightarrow M_n | f$. Il suffit de prouver que le processus doit se terminer pour le cas où $F \neq 0$.

En fait, si $F \neq 0$, il existe $\delta > 0$ tel que $|F| > \delta$. Dès que $\ell > -\lceil \log_2 \delta \rceil - 1$, on a $F^+ - F^- < \frac{\delta}{2}$ ce qui confirme que F^- et F^+ ont le même signe, c'est le point où l'algorithme se termine. □

Remarque 4.16. Dès que le test `SIGNE` se termine, on peut garder le nombre k d'itérations exécutées. Ce nombre nous permet de minorer la valeur absolue de F. En fait :

$$|F| \geq 2^{-2^k}.$$

Proposition 4.17. *Soit f une forme de Chebyshev de degré au plus $d < \frac{n}{2}$, $\tau_{\mathcal{T}}(f) = \tau$, $\gamma = 2\cos k\frac{\pi}{n}$ avec $(k, 2n) = 1$. Alors :*

1. *Une valeur approximative de $f(\gamma)$ à la précision $2^{-\ell}$ peut être calculée en $\widetilde{\mathcal{O}}(n\ell+n\tau)$ opérations binaires ;*
2. *Le signe de $f(\gamma)$ peut être déterminé par $\widetilde{\mathcal{O}}(n^2\tau)$ opérations binaires ;*

Démonstration. 1. La première conclusion est énnoncée au corollaire 4.4 ;
2. Considérons l'algorithme 4.15 :

Nous faisons d'abord le `TestAZéro` en $C_B^{(1)} = \widetilde{\mathcal{O}}(n^2 + n\tau)$ opérations binaires (proposition 4.10) ;

Dans l'inégalité (4.1), posons

$$k = \lceil -\log_2 \|f\|_T^{1-\frac{n}{2}} \rceil, \tag{4.6}$$

alors

$$\begin{aligned} k &\leq 1 + \lceil \tfrac{n-2}{2} \log_2 \|f\|_T \rceil \\ &\leq 1 + \left\lceil \tfrac{n-2}{2} [\log_2(2n+1) + \tau] \right\rceil = \widetilde{\mathcal{O}}(n\tau) \end{aligned}$$

Nous savons que la boucle (lignes 6 - 8) doit se terminer après au plus $\ell = \lceil \log_2 k \rceil$ itérations. Il faut alors calculer les valeurs approximatives de $f(\gamma)$ ℓ fois, aux précisions successivement $2^{-2^j}, j = 1, \ldots, \ell$.

D'après le corollaire 4.4, le nombre d'opérations binaires utilisées est donc borné par :

$$C_B^{(2)} = \sum_{j=1}^{\ell} \widetilde{\mathcal{O}}(n2^j + n\tau) = \widetilde{\mathcal{O}}(n2^{\ell+1} + n\ell\tau).$$

Mais $\ell \leq 1 + \log_2 k$, $k = \widetilde{\mathcal{O}}(n\tau)$ donc $\widetilde{\mathcal{O}}(n2^{\ell+1} + n\ell\tau) = \widetilde{\mathcal{O}}(n^2\tau)$.

En résumé, l'algorithme marche en utilisant $C_B = C_B^{(1)} + C_B^{(2)} = \widetilde{\mathcal{O}}(n^2\tau)$ opérations binaires. □

La complexité $\widetilde{\mathcal{O}}(n^2\tau)$ est évaluée pour le pire cas. En réalité, la minoration dans l'inégalité (4.1) est normalement trop sévère. Généralement, si le `TestAZéro` échoue, la boucle va se terminer plus tôt.

Exemple 4.18. Considérons quatre nombres algébriques $a_j, j = 1, 2, 3, 4$ définis par :

$$\begin{aligned} a_1 &= 16 \sin \tfrac{\pi}{9} \sin \tfrac{5\pi}{18} \sin \tfrac{11\pi}{39} \sin \tfrac{3\pi}{8} \\ a_2 &= 16 \sin \tfrac{2\pi}{45} \sin \tfrac{4\pi}{25} \sin \tfrac{20\pi}{49} \sin \tfrac{17\pi}{40} \\ a_3 &= 48 \cos \tfrac{\pi}{18} \cos \tfrac{7\pi}{15} \cos \tfrac{9\pi}{22} \cos \tfrac{12\pi}{49} \\ a_4 &= 16 \cos \tfrac{2\pi}{5} \cos \tfrac{5\pi}{16} \cos \tfrac{8\pi}{27} \cos \tfrac{104\pi}{357}. \end{aligned}$$

Dans son travail [Mye93], Myerson a indiqué que $a_1 \simeq 3$;, $a_2 \simeq 1$;

En ayant travaillé sur Maple, nous avons trouvé aussi $a_3 \simeq 1$, $a_4 \simeq 1$.

Ecrivons a_j, $j = 1, \ldots, 4$ sous une somme des cosinus puis la valeur d'une forme de Chebyshev (calculé avec Maple 18, voir la figure A.5, l'annexe A.2), on obtient :

$$\begin{aligned}
a_1 &= [-T_{459} - T_{451} - T_{277} + T_{251} + T_{243} + T_{165} + T_{69} - T_{43}](2\cos\tfrac{\pi}{936}) \\
&= f_1(2\cos\tfrac{\pi}{936}); \\
a_2 &= [-T_{32747} + T_{24907} - T_{19517} - T_{16547} + T_{11677} + T_{8707} + T_{4523} - T_{3317}](2\cos\tfrac{\pi}{88200}) \\
&= f_2(2\cos\tfrac{\pi}{88200}); \\
a_3 &= [-3T_{10301} + 3T_{8684} - 3T_{7606} + 3T_{5989} + 3T_{5891} - 3T_{4274} + 3T_{3196} - 3T_{1579} - 1](2\cos\tfrac{\pi}{24255}) \\
&= f_3((2\cos\tfrac{\pi}{24255}); \\
a_4 &= [-T_{128491} - T_{83509} - T_{77141} - T_{75179} - T_{72619} + T_{32101} + T_{23771} + T_{21211} - 1](2\cos\tfrac{\pi}{257040}) \\
&= f_4(2\cos\tfrac{\pi}{257040}).
\end{aligned}$$

Les quatre nombres algébriques $a_1 - 3$, $a_2 - 1$, $a_3 - 1$, $a_4 - 1$ sont tous très proches de zéro.

- Pour $a_1 - 3$: $n = 936$, $\frac{\varphi(n)}{2} = 288$, l'inégalité (4.5) devient : $|a_1 - 3| \geq \epsilon_1$, avec

$$\epsilon_1 = \|f_1\|_T^{1-288} = 19^{-287} > 2^{-1220},$$

  le test `SIGNE` doit se terminer après $\lceil \log_2 1220 \rceil = 11$ itérations au plus.

- Pour $a_2 - 1$: $n = 88200$, $\frac{\varphi(n)}{2} = 20160$, l'inégalité (4.5) devient : $|a_2 - 1| \geq \epsilon_2$, avec

$$\epsilon_2 = \|f_2\|_T^{1-20160} = 17^{-20159} > 2^{-82400},$$

  le test `SIGNE` doit se terminer après $\lceil \log_2 82400 \rceil = 17$ itérations au plus.

- Pour $a_3 - 1$: $n = 24255$, $\frac{\varphi(n)}{2} = 5040$, l'inégalité (4.5) devient : $|a_3 - 1| \geq \epsilon_3$, avec

$$\epsilon_3 = \|f_3\|_T^{1-5040} = 49^{-5039} > 2^{-28293},$$

  le test `SIGNE` doit se terminer après $\lceil \log_2 28293 \rceil = 15$ itérations au plus.

- Pour $a_4 - 1$: $n = 257040$, $\frac{\varphi(n)}{2} = 55296$, l'inégalité (4.5) devient : $|a_4 - 1| \geq \epsilon_4$, avec

$$\epsilon_4 = \|f_4\|_T^{1-55296} = 17^{-55295} > 2^{-226017},$$

  le test `SIGNE` doit se terminer après $\lceil \log_2 226017 \rceil = 18$ itérations au plus.

En pratique, pour toutes les quatre expressions, l'algorithme 4.15 se termine après 5 itérations.

Selon la remarque 4.16, on a $|a_j| > 2^{-32}, j = 1, \ldots, 4$.

Il est aussi possible d'expérimenter avec un outil existent de Maple pour reconnaitre que $|a_j| > 10^{-12}, j = 1, \ldots, 4$ (la commande `shake` par exemple, voir la figure A.5).

4.2 Calcul dans l'anneau $\mathbb{Z}[x]/\langle M_n \rangle$

Dans ce travail nous rencontrons très souvent des nombres algébriques déterminés par des expressions trigonométriques de type $r = r_0 + \sum_{j=1}^{D} r_j \cos\frac{j\pi}{n}, r_j \in \mathbb{Z}$. Calculer sur ces nombres peut être considéré comme travaillant dans l'anneau $\mathbb{Z}[x]/\langle M_n \rangle$.

4.2.1 Abréger un élément de $\mathbb{Z}[x]/\langle M_n\rangle$

La vraie réduction modulo M_n d'un polynôme $f \in \mathbb{Z}[x]$ est $\mathrm{Rem}(f, M_n)$; Vu la proposition 2.21, nous savons que la taille binaire des coefficients du reste risque d'être multipliée : $\tau_{\mathcal{T}}(\mathrm{Rem}(f, M_n)) = \mathcal{O}(\deg f \cdot \tau_{\mathcal{T}}(f))$.

C'est pour cette raison qu'on n'utilise presque jamais la vraie réduction en pratique.

La fonction cosinus est périodique, en plus $\cos(k\pi \pm x) = (-1)^k \cos x$, alors $(T_{k.n\pm i} + T_i)(2\cos\frac{\pi}{n}) = 0$ ou :

$$T_{k.n+i} \equiv T_{kn-i} \equiv (-1)^k T_i \pmod{M_n}. \tag{4.7}$$

Ainsi, toutes les formes de Chebyshev peuvent être abrégées par modulo M_n à un degré inférieur à d. Nous allons préciser cette procédure en deux aspects : la complexité de la réduction et l'augmentation de taille binaire des coefficients.

Définition 4.19 (Forme abrégée d'une forme de Chebyshev)**.** Soit $f = f_0 + \sum_{j=1}^{D} f_j T_j$, une forme de Chebyshev dans $\mathbb{Z}[x]$, n un nombre entier positif. En posant

$$d = \lfloor \tfrac{n-1}{2} \rfloor; \quad r = \mathrm{Rem}(D, 2n), D = k.2n + r; \quad f_j = 0, j = r+1, \ldots, 2n-1,$$

on définit la forme de Chebyshev $\mathrm{Abr}_n(f)$ par :

$$\begin{aligned}\mathrm{Abr}_n(f) &\overset{\text{déf}}{=} f_0 + \sum_{j=1}^{D} (-1)^{\lfloor \frac{i}{n} \rfloor} f_i T_{i(\bmod n)} \\ &= \Big(f_0 + 2\sum_{j=1}^{2k+1} (-1)^j f_{nj}\Big) + \sum_{j=1}^{d} \Big(\sum_{i=1}^{k} (f_{i.2n+j} - f_{i.2n+n-j} - f_{i.2n+n+j} + f_{i.2n+2n-j})\Big) T_j\end{aligned}$$

Remarque 4.20.

1. Grâce à la formule (4.7), nous signalons que la forme $\mathrm{Abr}_n(f)$ est définie uniquement par f, $\deg(\mathrm{Abr}_n(f)) \le d$, en plus :

   $$\mathrm{Abr}_n(f) \equiv f \pmod{M_n}, \text{ i.e. } (\mathrm{Abr}_n(f) - f)(2\cos\tfrac{\pi}{n}) = 0.$$

   L'application Abr_n ne change pas la valeur des formes de Chebyshev dans $\mathbb{Z}[x]/\langle M_n(x)\rangle$. S'il n'y a pas de malentendu, on va écrire simplement $\mathrm{Abr}(f)$ à la place de $\mathrm{Abr}_n(f)$.
2. Il est clair que :

   $$\|\mathrm{Abr}_n(f)\|_T \le \|f\|_T.$$

Lemme 4.21. *Soit $f = f_0 + \sum_{j=1}^{D} f_j T_j$ une forme de Chebyshev de $\mathbb{Z}[x]$, ($\deg f = D$, $\tau_{\mathcal{T}}(f) = \tau$), n un nombre entier positif, alors :*

$$\tau_{\mathcal{T}}(\mathrm{Abr}(f)) \le \tau + 1 + \log_2 \tfrac{D}{n}. \tag{4.8}$$

On peut calculer $\mathrm{Abr}(f)$ avec D additions de $\mathbb{Z}$, dans $D\tau$ opérations binaires.

Démonstration. Vu la définition de $\mathrm{Abr}(f)$ et la taille binaire τ de tous les $f_j, j = 1, \ldots, D$, le nombre d'additions D et le nombre d'opérations binaires $D\tau$ sont tous les deux clairs.

Pour la taille binaire des coefficients de $\mathrm{Abr}(f)$, remarquons qu'il y a au plus $k \times 4$ termes f_i's dans chaque $[T_j](\mathrm{Abr}(f))$, alors :

$$|[T_j](\mathrm{Abr}(f))| \le 4k\|f\|_\infty, j = 0, \ldots, d,$$

Car $k = \lfloor \frac{D}{2n} \rfloor$, en travaillant dans $\mathbb{Z}[x]$ on peut déduire :

$$\tau_{\mathcal{T}}(\mathrm{Abr}(f)) \le \tau_{\mathcal{T}}(f) + 1 + \log_2 \tfrac{D}{n} \text{ c.q.f.d}$$

□

Remarque 4.22. L'un des avantages du calcul dans $\mathbb{Z}[x]/\langle M_n\rangle$ est qu'on peut toujours borner le degré des données par $d = \lfloor \frac{n-1}{2} \rfloor$. Ce fait nous aide à économiser notablement le nombre d'opérations arithmétiques.

4.2.2 Les opérations dans $\mathbb{Z}[x]/\langle M_n\rangle$

Nous voyons comment l'application Abr peut aider à calculer rapidement dans l'anneau $\mathbb{Z}[x]/\langle M_n(x)\rangle$.

Pour l'addition, par la définition de Abr, il est clair que :

$$\mathrm{Abr}(f+g) = \mathrm{Abr}(f) + \mathrm{Abr}(g)$$

pour toute forme de Chebyshev f, g.

En ce qui concerne la multiplication, nous avons autrement :

Lemme 4.23. *Soit $f = f_0 + \sum_{i=1}^{d} f_i T_i$, $g = g_0 + \sum_{i=1}^{d} g_i T_i$ deux formes de Chebyshev dans $\mathbb{Z}[x]$ avec les coefficients de taille binaire inférieure à τ. Alors :*

$$\tau_{\mathcal{T}}(\mathrm{Abr}([\mathcal{T}]f \cdot g)) \leq \tau_{\mathcal{T}}(f) + \tau_{\mathcal{T}}(g) + \log_2 n + 1. \tag{4.9}$$

Cette forme abrégée de Chebyshev peut être calculée en $\widetilde{\mathcal{O}}(n\tau)$ operations binaires.

Démonstration. Posons $h = [\mathcal{T}](f \cdot g)$. Vu la proposition 2.17, $\tau_{\mathcal{T}}(h) \leq \tau_{\mathcal{T}}(f) + \tau_{\mathcal{T}}(g) + \log_2 n$, elle peut être calculée en $C_B^{(1)} = \widetilde{\mathcal{O}}(d\tau)$ opérations binaires ;

D'après le lemme 4.21, le calcul de $\mathrm{Abr}(h)$ utilise $C_B^{(2)} = n\tau_{\mathcal{T}}(h) = \widetilde{\mathcal{O}}(n\tau_{\mathcal{T}})$ opérations binaires ;

En résumé, le calcul de $\mathrm{Abr}([\mathcal{T}]fg)$ utilise

$$C_B = C_B^{(1)} + C_B^{(2)} = \widetilde{\mathcal{O}}(n\tau) \text{ opérations binaires.}$$

□

Si l'on considère le produit de plusieurs éléments de $\mathbb{Z}[x]/\langle M_n(x)\rangle$, l'avantage de la base $\mathcal{T}$ s'est distingué.

Proposition 4.24. *Soit $[\mathcal{T}]f_j, j = 0, \ldots, k-1$ des formes de Chebyshev de degrés au plus d avec les coefficients entiers de taille binaire au plus τ, alors il existe une forme de Chebyshev h de $\mathbb{Z}[x]$ ($\deg h \leq d$) qu'on peut calculer en $\widetilde{\mathcal{O}}(kn\tau)$ opérations binaires tel que :*

$$\begin{cases} \tau_{\mathcal{T}}(h) & \leq k\tau + (k-1)(\log_2 n + 1) = k\tau + \mathcal{O}(k \log_2 n) \\ h & \equiv \prod_{j=0}^{k-1} f_j \pmod{M_n}. \end{cases} \tag{4.10}$$

Démonstration. Supposons que $k = 2^\ell$ (sinon on dénote $\ell = \lceil \log_2(k+1) \rceil$ puis rajoutons $f_j = 1$ pour tout $j = k, \ldots, 2^\ell - 1$). On va utiliser la stratégie "Diviser pour Régner" illustrée par l'arbre dans le schéma ((2.8)) pour calculer successivement :

$$\begin{aligned} f_{0,j} &= f_j, & j &= 0, \ldots, 2^\ell - 1 \\ f_{c,j} &= \mathrm{Abr}(f_{c-1,2j} \cdot f_{c-1,2j+1}), & j &= 0, \ldots, 2^{\ell-c} - 1 \\ & & &\text{pour } c = 1, \ldots, \ell. \end{aligned} \tag{4.11}$$

Vu la remarque 4.20, en dernière étape, si l'on prend $h = f_{\ell,0}$, $\deg h \leq d$ et $h \equiv \prod_{j=0}^{k-1} f_j \pmod{M_n}$.

1. Pour la taille binaire de h : appliquons le lemme 4.23 sur la formule $f_{c,j} = \mathrm{Abr}(f_{c-1,2j} \cdot f_{c-1,2j+1})$ on a :

$$\tau_{\mathcal{T}}(f_{c,j}) \leq \tau_{\mathcal{T}}(f_{c-1,2j}) + \tau_{\mathcal{T}}(f_{c-1,2j+1}) + \log_2 n + 1,$$

en posant $\tau_c = \max_j\{\tau_{\mathcal{T}}(f_{c,j})\}$, cela nous donne :

$$\tau_c \leq 2\tau_{c-1} + \log_2 n + 1 \text{ ou bien } (\tau_c + \log_2 n + 1) \leq 2(\tau_{c-1} + \log_2 n + 1).$$

Par l'induction on a :

$$\tau_c + \log_2 n + 1 \leq 2^c(\tau_0 + \log_2 n + 1) \leq 2^c(\tau + \log_2 n + 1) \Rightarrow \tau_c \leq 2^c\tau + (2^c - 1)(\log_2 n + 1)\ \forall c.$$

En particulier $\tau_{\mathcal{T}}(h) \leq \tau_\ell \leq k\tau + (k-1)(\log_2 n + 1) = k\tau + \mathcal{O}(k \log_2 n)$.

2. Maintenant nous évaluons la complexité. Pour chaque $c = \ldots 1, \ldots, \ell$, on fait $2^{\ell-c}$ calculs $f_{c,j} = \mathrm{Abr}(f_{c-1,2j} \cdot f_{c-1,2j+1})$ où les entrées sont de taille :

$$\deg f_{c-1,2j}, \deg f_{c-1,2j+1} \leq d \quad ; \quad \tau_{\mathcal{T}}(f_{c-1,2j}), \tau_{\mathcal{T}}(f_{c-1,2j+1}) \leq \tau_{c-1} = 2^c\tau + \mathcal{O}(2^c \log_2 n).$$

Vu le lemme 4.23 ce c-ième "étage" du calcul utilise :

$$\begin{aligned} C_B^{(c)} &= 2^{\ell-c}\widetilde{\mathcal{O}}(n\tau_{c-1}) = 2^{\ell-c}\widetilde{\mathcal{O}}(n(2^c\tau + \mathcal{O}(2^c \log_2 n))) \\ &= \widetilde{\mathcal{O}}(n2^\ell\tau) + \widetilde{\mathcal{O}}(n2^\ell \log_2 n) \text{ opérations binaires.} \end{aligned}$$

En résumé, car $\ell = \log_2 k$, la complexité binaire de tout le calcul est :

$$C_B = \sum_{c=1}^{\ell} C_B^{(c)} = \widetilde{\mathcal{O}}(kn\tau).$$

□

Remarque 4.25. Dans $\mathbb{Z}[x]$, le calcul du produit $\prod_{j=0}^{k-1} f_j$ utilise $\widetilde{\mathcal{O}}(k^2 d\tau)$ opérations binaires. Ici sur le même produit mais dans $\mathbb{Z}[x]/\langle M_n(x)\rangle$, on peut remplacer le facteur k^2 par k dans la complexité du calcul.

4.3 Le polynôme minimal d'un élément de $\mathbb{Z}[2\cos\frac{\pi}{n}]$

Par rapport aux informations présentées sur M_n dans le chapitre 1, nous considérons ici une question plus générale : comment est-ce qu'il est possible de construire le polynôme minimal d'une combinaison linéaire quelconque de cosinus d'angles $\frac{k\pi}{n}$?

4.3.1 Définitions

Définition 4.26 (Polynôme minimal M_F)**.** Soit $f \in \mathbb{Z}[x]$ donné par sa forme de Chebyshev, $\deg f = d$, $F = f(2\cos\frac{\pi}{n})$. Alors le polynôme minimal de F dans $\mathbb{Q}[z]$ de F est dénoté par M_F.

Décrivons d'abord M_F algébriquement :

L'annulateur de F

Effectivement, un polynôme dans $\mathbb{Z}[z]$ qui s'annule en F peut être construit par une idée naturelle : Si l'on prend les valeurs du polynôme f en toutes les racines de M_n, elles forment l'ensemble de zéros d'un polynôme unitaire :

$$P_f(z) = \prod_{M(\gamma)=0} (z - f(\gamma)). \tag{4.12}$$

Substituons $B \leftarrow f(x)$, $A \leftarrow M_n(x)$ à l'article 2 du théorème 4.13 on obtient :

$$P_f(z) = \mathrm{Res}_x(z - f(x), M_n(x)). \tag{4.13}$$

Étant déterminant d'une matrice polynomiale à laquelle tous les termes appartiennent $\mathbb{Z}[z]$, alors $P_f \in \mathbb{Z}[z]$.

Relation entre M_F et P_f

Parce que $P_f \in \mathbb{Z}[z]$, $P_f(F) = 0$ alors $M_F | P_f$. Le lemme de Gauss nous permet de conclure que $M_F \in \mathbb{Z}[z]$.

Grâce à la particularité de l'ensemble de racines de M_n, on peut aller plus loin :

Proposition 4.27. *Il existe $\nu \in \mathbb{N}_{>0}$ tel que*

$$P_f = M_F^{\nu}; \tag{4.14}$$

et donc

$$M_F = \frac{P_f}{\gcd(P_f, P_f')}. \tag{4.15}$$

Démonstration. Il suffit de démontrer la formule (4.14) parce que la relation (4.15) est clairement sa conséquence.

D'abord, $P_f \in \mathbb{Z}[z]$, $P_f(F) = 0$ alors $M_F \mid P_f$, selon le lemme de Gauss, $M_F \in \mathbb{Z}[z]$.

Pour toute racine γ_k de M_n, dénotons M_{F_k} son polynôme minimal, nous avons aussi que $M_{F_k} \in \mathbb{Z}[x]$.

$$M_{F_k}(F_k) = 0 \Leftrightarrow M_{F_k}(f(\gamma_k)) = 0 \Leftrightarrow (M_{F_k} \circ f)(\gamma_k) = 0.$$

Mais le polynôme minimal de γ_k est M_n, donc $M_n \mid M_{F_k} \circ f$; γ_1 est zéro de M_n cela implique que γ_1 est aussi zéro de $M_{F_k} \circ f$.

$$(M_{F_k} \circ f)(\gamma_1) = M_{F_k}(f(\gamma_1)) = M_{F_k}(F) = 0,$$

F est racine de M_{F_k}, donc $M_F \mid M_{F_k}$. Les deux sont polynômes minimaux, ils doivent être identiques.

Toutes les racines de P_f ont un unique polynôme minimal M_f, alors on confirme l'identité (4.14). □

Exemple 4.28. Considérons $F = 2\cos\frac{\pi}{7} + 4\cos\frac{\pi}{5} = (T_5 + 2T_7)(2\cos\frac{\pi}{35})$. L'ensemble de racine de P_f est :

$$\mathcal{Z}(P_f) = \{f(2\cos\tfrac{k\pi}{35}), k = 1, 3, 9, 11, 13, 17, 19, 23, 27, 29, 31, 33\}.$$

On n'a que l'indice $k = 29$ satisfait $f(\gamma_{29}) = 2\cos\frac{29\pi}{7} + 4\cos\frac{29\pi}{5} = F$ alors $\nu = 2$, $N_0 = \frac{N}{\nu} = 6$.

Par l'application $\gamma \mapsto T_{29}(\gamma)$ on obtient $f(\gamma_3) = f(\gamma_{17})$, $f(\gamma_9) = f(\gamma_{19})$, $f(\gamma_{11}) = f(\gamma_{31})$, $f(\gamma_{13}) = f(\gamma_{27})$, $f(\gamma_{23}) = f(\gamma_{23})$, donc

$$M_F(z) = (z - f(\gamma_1))(z - f(\gamma_3))(z - f(\gamma_9))(z - f(\gamma_{11}))(z - f(\gamma_{13}))(z - f(\gamma_{23})).$$

4.3.2 Calcul de polynôme annulateur P_f

Nous savons que $P_f(z) = \mathrm{Res}_x(z - f(x), M_n(x))$, que nous pouvons calculer en utilisant des algorithmes généraux [GG13, Section 11.2]. Cela utiliserait $\widetilde{\mathcal{O}}(\delta\tau_s)$ opérations binaires où $\delta = \deg(f) + \deg(M_n)$ et τ_s est un majorant de la somme de tailles binaires des sous-résultants principaux, qui sont des polynômes de degré $\mathcal{O}(\delta)$ en z et les coefficient de taille $\widetilde{\mathcal{O}}(\delta\tau(f))$ (voir la proposition 8.50 dans [BPR06]).

Dans notre cas, cet algorithme nécessiterait $\widetilde{\mathcal{O}}(n^3\tau(f))$ opérations binaires (et même plus en effectuant le changement de base). Vu la proposition 2.37, $\tau(f) = \tau_{\mathcal{T}}(f) + \mathcal{O}(d) = \tau + \mathcal{O}(n)$, la dernière complexité est donc $\widetilde{\mathcal{O}}(n^4 + n^3\tau)$. De plus, il est basé sur l'algorithme *Half-Gcd*, qui n'est efficace qu'à partir de degrés élevés.

Nous proposons ici une méthode simple, qui ne prétend pas d'être la meilleure nécessairement et qui est efficace dès les petits degrés. Elle est basée sur le calcul des sommes de Newton de P_f, considéré comme polynôme à coefficients dans $\mathbb{Z}[x]/(M_n)$.

Nous essayons de déterminer les sommes de Newton de P_f, comme dans le cas de Φ_n et M_n.

Calculer les $S_m(P_f)$

Considérons la i-ième somme de Newton de P_f :

$$S_i(P_f) = \sum_{\substack{1\le k\le n\\(k,2n)=1}} f(T_k(x))^i.$$

Vu la proposition 4.24 on a :

Remarque 4.29. 1. Pour tout $i = 1, \dots, \deg P_f$, il existe une forme de Chebyshev $f^{(i)}$ ($\deg f^{(i)} \le d$) qu'on peut calculer en temps de calcul $\widetilde{\mathcal{O}}(in\tau)$ tel que :

$$\begin{cases} \tau_{\mathcal{T}}(f^{(i)}) & = i\tau + \mathcal{O}(i\log_2 n) \\ f^{(i)} & \equiv f^i(x) \pmod{M_n}. \end{cases} \tag{4.16}$$

2. Le calcul de tous les $f^{(i)}, i = 1, \dots, \deg P_f$ peut être complété en utilisant :

$$\sum_{i=1}^{\deg P_f} \widetilde{\mathcal{O}}(in\tau) = \widetilde{\mathcal{O}}\Big(\deg^2 P_f n\tau\Big) = \widetilde{\mathcal{O}}(n^3\tau).$$

Ecrivons $f^{(i)} = f(x)^i = f_{i,0} + \sum_{j=1}^d f_{i,j}T_j$ dans $\mathbb{Z}[x]/(M_n)$, on obtient :

$$f(T_k(x))^i \equiv f_{i,0} + \sum_{j=1}^d f_{i,j}T_{jk} \pmod{M_n}.$$

Ainsi :

$$S_i(P_f) \equiv \deg P_f f_{i,0} + \sum_{j=1}^d f_{i,j}\Big(\sum_{\substack{1\le k\le n\\(k,2n)=1}} T_{jk}\Big) \pmod{M_n} \tag{4.17}$$

En outre, chaque somme intérieure peut être déterminée par une somme de Newton de Φ_{2n} : On sait que (théorème de Hölder, la formule (1.19), page 24) une somme de Newton de Φ_{2n} est identique la somme de Ramanujan :

$$S_j(\Phi_{2n}) = \sum_{\substack{1\le k\le 2n\\(k,2n)=1}} e^{jk\imath\frac{\pi}{n}} \in \mathbb{Z}$$

$$\Rightarrow 2S_j(\Phi_{2n}) = 2\mathrm{Re}\Big(\sum_{\substack{1\le k\le n\\(k,2n)=1}} e^{jki\frac{\pi}{n}}\Big) = \sum_{\substack{1\le k\le n\\(k,2n)=1}} T_{jk}(2\cos\tfrac{\pi}{n}),$$

cela implique que :

$$\sum_{\substack{1\le k\le n\\(k,2n)=1}} T_{jk} \equiv 2S_j(\Phi_{2n}) \pmod{M_n}. \tag{4.18}$$

D'après deux formules (4.17) et (4.18), les sommes de Newton $S_j(P_f)$ peuvent être obtenues directement depuis les coefficients $f_{i,j}$ de $f(x)^j \pmod{M_n}$ par la formule :

$$S_i(P_f) = f_{i,0}\deg P_f + 2\sum_{j=1}^{d} f_{i,j}S_j(\Phi_{2n}) \tag{4.19}$$

Remarque 4.30. L'identité (4.19) fournit une relation linaire entre les sommes de Newton de Φ_{2n} et celles de P_f, la matrice d'association est $(f_{i,j}) = [\mathcal{I}]_j(f^i)$. Remarquons que la famille $(f_{i,j})$ n'est pas unique mais dans la formule (4.19), les $S_m(\Phi_n)$ sont des invariants.

Nous construisons l'algorithme 4.31 permettant de calculer P_f :

Algorithme 4.31 : Calculer P_f

Entrées : $[\mathcal{T}]f = f_0 + \sum_{j=1}^{d} f_iT_i$
Sorties : $P_f = \mathrm{Res}_x(z - f(x), M_n(x))$

1 **begin**
2 **pour** $j \leftarrow 1$ **a** $\deg P_f$ **faire**
3 Calculer $S_j(\Phi_{2n})$
`/* Utiliser le théorème de Hölder (théorème 1.20) */`
4 **pour** $i \leftarrow 2$ **a** $\deg P_f$ **faire**
5 Calculer les $f^{(i)}$ mentionnés dans la remarque 4.29 ;
`/* Dénotons` $f^{(i)} = f_{i,0} + \sum_{j=1}^{d} f_{i,j}T_j$ `*/`
6 **pour** $i \leftarrow 1$ **a** $\deg P_f$ **faire**
7 $S_i(P_f) = f_{i,0}\cdot\deg P_f + 2\sum_{j=1}^{d} f_{i,j}S_j(\Phi_{2n})$
`/* Les sommes de Newton de` P_f `*/`
8 $p_1 \leftarrow -S_1(P_f)$;
9 **pour** $i \leftarrow 2$ **a** $\deg P_f$ **faire**
10 $p_j = \frac{1}{i}[S_1(P_f)p_{i-1} + \ldots + S_{i-1}(P_f)p_1 + S_i(P_f)]$
11 **retourner** $z^{\deg P_f} + \sum_{i=1}^{\deg P_f} p_i(P_f)z^{\deg P_f - i}$

Preuve de correction. L'algorithme est bien correct grâce à la relation (4.19) et les formules de Newton (1.17). □

Proposition 4.32. *Soit n un nombre entier positif, $d = \lfloor\frac{n}{2}\rfloor$, $f = f_0 + \sum_{j=1}^{d} f_jT_j$ une forme de Chebyshev avec les coefficients entiers de taille binaire au plus τ. Alors on peut calculer le polynôme P_f en exécutant $\widetilde{\mathcal{O}}(n^3\tau)$ opérations binaires.*

Démonstration. Nous utilisons l'algorithme 4.31 :

1. D'abord, le calcul de tous les $S_j(\Phi_{2n})$, $j = 1, \ldots, \deg P_f$ utilise $C_B^{(1)} = \widetilde{\mathcal{O}}(2n) = \widetilde{\mathcal{O}}(n)$ opérations binaires (voir les étapes 2 - 3 dans la preuve de la proposition 3.11) ;

$\rightsquigarrow$ Vu l'identité (1.20), on a $S_j(\Phi_{2n}) \leq \varphi(2n)$ ou $\tau(S_j(\Phi_{2n})) \leq 1 + \log_2 n$ pour tout $j = 1, \ldots, \deg P_f$;

2. Appliquons la remarque 4.29, calculer tous les $f^{(i)}, i = 1, \ldots, \deg P_f$ utilise $C_B^{(2)} = \widetilde{\mathcal{O}}(n^3\tau)$ opérations binaires ;

$\rightsquigarrow$ $\tau_{\mathcal{T}}(f^{(i)}) = i\tau + \mathcal{O}(i \log_2 n)$ pour tout $i = 1, \ldots, \deg P_f$;

3. Une fois que les $f_{j,i}$ sont prêts, la relation (4.19) permet de calculer les sommes de Newton de P_f par $\deg P_f.(2d+2) = \mathcal{O}(n^2)$ opérations arithmétiques dont la taille maximale d'entrées est connue : $\tau(f_{j,i}) = j\tau + \mathcal{O}(j \log_2 n) = \mathcal{O}(n\tau)$, $\tau(S_j(\Phi_{2n})) < 1 + \log_2 n$. Donc le calcul de tous les $S_j(P_f), j = 1, \ldots, \deg P_f$ via l'identité (4.19) utilise :

$$C_B^{(3)} = \mathcal{O}(\deg^2 P_f) \times \mathcal{O}(n\tau) = \mathcal{O}(n^3\tau) \text{ opérations binaires;}$$

$\rightsquigarrow$ La taille des $S_i(P_f)$, $i = 1, \ldots, \deg P_f$ peut être bornée par :

$$\|S_i(P_f)\|_\infty \leq \deg P_f \|f^{(i)}\|_\infty \max\{|S_j(\Phi_{2n})|\}$$

c'est-à-dire que :

$$\tau(S_i(P_f)) \leq \tau_{\mathcal{T}}(f^{(i)}) + 2\log_2 n = i\tau + \mathcal{O}(i \log_2 n).$$

4. Maintenant on a les $S_j(P_f), j = 1, \ldots, \deg P_f$, les formules de Newton (formule (1.17b)) permet de calculer tous les coefficients de P_f par $\mathcal{O}(\deg^2 P_f) = \mathcal{O}(n^2)$ opérations arithmétiques sur les données $\{1, \ldots, \deg P_f\}, \{S_j(P_f)\}$. Vu la taille des $S_j(P_f)$ qui vient être évaluée, pour renvoyer P_f, on utilise :

$$C_B^{(4)} = \mathcal{O}(n^2)\mathcal{O}(n\tau) = \mathcal{O}(n^3\tau) \text{ opérations binaires.}$$

En résumé, la complexité du calcul de P_f par l'algorithme 4.31 est :

$$C_B = \sum_{i=1}^{4} C_B^{(i)} = \widetilde{\mathcal{O}}(n^3\tau).$$

□

4.3.3 Complexité du calcul de M_F

Maintenant P_f est obtenu par $\widetilde{\mathcal{O}}(n^3\tau)$ opérations binaires. On calcule ensuite M_F comme le quotient $\frac{P_f}{\mathrm{pgcd}(P_f, P_f')}$. Ici $\deg P_f = N$, $\tau(P_f) = \mathcal{O}(n\tau)$.

Corollaire 4.33. *Soit n un nombre entier positif, $d = \lfloor \frac{n}{2} \rfloor$, $f = f_0 + \sum_{j=1}^{d} f_j T_j$ une forme de Chebyshev avec les coefficients entiers de taille binaire au plus τ. Alors on peut calculer le polynôme minimal dans $\mathbb{Q}[z]$ de $f(2\cos\pi/n)$ en exécutant $\widetilde{\mathcal{O}}(n^3\tau)$ opérations binaires.*

Démonstration. Lorsque P_f est calculé, on peut avoir P_f' par sa définition en

$$C_B^{(1)} = \widetilde{\mathcal{O}}(\deg P_f \cdot \tau_{P_f}) = \widetilde{\mathcal{O}}(n^2\tau) \text{ opérations binaires.}$$

Il est clair que $\deg(P_f') = N - 1$, $\tau_{P_f'} \leq \tau_{P_f} + \log_2 N = \mathcal{O}(n\tau)$

Pour le calcul du pgcd, on applique le coût mentionné dans la proposition 2.22, donc le calcul de $h = \mathrm{pgcd}(P_f, P_f')$ utilise

$$C_B^{(2)} = \widetilde{\mathcal{O}}(\deg^2 P_f \max\{\tau_{P_f}, \tau_{P_f'}\}) = \widetilde{\mathcal{O}}(n^3\tau)$$

opérations binaires ; h étant sortie du calcul du pgcd, donc $\tau(h) = \mathcal{O}(n\tau)$.

Vu le coût de la division exacte énoncé dans le corollaire 2.26, le quotient $M_f = \frac{P_f}{h}$ est calculé en exécutant

$$C_B^{(3)} = \widetilde{\mathcal{O}}(\deg^2 P_f + \deg P_f \max\{\tau_{P_f}, \tau_h\}) = \widetilde{\mathcal{O}}(n^2\tau)$$

opérations binaires.

En résumé, tout le calcul de M_F s'est fait en complexité binaire

$$C_B = C_B^{(1)} + C_B^{(2)} + C_B^{(3)} = \widetilde{\mathcal{O}}(n^3\tau).$$

□

Exemple 4.34. Considérons $F = 2a\cos\frac{\pi}{7} + 2b\cos\frac{\pi}{5}$, on réécrit

$$F = f(2\cos\tfrac{\pi}{35}),$$

où $f = aT_5 + bT_7$.

Le polynôme minimal de $2\cos\frac{\pi}{35}$ est

$$\begin{aligned} M_{35} &= T_{12} + T_{11} - T_7 - T_6 - T_5 - T_4 + T_2 + T_1 + 1 \\ &= x^{12} + x^{11} - 12x^{10} - 11x^9 + 54x^8 + 43x^7 \\ &\quad - 113x^6 - 71x^5 + 110x^4 + 46x^3 - 40x^2 - 8x + 1. \end{aligned} \tag{4.20}$$

Par la formule $P_f = \mathrm{Res}(z - aT_5 - bT_7, M_{35})$ on obtient $P_f = M_F^2$ avec

$$\begin{aligned} M_F = {} & z^6 - (3\,a + 2\,b)\,z^5 + b\,(-3\,b + 5\,a)\,z^4 + \left(5\,a^3 + 6\,ab^2 + 6\,b^3\right) z^3 \\ & - b\left(5\,a^3 + 7\,a^2b + 9\,ab^2 - 2\,b^3\right) z^2 - \left(3\,a^5 - 4\,a^3b^2 - 7\,a^2b^3 + 2\,ab^4 + 4\,b^5\right) z \\ & - a^6 + a^5b + 7\,b^2a^4 - 2\,a^3b^3 - 7\,a^2b^4 + 2\,ab^5 + b^6. \end{aligned} \tag{4.21}$$

Conclusion du chapitre 4

1. Nous avons développé une méthode composite algébrique-numérique qui permet de calculer le signe de $F = f(2\cos\frac{\pi}{n})$ avec une bonne complexité $\widetilde{\mathcal{O}}(n^2\tau)$ qui est meilleure que la méthode algébrique utilisant l'isolation rapide (section 4.1) ;
2. Par une méthode simple, classique n'utilisant que les sommes de Newton, à l'aide des opérations dans $\mathbb{Z}[x]/\langle M_n\rangle$ (section 4.3), le polynôme unitaire de F peut être calculé en complexité $\widetilde{\mathcal{O}}(n^3\tau)$ qui est actuellement la meilleure complexité connue.

Chapitre 5

Applications aux diagrammes de nœuds de Chebyshev

Sommaire

5.1 Introduction **88**

5.2 Calcul du polynôme caractéristique **90**

5.2.1 Factorisation de $R_{a,b,c}$ 91

5.2.2 Calculer $R_{a,b,c}$ comme élément de $\mathbb{Z}[2\cos\frac{\pi}{n}][\phi]$ 93

5.2.3 Calculer $R_{a,b,c}$ par des approximations numériques 95

5.3 Calculer les racines réelles de $R_{a,b,c}$ **96**

5.3.1 Les racines des facteurs $P_{\alpha,\beta,\gamma}$ 97

5.3.2 La borne de séparation de $R_{a,b,c}$ 99

5.3.3 Isoler les valeurs critiques 100

5.4 Calculer les diagrammes des nœuds **101**

Il est connu que tout nœud $K \subset \mathbb{R}^3 \subset \mathbf{S}^3$ est isotope à une courbe polynomiale (voir [Vas90]). C'est un problème difficile de déterminer un triplet (a, b, c) tel qu'il existe une courbe polynomiale de multi-degré (a, b, c) qui paramétrise K.

Il est prouvé dans [KP11] que tous les nœuds $K \subset \mathbb{R}^3 \subset \mathbf{S}^3$ sont des nœuds de Chebyshev, c'est-à-dire qu'il existe des entiers a, b et c où a et b premiers entre-eux, et un nombre réel ϕ tels que K est isotope à la courbe gauche $\mathcal{C}(a, b, c, \phi) : x = T_a(t)$, $y = T_b(t)$, $z = T_c(t + \phi)$. On l'appelle nœud de Chebyshev.

Le but ici est de déterminer le diagramme correspondant à la courbe gauche $\mathcal{C}(a, b, c, \phi)$. Il reste alors le problème de l'identification du nœud par le calcul d'invariants classiques, question difficile que nous n'abordons pas ici, sauf dans le cas des courbes trigonales non singulières, où le calcul de la fraction de Schubert est simple et permet de déterminer le nœud correspondant.

Quand a, b, c sont fixés, a et b premiers entre-eux, on peut décider quels sont les ϕ pour lesquels $\mathcal{C}(a, b, c, \phi)$ est une courbe régulière en $\mathcal{O}(n^2)$ opérations binaires, où $n = abc$. Quand a, b, c sont fixés, on peut lister tous les nœuds de Chebyshev $\mathcal{C}(a, b, c, \phi)$ possibles en $\widetilde{\mathcal{O}}(n^2)$ opérations binaires.

Si l'on se donne un nombre rationnel ϕ de taille binaire τ_ϕ, le problème de l'identification de $\mathcal{C}(a, b, c, \phi)$ est généralement plus compliqué : pour savoir si c'est un nœud, nous utilisons $\widetilde{\mathcal{O}}(n^2 + n\tau_\phi)$ opérations binaires ; Afin de décrire la nature de tous ses croisements, i.e. de connaître son diagramme, nous utilisons $\widetilde{\mathcal{O}}(n^2\tau_\phi)$ opérations binaires.

5.1 Introduction

Dans [KP11], il est prouvé que tous les nœuds $K \subset \mathbb{R}^3 \subset \mathcal{S}^3$ sont des nœuds de Chebyshev, c'est-à-dire qu'il existe des nombres entiers positifs a, b, c, où a et b sont premiers entre-eux, et un nombre réel ϕ tels que K est isotope dans $\mathcal{S}^3$ à la courbe

$$\mathcal{C}(a,b,c,\phi) : x = T_a(t),\ y = T_b(t),\ z = T_c(t+\phi),$$

où T_n indique le n-ième polynôme unitaire de Chebyshev.

Réciproquement, quand la courbe $\mathcal{C}(a,b,c,\phi)$ n'a pas de point multiple, elle définit un nœud polynomial.

Un tel nœud est déterminé par son diagramme, c'est la projection dans $\mathbb{R}^2$ sur laquelle on précise la nature (au-dessus/en dessous) des croisements.

Quand a et b sont premiers entre-eux, la courbe $\mathcal{C}(a,b,c,\phi)$ est singulière si et seulement si elle possède des points doubles.

Considérons les polynômes dans $\mathbb{Q}[s,t,\phi]$:

$$P_n(t,s) = \frac{T_n(t) - T_n(s)}{t-s},\ Q_n(t,s) = \frac{T_n(t+\phi) - T_n(s+\phi)}{t-s}.$$

Disons que $\mathcal{C}(a,b,c,\phi)$ est un nœud si et seulement si l'ensemble

$$\{(s,t) : \exists\phi,\ P_a(s,t) = P_b(s,t) = Q_c(s,t,\phi) = 0\}$$

est vide.

D'autre part, étant donné un diagramme, c'est généralement difficile à trouver le nœud correspondant quand le nombre de croisements dépasse 16. Nous ne discuterons pas cette question.

Diagrammes de Chebyshev

La projection de la courbe gauche $\mathcal{C}(a,b,c,\phi)$ sur le plan Oxy est la *courbe de Chebyshev* :

$$\mathcal{C}(a,b) : x = T_a(t),\ y = T_b(t).$$

Proposition 5.1 ([KP11],[KPR10]-Proposition 2)**.** *Soit a and b deux entiers positifs qui sont premiers entre-eux, alors la courbe de Chebyshev $x = T_a(t), y = T_b(t)$ a $\frac{(a-1)(b-1)}{2}$ points singuliers qui sont tous croisements, paramétrisés par :*

$$t = 2\cos\Big(\frac{i}{a} + \frac{j}{b}\Big)\pi,\ s = 2\cos\Big(\frac{i}{a} - \frac{j}{b}\Big)\pi,$$

où i, j sont entiers positifs tels que $\frac{i}{a} + \frac{j}{b} < 1$.

Les croisements de la courbe de Chebyshev $\mathcal{C}(a,b)$ se trouvent sur les $(b-1)$ lignes verticales $T_b'(x) = 0$ et aussi sur $(a-1)$ lignes horizontales $T_a'(y) = 0$. On peut représenter ce diagramme de $\mathcal{C}(a,b,c,r)$ par une trajectoire de billiard [KP11] qui est un objet combinatoire. Voyons par exemple les trois nœuds $\overline{5}_2 = \mathcal{C}(4,5,7,0)$, $5_2 = \mathcal{C}(5,6,7,0)$, $\overline{4}_1 = \mathcal{C}(3,5,7,0)$, leurs diagrammes et les trajectoires de billiard correspondantes dans la figure 5.1.

Il y a deux possibilités pour chaque croisements : droit ou gauche (voir [Mur96] et la figure 5.2).

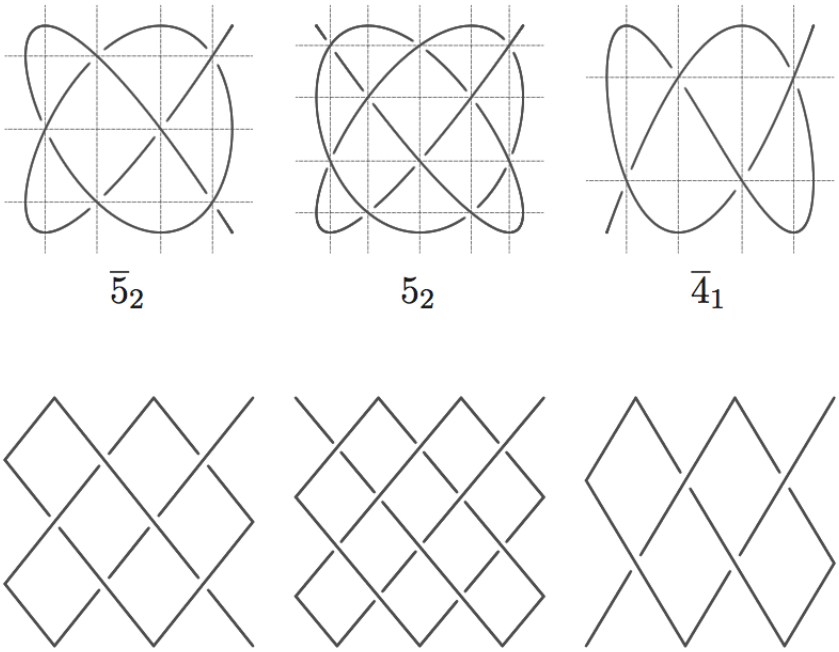


Figure 5.1 – Les nœuds $\overline{5}_2, 5_4, \overline{4}_1$ avec leurs diagrammes de Chebyshev et trajectoires de billiard

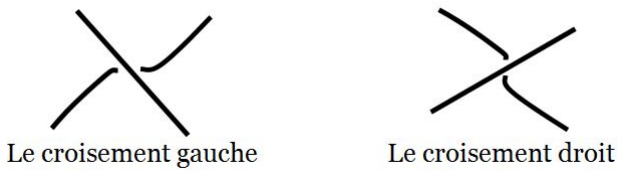


Figure 5.2 – Deux types de croisement

Proposition 5.2 ([KPR10]-Lemme 6)**.** *Considérons le diagramme de la courbe $\mathcal{C}(a,b,c,\phi)$ où a, b et c sont entiers, a et b sont premiers entre-eux. Soit (s,t) les paramètres d'un point double de la projection sur Oxy. Posons $D(s,t,\phi) = Q_c(s,t,\phi)P_{b-a}(s,t)$ alors :*

$$D(s,t,\phi) = (-1)^{i+j+\lfloor\frac{ib}{a}\rfloor+\lfloor\frac{ja}{b}\rfloor}Q_c(s,t,\phi). \tag{5.1}$$

De plus, $D(s,t,\phi) > 0$ si et seulement si le croisement sur le point double $A_{\alpha,\beta}$ de paramètres $(t = 2\cos(\alpha+\beta), s = 2\cos(\alpha-\beta))$ où $\alpha = \frac{i\pi}{a}$, $\beta = \frac{j\pi}{b}$ est un croisement droit.

Le polynôme caractéristique

Si $(a,b) = 1$ alors l'ensemble algébrique

$$\mathcal{V}_{a,b} = \{(s,t) \in \mathbb{C}^2,\ P_a(s,t) = P_b(s,t) = 0\}$$

a exactement $(a-1)(b-1)$ points qui sont tous réels (voir [KP11]).

En considérant Q_c comme un polynôme univarié en ϕ, son coefficient dominant est $\mathrm{lc}_\phi(Q_c) = c$ donc

$$\mathcal{V}_{a,b,c} = \Big\{(s,t,\phi) \in \mathbb{C}^3, P_a(s,t) = P_b(s,t) = Q_c(s,t,\phi)\Big\}$$

est un ensemble fini des points complexes (voir [KPR10, Proposition 5]). La projection

$$\mathcal{Z}_{a,b,c} = \{\phi \in \mathbb{C},\ P_a(s,t) = P_b(s,t) = Q_c(s,t,\phi) = 0\}$$

a aussi un nombre fini des points. Si l'intervalle (ϕ_1, ϕ_2) n'intersecte pas $\mathcal{Z}_{a,b,c}$, alors $\mathcal{C}(a,b,c,\phi_1)$, $\mathcal{C}(a,b,c,\phi_2)$ sont un même nœud, parce que la nature de leurs croisements ne change pas.

Les éléments de $\mathcal{Z}_{a,b,c}$ sont des nombres pour lesquels $\mathcal{C}(a,b,c,\phi)$ est singulière. Ces *valeurs critiques* sont les zéros d'un polynôme $\tilde{R}_{a,b,c}$ qui peut être considéré comme le polynôme caractéristique de ϕ dans $\mathbb{Q}[s,t,\phi]/\langle P_a, P_b, Q_c\rangle$. Signalons que $\widetilde{R}_{a,b,c}$ satisfait $\langle \tilde{R}_{a,b,c}\rangle = \langle P_a, P_b, Q_c\rangle \cap \mathbb{Q}[\phi]$, on pourrait le calculer en utilisant un outil d'élimination classique (voir [KPR10]), e.g. en calculant une base de Gröbner selon un ordre d'élimination $<$ quelconque pourvu que $\phi < s, t$.

La multiplicité des racines de $\widetilde{R}_{a,b,c}$ ne présente pas d'intérêt particulier. $\widetilde{R}_{a,b,c}$ peut être remplacé par un polynôme $R_{a,b,c}$ de coefficients entiers qui possède les mêmes racines, s'il existe. Par abus de la langue nous disons un tel polynôme un *polynôme caractéristique*.

Motivation

Notre premier objectif est : étant donnés a, b, c entiers, a, b premiers entre-eux et un nombre réel ϕ

† décider si $\mathcal{C}(a,b,c,\phi)$ est singulière ;

† sinon, déterminer son diagramme, i.e. le signe des $D(s,t,\phi) = Q_c(s,t,\phi)P_{b-a}(s,t)$ pour tout $(s,t) \in \mathcal{V}_{a,b}$.

Notre deuxième objectif est : étant donnés a, b, c entiers, a, b sont premiers entre-eux,

‡ calculer un polynôme caractéristique $R_{a,b,c}(\phi)$ possédant les mêmes racines avec $\tilde{R}_{a,b,c}$ qui satisfait $\langle R_{a,b,c}\rangle = \langle P_a, P_b, Q_c\rangle \cap \mathbb{Q}[\phi]$;

‡ calculer ses racines réelles $\phi_1 < \cdots < \phi_s$;

‡ calculer les diagrammes $\mathcal{C}(a,b,c,r_i)$ pour des nombres rationnels $r_0 < \phi_1 < \cdots < \phi_s < r_s$.

Dans [KPR10], il est restreint dans le cas $3 \leq a \leq 4$ où l'on obtient des *nœuds à deux ponts*. La raison était qu'en ce cas, on pouvait identifier facilement un nœud à partir de son diagramme en calculant sa fraction de Schubert (voir [Mur96]). Il était également possible d'obtenir directement le polynôme caractéristique comme produit de résultants à coefficients entiers.

Cette méthode était basée essentiellement sur une boîte noire permettant de résoudre le système de dimension zéro $\{P_a(s,t) = P_b(s,t) = Q_c(s,t,\phi) = T - D(s,t,\phi) = 0\}$, par exemple, en calculant une Représentation Rationnelle Univariée (voir [Rou99]), puis de calculer le signe de T en chaque zéro.

Ici, nous ne gardons plus la condition $a \in \{3,4\}$. Nous allons utiliser quelques propriétés des courbes de Chebyshev implicites afin de factoriser le problème sur le corps d'extension $\mathbb{Q}[2\cos\frac{\pi}{n}]$ permettant de changer entièrement la stratégie du calcul : on traite des polynômes de petits degrés avec les coefficients dans des corps d'extension de hauts degrés.

5.2 Calcul du polynôme caractéristique

Comme on a dit, $R_{a,b,c}$ peut être considéré comme générateur de l'idéal principal $\langle P_a, P_b, Q_c\rangle\mathbb{Q}[\phi]$, on pourra l'obtenir donc à partir d'une base de Gröebner de $\langle P_a, P_b, Q_c\rangle$

selon un ordre polynomial quelconque tel que $\phi < s, t$.

Une telle méthode peut être optimisée en utilisant la structure du système $P_a, P_b \in \mathbb{Q}[s,t]$, $Q_c \in \mathbb{Q}[s,t,\phi]$ et en plus le coefficient dominant en ϕ est entier ; on peut aussi calculer premièrement une base de Gröbner G_{ab} de l'idéal $\langle P_a, P_b\rangle$ selon un ordre $<_{ab}$ arbitraire, puis on obtient immédiatement une base de Gröbner G_{abc} de $\langle P_a, P_b, Q_c\rangle$ selon tous les ordres compatibles avec $<_{ab}$ tel que $s, t < \phi$, justement en ajoutant Q_c à G_{ab}.

Dans cette section, nous étudions deux méthodes spécifiques de calcul de $R_{a,b,c}$:
- La première technique utilise $\widetilde{\mathcal{O}}(n^4)$ opérations binaires, qui est basée sur les calculs avec les formes de Chebyshev développés dans les parties précédentes de la thèse ;
- L'autre méthode qui est basée sur les calculs approchés des sommes de cosinus, utilise $\widetilde{\mathcal{O}}(n^3)$ opérations binaires.

Proposition 5.3. *Soit a, b deux entiers positifs qui sont premiers entre-eux où a est impair, soit c un entier positif. Considérons le polynôme*

$$R_{a,b,c} = \prod_{i=1}^{\frac{a-1}{2}} \prod_{j=1}^{b-1} Q_c\Big(2\cos(\tfrac{i\pi}{a} + \tfrac{j\pi}{b}), 2\cos(\tfrac{i\pi}{a} - \tfrac{j\pi}{b}), \phi\Big).$$

Alors $R_{a,b,c} \in \mathbb{Z}[\phi]$;

En plus $\mathcal{C}(a,b,c,\phi)$ est singulière si et seulement si $R_{a,b,c} = 0$.

Démonstration. La courbe $\mathcal{C}(a,b,c,\phi)$ est singulière si et seulement si elle possède des points doubles. Cette condition implique qu'il existe $t = 2\cos(\frac{j\pi}{b} + \frac{i\pi}{a})$, $s = 2\cos(\frac{j\pi}{b} - \frac{i\pi}{a})$ tels que $Q_c(s,t,\phi) = 0$, où $1 \le i \le \frac{a-1}{2}$, $1 \le j \le b-1$. Grâce à la proposition 5.1, $\mathcal{C}(a,b,c,\phi)$ est singulière si et seulement si ϕ est racine de

$$R_{a,b,c} = \prod_{i=1}^{\frac{a-1}{2}} \prod_{j=1}^{b-1} Q_c\Big(2\cos(\tfrac{i\pi}{a} + \tfrac{j\pi}{b}), 2\cos(\tfrac{i\pi}{a} - \tfrac{j\pi}{b}), \phi\Big).$$

$Q_c(s,t,\phi)$ est un polynôme symmétrique dans $\mathbb{Z}[\phi][t,s]$. En posant $\alpha_i = \frac{i\pi}{a}$, $\beta_j = \frac{j\pi}{b}$, $s = 2\cos(\alpha_i+\beta_j)$, $t = 2\cos(\alpha_i-\beta_j)$, parce que $s+t = 4\cos\alpha_i\cos\beta_j$ et $st = 2\cos 2\alpha_i + 2\cos 2\beta_j$, on a $Q_c(s,t,\phi) \in \mathbb{Z}[\phi, 2\cos\alpha_i][2\cos\beta_j]$. D'autre part, $2\cos\beta_j$ est racine de $U_b \in \mathbb{Z}[t]$, donc

$$R_i = \prod_{j=1}^{b-1} Q_c(2\cos(\alpha_i+\beta_j), \cos 2(\alpha_i - \beta_j), \phi) \tag{5.2}$$

appartient à $\mathbb{Z}[\phi, 2\cos\alpha_i]$.

Par définition, $Q_c(-s,-t,-\phi) = (-1)^{c-1}Q_c(s,t,\phi)$. On en déduit

$$\prod_{i=1}^{\frac{a-1}{2}} R_i(-\phi)R_i(\phi) = \pm \prod_{i=1}^{a-1} R_i(\phi) \in \mathbb{Z}[\phi].$$

Enfin $R_{a,b,c}^2 \in \mathbb{Z}[\phi]$ alors $R_{a,b,c} \in \mathbb{Z}[\phi]$. □

5.2.1 Factorisation de $R_{a,b,c}$

Le polynôme $P_n(t,s)$ se factorise en polynômes de degré 1 ou 2, ce qui permet d'obtenir la factorisation de $Q_c(2\cos(\alpha+\beta), 2\cos(\alpha-\beta), \phi)$.

Lemme 5.4. *Soit $\eta \in \mathbb{R}$, en dénotant*

$$E_\eta(t,s) = \begin{cases} t-s & si\ \eta = 0, \\ t+s & si\ \eta = \pi, \\ t^2+s^2-2\cos(\eta)ts - 4\sin^2(\eta) & si\ \eta \not\equiv 0 \pmod{\pi}, \end{cases} \tag{5.3}$$

alors :

$$\frac{T_n(t)-T_n(s)}{t-s} = \prod_{k=1}^{\lfloor n/2 \rfloor} E_{\frac{2k\pi}{n}}(t,s) \tag{5.4}$$

Démonstration. D'abord $\deg_t(T_n(t)-T_n(s)) = \deg_s(T_n(t)-T_n(s)) = n$.

À part de $t-s$, on cherche les autres facteurs de $T_n(t)-T_n(s)$.
Soit $t \in [-2,2]$, posons $t = 2\cos\rho$ alors pour tout $s = 2\cos(\pm\rho + \frac{2k\pi}{n})$, $k = 1,\dots,n-1$ on a $T_n(t)-T_n(s) = 0$. L'équation implicite de la paramétrisation $(t = 2\cos\rho, s = 2\cos(\pm\rho + \frac{k\pi}{n}))$ est

$$E_{\frac{2k\pi}{n}}(t,s) = t^2+s^2-2ts\cos\tfrac{2k\pi}{n} - 4\sin^2\tfrac{2k\pi}{n} = 0,$$

alors $t-s$ et les $E_{\frac{2k\pi}{n}}, k = 1,\dots,\lceil n/2 \rceil$ sont tous facteurs de $T_n(t)-T_n(s)$.

À remarquer que si n est pair, l'ensemble $\{1,\dots,n-1\}$ contient l'élément $\frac{n}{2}$ qui donne $E_\pi(t,s) = t+s$.

La factorisation sera confirmée en comparant le coefficient dominant et le degré : Il est clair que $\mathrm{lc}_t(T_n(t)-T_n(s)) = \mathrm{lc}_t(T_n(t)-T_n(s)) = 1$, $\mathrm{lc}_t E_{\frac{2k\pi}{n}} = \mathrm{lc}_s E_{\frac{2k\pi}{n}} = 1$. De plus $1 + \sum_{k=1}^{\lfloor n/2 \rfloor} \deg_t E_{2\frac{k\pi}{n}} = 1 + \sum_{k=1}^{\lfloor n/2 \rfloor} \deg_s E_{2\frac{k\pi}{n}} = n$. □

Grâce au lemme 5.4, on obtient

$$Q_c(t,s,\phi) = \prod_{k=1}^{\lfloor c/2 \rfloor} E_{2k\pi/c}(t+\phi, s+\phi). \tag{5.5}$$

Définition 5.5. Dénotons $\widetilde{P}_{\alpha,\beta,\gamma}(\phi) = E_{2\gamma}(2\cos(\alpha+\beta)+\phi, 2\cos(\alpha-\beta)+\phi)$, concrètement :

$$\widetilde{P}_{\alpha,\beta,\gamma}(\phi) = \begin{cases} 2\phi + 4\cos\alpha\cos\beta & \text{si } \gamma = \frac{\pi}{2} \\ 4\sin^2\gamma\left(\phi^2 + 4\cos\alpha\cos\beta\phi + 4\frac{(\cos^2\alpha-\cos^2\gamma)(\cos^2\beta-\cos^2\gamma)}{\sin^2\gamma}\right) & \text{sinon} \end{cases} \tag{5.6}$$

On définit aussi :

$$P_{\alpha,\beta,\gamma}(\phi) = \begin{cases} \phi + 2\cos\alpha\cos\beta & \text{si } \gamma = \frac{\pi}{2} \\ \phi^2 + 4\cos\alpha\cos\beta\phi + 4\frac{(\cos^2\alpha-\cos^2\gamma)(\cos^2\beta-\cos^2\gamma)}{\sin^2\gamma} & \text{sinon} \end{cases}. \tag{5.7}$$

Alors on obtient la factorisation :

Proposition 5.6. *Soit a, b deux entiers positifs premiers entre-eux, soit c un entier positif. Alors :*

$$R_{a,b,c}(\phi) = \prod_{i=1}^{\frac{a-1}{2}} \prod_{j=1}^{b-1} \prod_{k=1}^{\lfloor c/2 \rfloor} \widetilde{P}_{\alpha,\beta,\gamma}(\phi) = c^{\frac{(a-1)(b-1)}{2}} \prod_{i=1}^{\frac{a-1}{2}} \prod_{j=1}^{b-1} \prod_{k=1}^{\lfloor c/2 \rfloor} P_{\frac{i\pi}{a},\frac{j\pi}{b},\frac{k\pi}{c}}(\phi). \tag{5.8}$$

Démonstration. Partons de la formule (5.5), en signalant que $\sin^2 \frac{k\pi}{c} = \sin \frac{k\pi}{c} \sin \frac{(c-k)\pi}{c}$ on a :

$$Q_c(2\cos(\alpha+\beta), 2\cos(\alpha-\beta), \phi) = \left(2^{c-1} \prod_{k=1}^{c-1} \sin \frac{k\pi}{c}\right) \prod_{k=1}^{\lfloor c/2 \rfloor} P_{\alpha,\beta,\gamma}(\phi).$$

Mais pour tout x, $\exp(\imath x) = \cos x + \imath \sin x$, donc :

$$\prod_{k=1}^{c-1} \sin \frac{k\pi}{c} = (2\imath)^{1-c} \prod_{k=1}^{c-1} (\exp(\frac{\imath k\pi}{c}) - \exp(\frac{-\imath k\pi}{c})) = (2\imath)^{1-c} \exp(-\frac{\imath\pi c(c-1)}{2c}) \prod_{k=1}^{c-1} (\exp(\frac{2\imath k\pi}{c}) - 1)$$
$$= 2^{1-c} \prod_{k=1}^{c-1} (1-\xi^k), \text{ avec } \xi = \exp(\frac{2\imath\pi}{c})$$

On sait que $x^c - 1 = (x-1)(\sum_{k=0}^{c-1} x^k)$ alors $\prod_{k=1}^{c-1}(x-\xi^k) = \sum_{k=0}^{c-1} x^k$. Particulèrement, en substituant $x = 1$, on obtient $\prod_{k=1}^{c-1}(1-\xi^k) = \sum_{k=0}^{c-1} x^k = c$, donc :

$$\prod_{k=1}^{c-1} \sin \frac{k\pi}{c} = c.2^{1-c},$$

ainsi :

$$Q_c(2\cos(\alpha+\beta), 2\cos(\alpha-\beta), \phi) = c \prod_{k=1}^{\lfloor c/2 \rfloor} P_{\alpha,\beta,\gamma}(\phi), \tag{5.9}$$

ce qui prouve la proposition. □

5.2.2 Calculer $R_{a,b,c}$ comme élément de $\mathbb{Z}[2\cos\frac{\pi}{n}][\phi]$

D'après la formule (5.8), $R_{a,b,c}$ est le produit de $\frac{(a-1)(b-1)}{2} \cdot \lfloor \frac{c}{2} \rfloor$ polynômes de degré 1 ou 2, en variable ϕ, avec les coefficients dans $\mathbb{Z}[2\cos\frac{\pi}{n}]$. Nous pouvons y appliquer les opérations dans $\mathbb{Z}[2\cos\frac{\pi}{n}]$ étudiées dans la sous-section 4.2.2 pour l'obtenir.

Lemme 5.7. *Soit $P = \sum_{i=0}^{D} p_i(2\cos\frac{\pi}{n})\phi^i$, $Q = \sum_{i=0}^{d} q_i(2\cos\frac{\pi}{n})\phi^i$ deux polynômes dans $\mathbb{Z}[2\cos\frac{\pi}{n}][\phi]$. Supposons que les formes de Chebyshev $p_i = \sum_{j=0}^{n-1} p_{i,j}T_j$ et $q_i = \sum_{j=0}^{n-1} q_{i,j}T_j$ satisfassent $\tau_{\mathcal{T}}(p_i) \le \tau$, $\tau_{\mathcal{T}}(q_i) \le \tau' \le \tau$.*

Alors on a $P \cdot Q = \sum_{i=0}^{D+d} h_i(2\cos\frac{\pi}{n})\phi^i$ où $h_i = \sum_{j=0}^{n-1} h_{i,j}T_j$, $\tau_{\mathcal{T}}(h_i) \le \tau + \tau' + \log_2 n + \log_2 d$. On peut calculer toutes les formes de Chebyshev $[\mathcal{T}]h_i$ en $\mathcal{O}(dD)$ opérations arithmétiques de $\mathbb{Z}[2\cos\frac{\pi}{n}]$, ou en $\widetilde{\mathcal{O}}(dDn\tau)$ opérations binaires.

Démonstration. Rappelons d'abord le lemme 4.23, on sait qu'avec $f = f_0 + \sum_{i=1}^{d} f_iT_i, g = g_0 + \sum_{i=1}^{d} g_iT_i \in \mathbb{Z}[x], \tau_{\mathcal{T}}(f), \tau_{\mathcal{T}}(g) \le \tau$. Alors $\tau_{\mathcal{T}}(\mathrm{Abr}([\mathcal{T}]f.g)) \le \tau_{\mathcal{T}}(f) + \tau_{\mathcal{T}}(g) + \log_2 n + 1$; $\mathrm{Abr}([\mathcal{T}]f.g))$ peut être calculée en $\widetilde{\mathcal{O}}(n\tau)$ operations binaires.

Le produit $P \cdot Q$ peut être écrit $P \cdot Q = \sum_{i=0}^{d+D} h_i\phi^i$ avec $h_i = \sum_{j=0}^{d} \mathrm{Abr}(p_{i-j}q_j)$. Chaque forme $\mathrm{Abr}(p_{i-j}q_j)$ peut être calculé en $\widetilde{\mathcal{O}}(n\tau)$ opérations binaires, donc toutes les formes $[\mathcal{T}]h_i$ sont calculées en $\widetilde{\mathcal{O}}(d(d+D)n\tau)$ opérations binaires, en plus $\tau_{\mathcal{T}}(h_i) \le \tau + \tau' + \log_2 n + \log_2 d$. □

On en déduit :

Corollaire 5.8. *Soit $P_i = \sum_{j=0}^{d} p_{i,j}(2\cos\frac{\pi}{n})\phi^j$ des polynômes dans $\mathbb{Z}[2\cos\frac{\pi}{n}][\phi]$ avec $\tau_{\mathcal{T}}(p_{i,j}) \le \tau$. Alors leur produit peut être exprimé par $P = \prod_{i=1}^{k} P_i = \sum_{i=0}^{dk} h_i(2\cos\frac{\pi}{n})\phi^j$ avec $h_i = \sum_{j=0}^{n-1} h_{i,j}T_j$, $\tau_{\mathcal{T}}(h_i) \le k\tau + k\log_2 nd$. On peut calculer P en $\widetilde{\mathcal{O}}(d^2k^3n\tau)$ opérations binaires.*

Démonstration. Posons $Q_i = P_1 \cdots P_i$. D'après le lemme 5.7 :

† $\tau_{\mathcal{T}}(Q_{i+1}) \leq \tau_{\mathcal{T}}(Q_i) + \tau + \log_2 nd \leq (i+1)(\tau + \log_2 nd)$;

† Q_{i+1} peut être calculé à partir de Q_i en $\widetilde{\mathcal{O}}(\tau_{\mathcal{T}}(Q_i)d(id+d)n\tau) = \widetilde{\mathcal{O}}(i^2d^2n\tau)$ opérations binaires.

En résumé, on obtient Q_k en $\widetilde{\mathcal{O}}(k^3d^2n\tau)$ opérations binaires. □

Travaillons dans $\mathbb{Z}[2\cos\frac{\pi}{a}, 2\cos\frac{\pi}{b}, 2\cos\frac{\pi}{c}][\phi] \subset \mathbb{Z}[2\cos\frac{\pi}{n}][\phi]$, le lemme 5.7 peut être appliqué à la formule (5.8), ce qui nous donne la première méthode pour calculer $R_{a,b,c}$:

Proposition 5.9. *Soit a et b premiers entre-eux où a est impair, soit c un entier. Alors $R_{a,b,c}$ peut être calculé comme un élément de $\mathbb{Z}[2\cos\frac{\pi}{n}][\phi]$ en $\mathcal{O}(n^4)$ opérations binaires.*

Démonstration. Nous devons calculer le produit des $\widetilde{P}_{\alpha,\beta,\gamma}(\phi)$. Les coefficients d'un tel polynôme seront écrits sous la forme de Chebyshev : (voir la figure A.13, annexe A.3)

- Quand c est impair, tous les $\widetilde{P}_{\alpha,\beta,\gamma}$ sont de degré 2, chacun peut être réécrit :

$$\widetilde{P}_{\alpha,\beta,\gamma}(\phi) = \tfrac{1}{4}(f_2(2\cos\tfrac{\pi}{n})\phi^2 + f_1(2\cos\tfrac{\pi}{n})\phi + f_0(2\cos\tfrac{\pi}{n})),$$

où

$$\begin{aligned} f_2 &= 2 - T_{2\,kab} \\ f_1 &= 2\,T_{cja-cib} + 2\,T_{cja+cib} - T_{2\,kab+cja-cib} - T_{2\,kab-cja+cib} \\ &\quad - T_{2\,kab-cja-cib} - T_{2\,kab+cja+cib} \\ f_0 &= 2 + T_{2\,cja-2\,cib} + T_{2\,cja+2\,cib} + T_{4\,kab} \\ &\quad - T_{2\,kab-2\,cib} - T_{2\,kab+2\,cib} - T_{2\,kab-2\,cja} - T_{2\,kab+2\,cja} \end{aligned} \tag{5.10}$$

- D'autre part, pour les c qui sont pairs, quand $k = \frac{1}{2}c$, on écrit :

$$\widetilde{P}_{\alpha,\beta,\gamma}(\phi) = \tfrac{1}{2}(g_1(2\cos\tfrac{\pi}{n})\phi + g_0(2\cos\tfrac{\pi}{n}))$$

où

$$g_1 = 2, \quad g_0 = T_{cja-cib} + T_{cja+cib}. \tag{5.11}$$

Comme $T_{n+i} \equiv T_{n-i} \equiv -T_i \pmod{M_n}$, on peut écrire f_0, f_1 and f_2 sous forme de Chebyshev de degré au plus $n-1$, $\tau_{\mathcal{T}}(f_i) \leq 4$; De la même façon, $\tau_{\mathcal{T}}(g_i) \leq 2$.

Appliquons le corollaire 5.8 avec $k = \frac{1}{2}(a-1)(b-1)\lfloor\frac{c-1}{2}\rfloor \leq \frac{n}{4}$, $\tau = 4$:

† $2^{2k}R_{a,b,c}$ est calculé comme élément de $\mathbb{Z}[2\cos\frac{\pi}{n}][\phi]$ en $\widetilde{\mathcal{O}}(n^4)$ opérations binaires.

† chaque coefficient de ce polynôme, $[\phi^j](2^{2k}R_{a,b,c})$ est une forme de Chebyshev :

$$[\phi^j](2^{2k}R_{a,b,c}) = \lambda_j(2\cos\tfrac{\pi}{n}),$$

où $\lambda_j \in \mathbb{Z}[x]$, $\deg\lambda_j \leq n-1$, $\tau_{\mathcal{T}}(\lambda_j) = \widetilde{\mathcal{O}}(n)$.

On est assuré que $\lambda_j(2\cos\frac{\pi}{n}) \in \mathbb{Z}$ alors sa valeur exacte peut être identifiée par $\lambda_j \bmod M_n$. On sait aussi la taille de M_n : $\deg M_n < n$, $\tau_{\mathcal{T}}(M_n) = \mathcal{O}(n)$ (voir le lemme 3.15). En plus, si $\deg f, \deg g \leq d$, $\tau_{\mathcal{T}}(f), \tau_{\mathcal{T}}(g) \leq \tau$, le reste $[\mathcal{T}]\mathrm{Rem}(f,g)$ peut être calculé en $\widetilde{\mathcal{O}}(d^2\tau)$ opérations binaires (voir la proposition 2.21).
Dans notre cas, ce reste, étant la valeur exacte du coefficient, est un entier qui est obtenu en utilisant $\widetilde{\mathcal{O}}(n^3)$ opérations binaires. Il y a $\mathcal{O}(n)$ coefficients à identifier, donc la complexité binaire synthétisée est enfin $\widetilde{\mathcal{O}}(n^4)$. □

5.2.3 Calculer $R_{a,b,c}$ par des approximations numériques

Dans le chapitre 4, on sait calculer approximativement une somme de cosinus à l'aide du corollaire 4.4.

Une fois que le j-ième coefficient de $2^{2k}R_{a,b,c}$ dans la preuve de la proposition 5.9 est calculé en forme $[\phi^j](2^{2k}R_{a,b,c}) = \lambda_j(2\cos\frac{\pi}{n})$, l'une de ses valeurs approximatives à la précision 2^{-1} sera suffisante pour décider sa valeur exacte, vu qu'il est un entier. On sait que $\tau_{\mathcal{T}}(\lambda_j) = \mathcal{O}(n)$, alors cette valeur approximative peut être trouvée en $\widetilde{\mathcal{O}}(n^2)$ opérations binaires.

Cette stratégie nous permet d'arriver à tous les coefficients entiers de λ_j en $\widetilde{\mathcal{O}}(n^3)$ opérations binaires.

Nous améliorons la stratégie en utilisant l'approximation numérique des facteurs $\widetilde{P}_{\alpha,\beta,\gamma}$ dans $\mathbb{Q}[\phi]$.

Démontrons tout d'abord ce lemme :

Lemme 5.10. *Soit $P_i = a_i\phi^2 + b_i\phi + c_i$, $i = 1, \dots, N$, des polynômes dans $\mathbb{R}_{\deg\leq 2}[\phi]$. Soit $\hat{P}_i \in \mathbb{R}_{\deg\leq 2}[\phi]$ tels que*

$$\|P_i\|_\infty \leq M, \quad \|\hat{P}_i - P_i\|_\infty \leq \delta.$$

Alors on a :

$$\|\prod_{i=1}^N P_i - \prod_{i=1}^N \hat{P}_i\|_\infty \leq \delta\, N\, 3^N (M+\delta)^N.$$

Démonstration. Nous allons utiliser le lemme 2.27 concernant le produit de plusieurs polynômes pour déduire que

$$\text{Si } \deg Q \leq 2 \text{ alors } \|PQ\|_\infty \leq 3\|P\|_\infty\|Q\|_\infty.$$

Supposons que $\|\hat{P}_i\|_\infty \leq M + \delta$, alors $\|P_1 \cdots P_k\|_\infty \leq 3^{k-1}M^k$. En plus on va déduire par induction que $\|\hat{P}_1 \cdots \hat{P}_k\|_\infty \leq 3^{k-1}(M+\delta)^k$.

En fait, si $\|P_1 \cdots P_k - \hat{P}_1 \cdots \hat{P}_k\|_\infty \leq \delta_k$, on a :

$$\begin{aligned}
\|P_1 \cdots P_{k+1} - \hat{P}_1 \cdots \hat{P}_{k+1}\|_\infty &= \|P_1 \cdots P_k(P_{k+1} - \hat{P}_{k+1}) + (P_1 \cdots P_k - \hat{P}_1 \cdots \hat{P}_k)\hat{P}_{k+1}\|_\infty \\
&\leq \|P_1 \cdots P_k(P_{k+1} - \hat{P}_{k+1})\|_\infty + \|(P_1 \cdots P_k - \hat{P}_1 \cdots \hat{P}_k)\hat{P}_{k+1}\|_\infty \\
&= 3^k M^k \delta + 3\delta_k(M+\delta)
\end{aligned}$$

Donc $\delta_{k+1} \leq 3(M+\delta)\delta_k + (3M)^k\delta$. Posons $u_k = \frac{\delta_k}{3^k(M+\delta)^k}$ alors :

$$u_{k+1} - u_k \leq \frac{\delta}{3(M+\delta)}\Big(\frac{M}{M+\delta}\Big)^k \leq \delta.$$

On obtient donc $u_{k+1} \leq u_1 + k\delta = (k+1)\delta$. □

On a prouvé aussi la proposition 2.28 qui confirme que : Si $f_0, \dots, f_{k-1} \in \mathbb{Z}[x]$, $\deg f_j \leq d$, $\tau(f_j) \leq \tau$ pour tout $j = 0, \dots, k-1$ alors leur produit $f = \prod_{j=0}^{k-1} f_j$ satisfait : $\tau(f) \leq k\tau + (k-1)\log_2(d+1)$. Il peut être calculé en $\widetilde{\mathcal{O}}(kd)$ opérations arithmétiques, $\widetilde{\mathcal{O}}(k^2 d\tau)$ opérations binaires. Il est clair que cette proposition sera toujours vraie si l'on considère $f_j \in \mathbb{Q}[x]$ avec des coefficients en forme de nombres dyadiques, ce qui, à l'aide du lemme 5.10, permet d'obtenir :

Corollaire 5.11. *Soit $P_i = a_i\phi^2 + b_i\phi + c_i$, $i = 1, \dots, N$ des polynômes dans $\mathbb{Q}_{\deg\leq 2}[\phi]$ avec tous les coefficients en forme de nombres dyadiques, $\tau(P_i) \leq \tau$. Alors $P = \prod_{i=1}^N P_i$ peut être calculé en $\widetilde{\mathcal{O}}(N^2\tau)$ opérations binaires ; De plus $\tau(P) \leq N\tau + (N-1)\log_2 3$.*

Ce corollaire nous amène :

Proposition 5.12. *Soit a et b premiers entre-eux où a est impair, soit c un entier. Alors on peut calculer $R_{a,b,c}$ en $\widetilde{\mathcal{O}}(n^3)$ opérations binaires, où $n = abc$.*

Démonstration. Il est clair que $M = \|\widetilde{P}_{\alpha,\beta,\gamma}\|_\infty \leq 16$. Prenons $N = \frac{1}{2}(a-1)(b-1)\lfloor \frac{c}{2} \rfloor$. On calcule approximativement chaque coefficient de $\widetilde{P}_{\alpha,\beta,\gamma}$ à la précision $\delta = 2^{-6N+1}$ pour obtenir N polynômes $\hat{P}_{\alpha,\beta,\gamma}$ dont les coefficients sont nombres dyadiques bornés par $M+\delta$.

Ensuite nous calculons $\hat{R}_{a,b,c} = \prod \hat{P}_{\alpha,\beta,\gamma}$ dans $\mathbb{Q}[\phi]$ en utilisant $\widetilde{\mathcal{O}}(N^3)$ opérations binaires (c'est la conclusion du lemme 5.11 avec $\tau = 6N+1$).

Par le lemme 5.10 :

$$\|R_{a,b,c} - \hat{R}_{a,b,c}\| \leq 2^{-6N+1} N 3^N \cdot (16+\delta)^N \leq \frac{1}{2}$$

ce qui permet de déterminer immédiatement les valeurs entiers de tous les coefficients de $R_{a,b,c}$. □

5.3 Calculer les racines réelles de $R_{a,b,c}$

Ici, notre objectif est de calculer les racines de $R_{a,b,c}$. Nous donnons d'abord une estimation de la taille des coefficients de $R_{a,b,c}$. Le lemme suivant montre que $\|R_{a,b,c}\|_1 \leq 6^N$, où $N = \frac{1}{2}(a-1)(b-1)(c-1) < n$. Grâce à cette borne, les algorithmes généraux récents peuvent trouver les racines de $R_{a,b,c}$ en $\tilde{\mathcal{O}}(n^3)$ opérations binaires (voir [SM16] par exemple).

Lemme 5.13. *Soit a, b deux entier positifs premiers entre-eux, a est impair et c un entier. Posons $N = \frac{1}{2}(a-1)(b-1)(c-1)$. Alors on a*

$$\|R_{a,b,c}\|_1 = \sum_{i=1}^{N} |a_i| \leq 6^N.$$

Démonstration. Il y a deux cas à considérer dans la formule (5.8)
- Si c est impair, $R_{a,b,c}$ apparait comme le produit

$$R^{(1)}_{a,b,c} = \prod_{i=1}^{\frac{a-1}{2}} \prod_{j=1}^{b-1} \prod_{k=1}^{\frac{c-1}{2}} \widetilde{P}_{\alpha,\beta,\gamma}. \tag{5.12}$$

- Si c est pair, il faut multiplier ce produit par le facteur :

$$R^{(0)}_{a,b,c} = \prod_{i=1}^{\frac{a-1}{2}} \prod_{j=1}^{b-1} (2\phi + 4\cos\alpha\cos\beta). \tag{5.13}$$

Soit $P = \sum_{i=0}^{m} a_i \phi^i \in \mathbb{R}[\phi]$ et $Q = \sum_{i=0}^{m} b_i \phi^i \in \mathbb{R}[\phi]$, on dira que $P \prec Q$ si $|a_i| \leq b_i$, $i = 0, \ldots, m$. Il est clair que :

$$\text{Si } P \prec Q \text{ alors } \|P\|_1 \leq \|Q\|_1,$$

en plus

$$\text{Si } P_1 \prec Q_1 \text{ et } P_2 \prec Q_2, \|P_1 P_2\|_1 \leq \|Q_1 Q_2\|_1.$$

On a : $\phi + 2\cos\frac{i\pi}{a}\cos\frac{j\pi}{b} \prec \phi + 2$, cela nous donne : $R^{(1)}_{a,b,c} \prec (2\phi+4)^{(a-1)(b-1)/2}$.
Si $\gamma = \frac{k\pi}{2} \neq \frac{\pi}{2}$ alors

$$\begin{aligned}\widetilde{P}_{\alpha,\beta,\gamma} &= 4\sin^2\gamma\cdot\phi^2 + 16\phi\cdot\cos\alpha\cos\beta\sin^2\gamma + 16(\cos^2\alpha-\cos^2\gamma)(\cos^2\beta-\cos^2\gamma)\\ &\prec 4(\phi+2)^2 \end{aligned} \tag{5.14}$$

On en déduit que $R^{(0)}_{a,b,c} \prec (2\phi+4)^{2\lfloor\frac{c-1}{2}\rfloor})^{(a-1)(b-1)/2}$ et $R_{a,b,c} \prec (2\phi+4)^N$. Enfin

$$\|R_{a,b,c}\|_1 \leq 2^N\|(\phi+2)^N\|_1 = 6^N.$$

□

Dans la suite, nous montrons que le calcul des racines réelles de $R_{a,b,c}$ peut être effectué en $\widetilde{\mathcal{O}}(n^2)$ opérations binaires.

5.3.1 Les racines des facteurs $P_{\alpha,\beta,\gamma}$

Si $\gamma = \frac{\pi}{2}$, $\deg P_{\alpha,\beta,\gamma} = 1$, il a une racine unique. Sinon il est de degré 2 donc possède au plus deux racines réelles distinctes.

Si $\deg P_{\alpha,\beta,\gamma} = 1$ $(\gamma = \frac{\pi}{2})$

Par définition, $\deg P_{\alpha,\beta,\gamma} = 1 \Leftrightarrow \gamma = \frac{\pi}{2}$, dans le cas $P_{\alpha,\beta,\frac{\pi}{2}}$ a une seule racine

$$\phi = -2\cos\alpha\cos\beta.$$

Si $\deg P_{\alpha,\beta,\gamma} = 2$ $(\gamma \neq \frac{\pi}{2})$

Dans tous les cas restants où $\gamma \neq \frac{\pi}{2}$, $P_{\alpha,\beta,\gamma}$ est de degré 2. Son discriminant est :

$$\Delta_{\alpha,\beta,\gamma} = 16\cot^2\gamma(\sin^2\gamma - \sin^2\alpha\sin^2\beta) \tag{5.15}$$

Nous allons utiliser le résultat ci-dessous :

Lemme 5.14. *[Mye93, Theo. 4] Considérons l'équation :*

$$\sin\pi x_1\sin\pi x_2 = \sin\pi x_3\sin\pi x_4, x_j \in \mathbb{Q}. \tag{5.16}$$

Pour tout ρ on a

$$\sin\tfrac{\pi}{6}\sin\rho = \sin\tfrac{\rho}{2}\sin(\tfrac{\pi}{2}-\tfrac{\rho}{2}),$$

toutes les autres solutions rationnelles (x_1,x_2,x_3,x_4) $(0 < x_1 < x_3 \leq x_4 < x_2)$ de l'équation (5.16) sont :

$$\begin{aligned}\{&(\tfrac{1}{21},\tfrac{8}{21},\tfrac{1}{14},\tfrac{3}{14}),(\tfrac{1}{14},\tfrac{5}{14},\tfrac{2}{21},\tfrac{5}{21}),(\tfrac{4}{21},\tfrac{10}{21},\tfrac{3}{14},\tfrac{5}{14}),(\tfrac{1}{20},\tfrac{9}{20},\tfrac{1}{15},\tfrac{4}{15}),\\ &(\tfrac{2}{15},\tfrac{7}{15},\tfrac{3}{20},\tfrac{7}{20}),(\tfrac{1}{30},\tfrac{3}{10},\tfrac{1}{15},\tfrac{2}{15}),(\tfrac{1}{15},\tfrac{7}{15},\tfrac{1}{10},\tfrac{7}{30}),(\tfrac{1}{10},\tfrac{13}{30},\tfrac{2}{15},\tfrac{4}{15}),\\ &(\tfrac{4}{15},\tfrac{7}{15},\tfrac{3}{10},\tfrac{11}{30}),(\tfrac{1}{30},\tfrac{11}{30},\tfrac{1}{10},\tfrac{1}{10}),(\tfrac{7}{30},\tfrac{13}{30},\tfrac{3}{10},\tfrac{3}{10}),(\tfrac{1}{15},\tfrac{4}{15},\tfrac{1}{10},\tfrac{1}{6}),\\ &(\tfrac{2}{15},\tfrac{8}{15},\tfrac{1}{6},\tfrac{3}{10}),(\tfrac{1}{12},\tfrac{5}{12},\tfrac{1}{10},\tfrac{3}{10}),(\tfrac{1}{10},\tfrac{3}{10},\tfrac{1}{6},\tfrac{1}{6})\}\end{aligned} \tag{5.17}$$

Proposition 5.15. *Soit $\alpha = \frac{i\pi}{a}$, $\beta = \frac{j\pi}{b}$, $\gamma = \frac{k\pi}{c}$ avec $1 \leq i \leq \frac{a-1}{2}$, $1 \leq j \leq b-1$, $1 \leq k \leq \lfloor\frac{c}{2}\rfloor$ alors :*

1. *Si $\beta = \frac{\pi}{2}$, $\alpha = \gamma$, $P_{\alpha,\beta,\gamma}$ a une racine double nulle ;*

2. *Sinon il existe* $k_0 \in \{1, \dots, c-1\}$ *tel que :*
   (a) *Pour tout* $k < k_0$, $P_{\alpha,\beta,\gamma}$ *n'a pas de racine réelle ;*
   (b) *Pour tout* $k \geq k_0$, $P_{\alpha,\beta,\gamma}$ *a deux racines réelles distinctes, en plus son discriminant satisfait :*

$$2^{-3n} < \Delta_{\alpha,\beta,\gamma} < 16. \tag{5.18}$$

Démonstration. Dans le cas où $\deg P_{\alpha,\beta,\gamma} = 2$, il est confirmé que $\gamma \neq \frac{\pi}{2}$ donc $\cos\gamma \neq 0$.

1. D'après l'identité (5.15) et le fait que $\cos\gamma \neq 0$, $\sin\alpha, \sin\beta, \sin\gamma > 0$, $\Delta_{\alpha,\beta,\gamma} = 0$ si et seulement si $\sin\gamma = \sin\alpha\sin\beta$, i.e. $(\frac{k}{c}, \frac{1}{2}, \frac{i}{a}, \frac{j}{b})$ est solution de l'équation de l''équation (5.16) qui a été résolue complètement par Myerson dans le lemme 5.14. Vu les possibilités dans l'ensemble des solutions, avec notre condition que $(a, b) = 1$, il faut que $\beta = \frac{\pi}{2}$ et $\sin\alpha = \sin\gamma$.

2. On est maintenant dans le cas où $\sin\alpha\sin\beta \neq \sin\gamma$, d'où $\Delta_{\alpha,\beta,\gamma} \neq 0$.

(a) Nous avons $0 < \sin\frac{\pi}{c} < \dots < \sin\frac{\lfloor c/2\rfloor\pi}{c} \leq 1$, $0 < \sin\alpha\sin\beta < 1$, il existe donc l'unique indice k_0 tel que $\sin\frac{(k_0-1)\pi}{c} < \sin\alpha\sin\beta < \sin\frac{k_0\pi}{c}$. Dans la formule (5.15), $k \geq k_0$ si et seulement si $\sin\gamma - \sin\alpha\sin\beta \geq \sin\frac{k_0\pi}{c} - \sin\alpha\sin\beta > 0$, ceci équivaut à $\Delta_{\alpha,\beta,\gamma} > 0$.

(b) Considérons $k \geq k_0$ i.e. $\Delta_{\alpha,\beta,\gamma} > 0$:

∘ Grâce à l'inégalité élémentaire $\sin x \geq \frac{2x}{\pi}$ on obtient

$$\cot^2\gamma \geq \cos^2\gamma \geq \sin^2\frac{\pi}{2c} \geq c^{-2}, \tag{5.19}$$

Quand $\Delta_{\alpha,\beta,\gamma}$ n'est pas nul, l'identité (5.15) s'écrit autrement :

$$\begin{aligned}\Delta_{\alpha,\beta,\gamma} &= \cot^2\gamma[4 - 2\cos(2\alpha - 2\beta) - 2\cos(2\alpha + 2\beta) + 4\cos 2\alpha + 4\cos 2\beta - 8\cos 2\gamma] \\ &= \cot^2\gamma\Big(4 - 4T_{2abk} - T_{2c(aj-bi)} - T_{2c(aj+bi)} + 2T_{2cbi} + 2T_{2caj}\Big)(2\cos\tfrac{\pi}{n}),\end{aligned} \tag{5.20}$$

où l'on pose $\delta_{\alpha,\beta,\gamma} = 4 - 4T_{2abk} - T_{2c(aj-bi)} - T_{2c(aj+bi)} + 2T_{2cbi} + 2T_{2caj}$ (voir le calcul avec Maple illustré dans la figure A.12, annexe A.3).

Comme $\|\delta_{\alpha,\beta,\gamma}\|_T = 24$, en appliquant l'inégalité (4.1), on a :

$$|\delta_{\alpha,\beta,\gamma}(2\cos\tfrac{\pi}{n})| \geq 24^{1-\frac{\phi(2n)}{2}}. \tag{5.21}$$

Les inégalités dans (5.21) et (5.19) nous donnent la première inégalité dans la suite (5.18) ;

∘ Posons $m = \sin\alpha\sin\beta$, $x = \sin^2\gamma$, alors

$$\Delta_{\alpha,\beta,\gamma} = 16(1 + m - x - \tfrac{m}{x}),$$

Considérons la fonction $g(x) = 16(1 + m - x - \frac{m}{x})$ sur $[m, 1]$, on a $g'(x) = -1 + \frac{m}{x^2} = \frac{(\sqrt{m}-x)(\sqrt{m}+x)}{x^2}$. La table de variation de g est :

x	m		$\sqrt{m}$		1
$g'(x)$		$+$	0	$-$	
$g(x)$	0	↗	$16(1-\sqrt{m})^2$	↘	0

Ainsi $\Delta_{\alpha,\beta,\gamma} < 16(1 - \sqrt{m})^2 < 16$. □

Pour tout α, β, $|2\cos\alpha\cos\beta| < 2$ et en cas où $P_{\alpha,\beta,\gamma}$ a des racines réelles, elles admettent l'une de trois valeurs $-2\cos\alpha\cos\beta$, $-2\cos\alpha\cos\beta \pm \frac{1}{2}\sqrt{\Delta_{\alpha,\beta,\gamma}}$, alors nous avons :

Corollaire 5.16. *Soit* ϕ *une valeur critique alors* $|\phi| < 4$.

5.3.2 La borne de séparation de $R_{a,b,c}$

Soit f un polynôme dans $\mathbb{Z}[x]$ alors la distance minimale entre les racines distinctes de f s'appelle sa borne de séparation :

$$\mathrm{Sep}(f) = \min\{|z_i - z_j|, f(z_i) = f(z_j) = 0, z_i \neq z_j\}.$$

Proposition 5.17. *Soit a, b, c trois entiers positifs tels que $(a, b) = 1$, a est impair. Soit $R_{a,b,c}$ le polynôme défini par l'identité (5.8). En posant $n = abc$ on a :*

$$\mathrm{Sep}(R_{a,b,c}) \geq 2^{-9n-1}.$$

Démonstration. Rappelons d'abord le contenu du lemme 4.11 qui va être utilisé à plusieurs reprises : Si $f(\gamma) \neq 0$ ($f \in \mathbb{Z}[x]$ donné dans la base $\mathcal{T}$, $\gamma = 2\cos\frac{\pi}{n}$) alors on a l'inégalité (4.1) :

$$|f(\gamma)| \geq \|f\|_T^{1-\frac{n}{2}}.$$

Considérons deux racines distinctes ϕ_1, ϕ_2 de $R_{a,b,c}$, il existe $(\alpha_1, \beta_1, \gamma_1)$, $(\alpha_2, \beta_2, \gamma_2)$ tels que $P_{\alpha_1,\beta_1,\gamma_1}(\phi_1) = P_{\alpha_2,\beta_2,\gamma_2}(\phi_2) = 0$. Afin de minorer la distance $|\phi_1 - \phi_2|$, nous allons considérer tous les cas possibles du couple $(P_{\alpha_1,\beta_1,\gamma_1}, P_{\alpha_2,\beta_2,\gamma_2})$.

1. Si $\deg P_{\alpha_1,\beta_1,\gamma_1} = \deg P_{\alpha_2,\beta_2,\gamma_2} = 1$, $\phi_1 = -2\cos\alpha_1\cos\beta_1$, $\phi_2 = -2\cos\alpha_2\cos\beta_2$ alors :

$$\begin{aligned}|\phi_1 - \phi_2| &= 2|\cos\alpha_1\cos\beta_1 - \cos\alpha_2\cos\beta_2| \\ &= |\cos(\alpha_1+\beta_1) + \cos(\alpha_1-\beta_1) + \cos(\alpha_2+\beta_2) + \cos(\alpha_2-\beta_2)| = \tfrac{1}{2}\left|f_1(2\cos\tfrac{\pi}{ab})\right|,\end{aligned}$$

où $f_1 = T_{b\,i_1+a\,j_1} + T_{b\,i_1-a\,j_1} + T_{b\,i_2+a\,j_2} + T_{b\,i_2-a\,j_2}$ (voir le calcul avec Maple illustré dans la figure A.7, annexe A.3). Comme $\|f_1\|_T = 8$ on a :

$$|\phi_1 - \phi_2| \geq 2^{2-\frac{3ab}{2}}. \tag{5.22}$$

2. Si $\deg P_{\alpha_1,\beta_1,\gamma_1} = 1, \deg P_{\alpha_2,\beta_2,\gamma_2} = 2$ alors $\gamma_1 = \frac{\pi}{2} \neq \gamma_2$.
On a : $\phi_1 = -2\cos\alpha_1\cos\beta_1$, $\phi_2 = -2\cos\alpha_2\cos\beta_2 \pm \frac{1}{2}\sqrt{\Delta_{\alpha_2,\beta_2,\gamma_2}}$. Ceci nous donne :

$$\begin{aligned}|\phi_1 - \phi_2| &= \left|2\cos\alpha_1\cos\beta_1 - 2\cos\alpha_2\cos\beta_2 \pm \tfrac{1}{2}\sqrt{\Delta_{\alpha_2,\beta_2,\gamma_2}}\right| \\ &= \frac{16|\sin^2\gamma_2(2\cos\alpha_1\cos\beta_1 - 2\cos\alpha_2\cos\beta_2)^2 - 4\cos^2\gamma_2(\sin^2\gamma_2 - \sin^2\alpha_2\sin^2\beta_2)|}{16\sin^2\gamma_2\left|2\cos\alpha_1\beta_1 - 2\cos\alpha_2\beta_2 \mp \frac{1}{2}\sqrt{\Delta_{\alpha_2,\beta_2,\gamma_2}}\right|}.\end{aligned}$$

Posons $A_1 = 16\sin^2\gamma_2(2\cos\alpha_1\cos\beta_1 - 2\cos\alpha_2\cos\beta_2)^2 - 4\cos^2\gamma_2(\sin^2\gamma_2 - \sin^2\alpha_2\sin^2\beta_2)$. Comme $\phi_1 \neq \phi_2$, $A_1 \neq 0$, on peut réécrire

$$A_1 = f_2(2\cos\tfrac{\pi}{n}),$$

avec f_2 (voir la figure A.8, annexe A.3) étant une forme de Chebyshev, $\|f_2\|_T = 256$. Vu l'inégalité (4.1), $|A_1| \geq 256^{1-\frac{n}{2}}$;

Mais $|\sin^2\gamma_2| \leq 1$, $|2\cos\alpha_1\beta_1 - 2\cos\alpha_2\beta_2| \leq 4$, $\left|\frac{1}{2}\sqrt{\Delta_{\alpha_2,\beta_2,\gamma_2}}\right| \leq 2$ (voir l'inégalité (5.18)), on a donc :

$$|\phi_1 - \phi_2| \geq \frac{256^{1-\frac{n}{2}}}{3.2^5} > 2^{1-4n}. \tag{5.23}$$

3. Quand $P_{\alpha_1,\beta_1,\gamma_1} \equiv P_{\alpha_2,\beta_2,\gamma_2}$, c'est-à-dire que ϕ_1 et ϕ_2 sont deux racines distinctes d'un même polynôme $P_{\alpha,\beta,\gamma}$ de degré 2 alors $|\phi_1 - \phi_2| = \sqrt{\Delta_{\alpha,\beta,\gamma}}$. En appliquant l'évaluation (5.18), on obtient immédiatement :

$$|\phi_1 - \phi_2| \geq 2^{-\frac{3n}{2}}; \tag{5.24}$$

4. Quand $\deg P_{\alpha_1,\beta_1,\gamma_1} = \deg P_{\alpha_2,\beta_2,\gamma_2} = 2$ mais $P_{\alpha_1,\beta_1,\gamma_1}$ et $P_{\alpha_2,\beta_2,\gamma_2}$ ont une racine commune, dénotons $\mathcal{Z}(P_{\alpha_1,\beta_1,\gamma_1}) = \{\phi_1, \phi\}$, $\mathcal{Z}(P_{\alpha_2,\beta_2,\gamma_2}) = \{\phi_2, \phi\}$. On a :

$$\begin{aligned} |\phi_1 - \phi_2| &= |(\phi + \phi_1) - (\phi - \phi_2)| = |4\cos\alpha_1\cos\beta_1 - 4\cos\alpha_2\cos\beta_2| \\ &= \left|f_1(2\cos\tfrac{\pi}{ab})\right| \geq 2^{3-\frac{3ab}{2}}. \end{aligned} \tag{5.25}$$

5. Quand $\deg P_{\alpha_1,\beta_1,\gamma_1} = \deg P_{\alpha_2,\beta_2,\gamma_2} = 2$, $\mathcal{Z}(P_{\alpha_1,\beta_1,\gamma_1})$ et $\mathcal{Z}(P_{\alpha_2,\beta_2,\gamma_2})$ sont disjoints alors $(P_{\alpha_1,\beta_1,\gamma_1}, P_{\alpha_2,\beta_2,\gamma_2}) = 1$. Dénotons $\mathcal{Z}(P_{\alpha_1,\beta_1,\gamma_1}) = \{\phi_1, \phi_1'\}$, $\mathcal{Z}(P_{\alpha_2,\beta_2,\gamma_2}) = \{\phi_2, \phi_2'\}$.

Comme $P_{\alpha_1,\beta_1,\gamma_1}$, $P_{\alpha_2,\beta_2,\gamma_2}$ sont tous unitaires on a $\mathrm{Res}(P_{\alpha_1,\beta_1,\gamma_1}, P_{\alpha_2,\beta_2,\gamma_2}) = (\phi_1 - \phi_2)(\phi_1 - \phi_2')(\phi_1' - \phi_2)(\phi_1' - \phi_2') \neq 0$.

On sait que $|\phi| \leq 4$ pour tout $\phi \in \{\phi_1, \phi_1', \phi_2, \phi_2'\}$, donc $|(\phi_1 - \phi_2')(\phi_1' - \phi_2)(\phi_1' - \phi_2')| \leq 2^9$, par la suite :

$$|\phi_1 - \phi_2| \geq \frac{|\mathrm{Res}(\widetilde{P}_{\alpha_1,\beta_1,\gamma_1}, \widetilde{P}_{\alpha_2,\beta_2,\gamma_2})|}{2^9} \geq \frac{|256\sin^4\gamma_1\sin^4\gamma_2\mathrm{Res}(P_{\alpha_1,\beta_1,\gamma_1}, P_{\alpha_2,\beta_2,\gamma_2})|}{2^{17}}$$

Posons $C_1 = 256\sin^4\gamma_1\sin^4\gamma_2\mathrm{Res}(P_{\alpha_1,\beta_1,\gamma_1}, P_{\alpha_2,\beta_2,\gamma_2})$, on a : $C_1 = f_4(2\cos\frac{\pi}{n})$ (voir la figure A.9, annexe A.3), où f_4 est une forme de Chebyshev dans $\mathbb{Z}[x]$, $\|f_4\|_T \leq 65552$. Avec $C_1 \neq 0$, appliquons l'inégalité (4.1) pour f_4 on obtient

$$|\phi_1 - \phi_2| \geq \frac{65552^{1-\frac{n}{2}}}{2^{17}} > 2^{-9n-1}. \tag{5.26}$$

En combinant les inégalités (5.22), (5.23), (5.24), (5.25), (5.26) on a la conclusion souhaitée. □

5.3.3 Isoler les valeurs critiques

Nous allons calculer indépendamment les racines des polynômes $P_{\alpha,\beta,\gamma}$ puis les comparer afin d'obtenir toutes les racines de $R_{a,b,c}$ avec leurs multiplicités.

La première étape est de calculer les racines de chaque $P_{\alpha,\beta,\gamma}$.

Lemme 5.18. *Soit a et b deux entiers positifs premiers entre-eux, soit c be un entier. Soit $\alpha = i\frac{\pi}{a}$, $\beta = j\frac{\pi}{b}$, $\gamma = k\frac{\pi}{c}$. Alors on peut calculer les racines réelles de $P_{\alpha,\beta,\gamma}$, si elles existent, à la précision $2^{-\ell}$ en $\widetilde{\mathcal{O}}(\ell + n)$ opérations binaires.*

Démonstration. La première opération à exécuter est d'identifier si $P_{\alpha,\beta,\gamma}$ a

(a) 1 racine simple,

(b) 1 racine double réelle (qui vaut zéro d'après la proposition 5.15),

(c) 2 racines simples réelles ou

(d) 0 racine réelle.

Si et seulement si $k = \frac{c}{2}$ on a (a) ; si non on teste la condition $\beta = \frac{\pi}{2}$ et $\gamma = \alpha$ qui assure (b).

Dans ces deux cas là, l'unique racine est $-2\cos\alpha\cos\beta = -\cos\frac{(ib+ja)\pi}{ab} - \cos\frac{(ib-ja)\pi}{ab}$ qu'on doit évaluer à la précision $2^{-\ell}$. Vu le lemme 4.3, cette tâche peut être faite en $\widetilde{\mathcal{O}}(\ell+n)$ opérations binaires.

Quand on est assuré que $P_{\alpha,\beta,\gamma}$ ne tombe pas dans les deux premiers cas, décider si c'est le cas (c) ou (d) retourne à calculer le signe du discriminant $\Delta_{\alpha,\beta,\gamma}$ de $P_{\alpha,\beta,\gamma}$.

La proposition 5.15 nous a affirmé que $|\Delta_{\alpha,\beta,\gamma}| \geq 2^{-3n}$. Afin de décider son signe, on évalue numériquement sa valeur à la précision 2^{-3n-1}, ce qui peut être effectué en $\widetilde{\mathcal{O}}(n)$ opérations binaires en utilisant à nouveau le calcul des fonctions élémentaires de Brent [Bre75, Bre76].

Quand $\Delta_{\alpha,\beta,\gamma} > 0$, la racine est alors calculée comme une racine d'un polynôme du second degré à la précision $2^{-\frac{3n}{2}}$, ce calcul utilise $\widetilde{\mathcal{O}}(n)$ opérations binaires selon [Bre75, Bre76]. □

Maintenant nous en déduisons les intervalles d'isolation des racines de $R_{a,b,c}$:

Corollaire 5.19. *Soit a et b deux entiers qui sont premiers entre-eux, soit c un entier. Alors on peut isoler les racines réelles de $R_{a,b,c}$ par des intervalles de longue 2^{-9n} en $\widetilde{\mathcal{O}}(n^2)$ opérations binaires. Il est aussi possible de calculer les racines réelles de $R_{a,b,c}$ et leurs multiplicités en $\widetilde{\mathcal{O}}(n^2)$ opérations binaires.*

Démonstration. $R_{a,b,c}$ est le produit de $\mathcal{O}(n)$ facteurs $\widetilde{P}_{\alpha,\beta,\gamma}$. D'après la proposition 5.17, la distance entre deux racines réelles arbitraires de $R_{a,b,c}$ est supérieure à 2^{-9n-1}.

Supposons que les racines de tous les facteurs $\widetilde{P}_{\alpha,\beta,\gamma}$ soient calculées indépendamment à la précision plus fine que 2^{-9n-2}, alors deux de ces valeurs présentent la même racine si et seulement si la distance entre elles est inférieure à 2^{-9n-1}.

Notre première étape est alors de calculer approximativement les racines de tous les facteurs $\widetilde{P}_{\alpha,\beta,\gamma}$ jusqu'à la précision 2^{-9n-2}. Cette étape utilise $\widetilde{\mathcal{O}}(n^2)$ opérations binaires, selon le lemme 5.20.

La deuxième étape est de trier la liste de valeurs d'approximation, ceci utilise $\widetilde{\mathcal{O}}(n)$ comparaison entre les points flottants de la précision $2^{-\mathcal{O}(n)}$. On utilise donc $\widetilde{\mathcal{O}}(n^2)$ opérations binaires.

La dernière étape est de grouper les valeurs qui se sont séparées par une distance inférieure à 2^{-9n-1}, cela utilise aussi $\widetilde{\mathcal{O}}(n^2)$ opérations binaires. □

5.4 Calculer les diagrammes des nœuds

D'abord nous décidons si la courbe $\mathcal{C}(a,b,c,\phi)$ est régulière ou non. Le lemme suivant sera très utile :

Lemme 5.20. *Soit a et b deux entiers premiers entre-eux, soit c un entier. Soit $\alpha = i\frac{\pi}{a}$, $\beta = j\frac{\pi}{b}$, $\gamma = k\frac{\pi}{c}$ et ϕ un nombre rationnel avec $\tau(\phi) = \tau$. Alors, on peut tester si $P_{\alpha,\beta,\gamma}(\phi) = 0$ en $\widetilde{\mathcal{O}}(n^2 + n\tau)$ opérations binaires. On peut calculer le signe de $P_{\alpha,\beta,\gamma}(\phi)$ en $\widetilde{\mathcal{O}}(n^2\tau)$ opérations binaires.*

Démonstration. Supposons que $\phi = \frac{u}{v}$.

Quand $\gamma = \frac{\pi}{2}$ on écrit

$$2vP_{\alpha,\beta,\gamma} = 2u + 4v\cos\tfrac{i\pi}{a}\cos\tfrac{j\pi}{b} = f(2\cos\tfrac{\pi}{ab}),$$

avec $f = vT_{-ja+ib} + vT_{ja+ib} + 2\,u$ (voir la figure A.10, annexe A.3), c'est une forme de Chebyshev avec $\|f\|_T \leq 4|v| + 2|u| \leq 6 \cdot 2^\tau$. Ceci nous donne $\tau_{\mathcal{T}}(uf) \leq 2\tau + \log_2 6 = \mathcal{O}(\tau)$. Le signe de $P_{\alpha,\beta,\gamma}(\phi)$ est celui de $uf(2\cos\frac{\pi}{ab})$.

Si $\gamma = \frac{k\pi}{c} \neq \frac{\pi}{2}$, la formule (5.6) se transforme en : $4v^2\sin^2\gamma P_{\alpha,\beta,\gamma}(\phi) = g(2\cos\frac{\pi}{n})$ (voir la figure A.11, annexe A.3), avec

$$\begin{aligned} g =& 2\,u^2 + 2\,v^2 - u^2T_{2\,kab} + v^2T_{4\,kab} + 2\,uvT_{ibc-jac} + 2\,uvT_{ibc+jac} \\ & - uvT_{ibc-jac-2\,kab} - uvT_{ibc+jac-2\,kab} - uvT_{ibc-jac+2\,kab} - uvT_{ibc+jac+2\,kab} \\ & + v^2T_{2\,ibc-2\,jac} + v^2T_{2\,ibc+2\,jac} - v^2T_{2\,ibc-2\,kab} - v^2T_{2\,ibc+2\,kab} \\ & - v^2T_{2\,jac-2\,kab} - v^2T_{2\,jac+2\,kab}, \end{aligned}$$

c'est une forme de Chebyshev avec $\|g\|_T \leq 4u^2 + 16|uv| + 16v^2 \leq 36 \cdot 2^{2\tau}$, donc $\tau_{\mathcal{T}}(g) \leq 2\tau + \log_2 36 = \mathcal{O}(\tau)$. Le signe de $P_{\alpha,\beta,\gamma}(\phi)$ est celui de $g(2\cos\frac{\pi}{n})$.

On est maintenant dans la situation déterminer le signe d'une somme de cosinus : vu le lemme 4.10, on sait le coût de décider si la somme est nulle. Grâce à la proposition 4.17, on peut calculer son signe par la complexité énoncée. □

Proposition 5.21. *Soit a et b deux entiers premiers entre-eux, soit c un entier. Soit ϕ un nombre rationnel de taille binaire τ. Alors on peut décider si $\mathcal{C}(a,b,c,\phi)$ est un nœud en temps de calcul $\widetilde{\mathcal{O}}(n^2 + n\tau)$ (avec $n = abc$). Si c'est un nœud, la nature de tous les croisements de $\mathcal{C}(a,b,c,\phi)$ peut être précisée en $\widetilde{\mathcal{O}}(n^2\tau)$ opérations binaires.*

Démonstration. $\mathcal{C}(a,b,c,\phi)$ est régulière si et seulement si $R_{a,b,c}(\phi) \neq 0$. Nous calculons d'abord les s racines réelles $\phi_1, \ldots, \phi_s$ de $R_{a,b,c}$ par les intervalles d'isolation de taille binaire 2^{-9n} en temps de calcul $\widetilde{\mathcal{O}}(n^2)$. Dénotons l'intervalle d'isolation de ϕ_i par $[u_i, v_i]$, où $u_i \leq \phi_i \leq v_i$ et $\tau(u_i), \tau(v_i) \leq -9n$.

Pour chaque i, supposons également qu'on connaisse la liste de α, β, γ tel que $P_{\alpha,\beta,\gamma}(\phi_i) = 0$. D'après le corollaire 5.19 précédent, nous obtenons ces racines en $\widetilde{\mathcal{O}}(n^2)$ opérations binaires.

Nous trouvons l'unique indice i_0 tel que $\phi_{i_0} \leq \phi < \phi_{i_0+1}$ en exécutant $\mathcal{O}(s\log_2 s)$ comparaisons entre ϕ et les u_i's. Ceci utilise $\widetilde{\mathcal{O}}(s(\tau+n)) = \widetilde{\mathcal{O}}(n^2 + n\tau)$ opérations binaires.

Deux cas peuvent apparaître :

- Si $v_{i_0} < \phi < u_{i_0+1}$, $R_{a,b,c}(\phi) \neq 0$ et $\phi_{i_0} < \phi < \phi_{i_0+1}$.

- Si $u_{i_0} \leq \phi \leq v_{i_0}$, on calcule le signe de $\phi - \phi_{i_0}$. Considérons le polynôme $P_{\alpha_0,\beta_0,\gamma_0}$ tel que $P_{\alpha_0,\beta_0,\gamma_0}(\phi_{i_0}) = 0$.

$R_{a,b,c}(\phi) = 0$ si et seulement si $P_{\alpha_0,\beta_0,\gamma_0}(\phi) = 0$. Cela peut être confirmée en $\widetilde{\mathcal{O}}(n^2+n\tau)$ opérations binaires, vu le lemme 5.20.

On peut aussi calculer le signe de $P_{\alpha_0,\beta_0,\gamma_0}(\phi)$ en $\widetilde{\mathcal{O}}(n^2\tau)$ opérations binaires par le lemme 5.20.

- Si $\gamma_0 = \frac{\pi}{2}$, il est clair que $\phi - \phi_{i_0}$ et $P_{\alpha_0,\beta_0,\gamma_0}(\phi)$ ont le même signe.

- Si $\gamma_0 \neq \frac{\pi}{2}$, $P_{\alpha_0,\beta_0,\gamma_0}(\phi)$ a deux racines ϕ_{i_0} et $-2\cos\alpha\cos\beta - \phi_{i_0}$. Comme $|\phi - \phi_{i_0}| < \cos\alpha\cos\beta$, on a $(\phi + 2\cos\alpha\cos\beta)(\phi - \phi_0)P_{\alpha_0,\beta_0,\gamma_0}(\phi) > 0$. Nous calculons le signe de $\phi + 2\cos\alpha\cos\beta$ en $\widetilde{\mathcal{O}}(n^2\tau)$ opérations binaires, puis déduisons le signe de $\phi - \phi_{i_0}$ en $\widetilde{\mathcal{O}}(n^2\tau)$ opérations binaires.

Nous avons décidé si $\mathcal{C}(a,b,c,\phi)$ est un nœud en $\widetilde{\mathcal{O}}(n^2 + n\tau)$ opérations binaires. Si $\mathcal{C}(a,b,c,\phi)$ est un nœud, on trouve i_0 tel que $\phi_{i_0} < \phi < \phi_{i_0+1}$ en $\widetilde{\mathcal{O}}(n^2\tau)$ opérations binaires.

Maintenant on détermine la nature des croisement sur les $\frac{1}{2}(a-1)(b-1)$ points doubles $A_{\alpha,\beta}$ de paramètres $(2\cos(\alpha+\beta), 2\cos(\alpha-\beta))$ de la courbe plane $\mathscr{C}(a,b)$.

Soit $\alpha = \frac{i\pi}{a}$ et $\beta = \frac{j\pi}{b}$. Les racines réelles $\phi'_1 = \phi_{j_1}, \ldots, \phi'_k = \phi_{j_k}$ de $Q_c(2\cos(\alpha + \beta), 2\cos(\alpha - \beta))$ sont collectées depuis l'ensemble de zéros de $R_{a,b,c}$ en $\widetilde{\mathcal{O}}(n)$ opérations binaires.

On trouve k_0 tel que $\phi'_{k_0} < \phi < \phi'_{k_0+1}$ en $\widetilde{\mathcal{O}}(\log_2 k \log_2 n) = \widetilde{\mathcal{O}}(\log_2^2 n)$ opérations binaires en insérant i_0 dans la suite $(j_1, \ldots, j_k)$. Le signe de $Q_c(2\cos(\alpha+\beta), 2\cos(\alpha-\beta), \phi)$ est donc $(-1)^{k_0}$. Le signe du croisement sur le point double $A_{\alpha,\beta}$ est alors $(-1)^{i+j+\lfloor\frac{ib}{a}\rfloor+\lfloor\frac{ja}{b}\rfloor+k_0}$, et peut être calculé en $\mathcal{O}(n)$ opérations binaires.

De cette manière, la nature de tous les croisements est calculé en $\mathcal{O}(ab) \cdot \mathcal{O}(n) = \mathcal{O}(n^2)$ opérations binaires. □

Maintenant nous listons tous les diagrammes de nœuds $\mathcal{C}(a, b, c, \phi)$ possibles.

Proposition 5.22. *Soit a, b, c trois entiers où a est impair, $(a, b) = 1$. On peut lister tous les diagrammes de nœuds $\mathcal{C}(a, b, c, \phi)$ possibles en $\widetilde{\mathcal{O}}(n^2)$ opérations binaires.*

Démonstration. $\mathcal{C}(a, b, c, \phi)$ est une courbe régulière si et seulement si $R_{a,b,c}(\phi) \neq 0$. Nous calculons d'abord les s racines réelles $\phi_1, \ldots, \phi_s$ de $R_{a,b,c}$ présentées par les intervalles d'isolation $[u_i, v_i]$ de taille binaire 2^{-9n} en temps de calcul $\widetilde{\mathcal{O}}(n^2)$.

Pour chaque i, supposons qu'on sait la liste des α, β, γ tel que $P_{\alpha,\beta,\gamma}(\phi_i) = 0$.

Le nœud $\mathcal{C}(a, b, c, \phi)$ est unique pour tout $\phi \in (v_k, u_{k+1})$ parce que le signe des croisements sur les points doubles $A_{\alpha,\beta}$ est constant. Pour tout k dans $1, \ldots, s$, nous choisissons un nombre rationnel r_k dans (v_k, u_{k+1}), (en particulier $r_0 = -4$, $r_{s+1} = 4$).

Pour tout α, β, on calcule les étiquettes $(i_1 < \cdots < i_k)$ des racines $\phi'_1 = \phi_{i_1}, \ldots, \phi'_k = \phi_{i_k}$ of $Q_c(2\cos(\alpha + \beta), 2\cos(\alpha - \beta), \phi)$ en $\widetilde{\mathcal{O}}(n)$ opérations binaires.

Soit $0 \leq k \leq s + 1$, on calcule le signe de $Q_c(2\cos(\alpha + \beta), 2\cos(\alpha - \beta), r_k)$ dans $\mathcal{O}(k) = \mathcal{O}(c)$ opérations binaires.

Le diagramme de $\mathcal{C}(a, b, c, r_k)$ est donc déterminé en $\widetilde{\mathcal{O}}(n)$ opérations binaires. □

Conclusion du chapitre 5

1. Nous avons démontré que le polynôme caractéristique $R_{a,b,c}$ peut être calculé en complexité $\widetilde{\mathcal{O}}(n^3)$ par une méthode numérique, et en complexité $\widetilde{\mathcal{O}}(n^4)$ à l'aide du calcul dans la base des polynômes unitaires de Chebyshev. En fait nous n'avons trouvé aucune d'autres techniques atteignant ces deux complexités ;
2. Travaillant dans la base des polynômes unitaires de Chebyshev nous permet aussi d'analyser entièrement la complexité binaire du calcul des diagrammes des des nœuds de Chebyshev ce qui est assez difficile avec les autres méthodes préexistentes.

Annexe A

Calculs faits avec Maple 18

A.1 Le paquetage `ChebUnit`

Grâce aux algorithmes dans trois premiers chapitres, le paquetage `ChebUnit` réalise les calculs dans la base des polynômes unitaires de Chebyshev :

- `ChebTn` : $n \mapsto [\mathcal{X}]T_n$;
- `ChebUn` : $n \mapsto [\mathcal{X}]U_n$;
- `p2t` : $([\mathcal{T}]f, n) \mapsto f(2\cos\frac{\pi}{n})$;
- `t2p` : $\sum_{j=0}^{d} f_j \cos\frac{j\pi}{n} \mapsto (f_0 + \sum_{j=1}^{d} \frac{f_j}{2} T_j, n)$;
- `x2T` : $[\mathcal{X}]f \mapsto [\mathcal{T}]f$;
- `T2x` : $[\mathcal{T}]f \mapsto [\mathcal{X}]f$;
- `mulT` : $([\mathcal{T}]f, [\mathcal{T}]g) \mapsto [\mathcal{T}](f \cdot g)$;
- `quoT` : $([\mathcal{T}]f, [\mathcal{T}]g) \mapsto [\mathcal{T}]\mathrm{Quo}(f, g)$;
- `MnT` : $n \mapsto [\mathcal{T}]M_n$;

Voyons quelques exemples dans la figure A.1, A.2

Exemple A.1. En exécutant `MnT(1260)`, on obtient :

$$\begin{aligned}[\mathcal{T}]M_{1260} = T_{288} - T_{276} + T_{264} + T_{228} - T_{216} + 2T_{204} - T_{192} \\ + T_{180} + T_{144} - T_{132} + T_{120} - T_{108} + T_{96} - T_{84} - T_{48} - T_{24} - 1,\end{aligned}$$

par contre, en utilisant `T2x`(M_{1260}), Maple renvoie un résultat illustré par la figure A.3.

Enfin, nous avons calculé $M_{996}, M_{1260}, M_{8640}$, puis nous avons écrit la procédure `VoirLaTaille` permettant de calculer les sommes de bits qu'il faut utiliser pour sauvegarder leurs coefficients dans la base des monômes, voir la figure A.4.

A.2 Trouver la forme de Chebyshev

Soit F une somme de type $F = f_0 + \sum_{j=1}^{d} f_j \cos r_j\pi$ où $f_j, r_j \in \mathbb{Q}$, $j = 0, \ldots, d$ alors elle peut être réécrite en

$$F = \widetilde{f}_0 + \sum_{j=1}^{\lfloor\frac{n-1}{2}\rfloor} \widetilde{f}_j \cdot 2\cos\frac{j\pi}{n}\pi,$$

```
> with(ChebUnit);
```

$$[ChebTn, ChebUn, MnT, T2x, comT, mulT, p2t, quoT, t2p, x2T] \quad (1)$$

```
> ChebTn(11,x);
  ChebUn(17,t);
```

$$x^{11} - 11\,x^9 + 44\,x^7 - 77\,x^5 + 55\,x^3 - 11\,x$$

$$t^{17} - 16\,t^{15} + 105\,t^{13} - 364\,t^{11} + 715\,t^9 - 792\,t^7 + 462\,t^5 - 120\,t^3 + 9\,t \quad (2)$$

```
> L:=RandomTools:-Generate(list(integer(range=-1000..1000),16)):
> ff:=L[1]+add(L[j]*T[j-1],j=2..nops(L));
```

$$ff := -140 + 782\,T_1 - 250\,T_2 + 913\,T_3 + 324\,T_4 - 995\,T_5 - 46\,T_6 - 701\,T_7 - 901\,T_8 + 573\,T_9 - 804\,T_{10} + 306\,T_{11} + 375\,T_{12} - 984\,T_{13} - 257\,T_{14} + 764\,T_{15} \quad (3)$$

```
> FF30:=p2t(ff,30);
```

$$FF30 := -944 + 1564\cos\left(\frac{1}{30}\pi\right) - 500\cos\left(\frac{1}{15}\pi\right) + 1826\cos\left(\frac{1}{10}\pi\right) + 648\cos\left(\frac{2}{15}\pi\right) - 995\sqrt{3} - 92\cos\left(\frac{1}{5}\pi\right) - 1402\cos\left(\frac{7}{30}\pi\right) - 1802\cos\left(\frac{4}{15}\pi\right) + 1146\cos\left(\frac{3}{10}\pi\right) + 612\cos\left(\frac{11}{30}\pi\right) + 750\cos\left(\frac{2}{5}\pi\right) - 1968\cos\left(\frac{13}{30}\pi\right) - 514\cos\left(\frac{7}{15}\pi\right) \quad (4)$$

```
> t2p(FF30,T);
  fT:=t2p(FF30,T)[1]:
```

$$[-944 + 782\,T_1 - 250\,T_2 + 913\,T_3 + 324\,T_4 - 995\,T_5 - 46\,T_6 - 701\,T_7 - 901\,T_8 + 573\,T_9 + 306\,T_{11} + 375\,T_{12} - 984\,T_{13} - 257\,T_{14}, 30] \quad (5)$$

```
> fx:=T2x(fT);
```

$$fx := -257\,x^{14} - 984\,x^{13} + 3973\,x^{12} + 13098\,x^{11} - 24289\,x^{10} - 66753\,x^9 + 73319\,x^8 + 161110\,x^7 - 110396\,x^6 - 183267\,x^5 + 72327\,x^4 + 85258\,x^3 - 13637\,x^2 - 13026\,x - 242 \quad (6)$$

```
> x2T(fx);
```

$$-944 + 782\,T_1 - 250\,T_2 + 913\,T_3 + 324\,T_4 - 995\,T_5 - 46\,T_6 - 701\,T_7 - 901\,T_8 + 573\,T_9 + 306\,T_{11} + 375\,T_{12} - 984\,T_{13} - 257\,T_{14} \quad (7)$$

```
> L1:=RandomTools:-Generate(list(integer(range=-1000..1000),7)):
  gT:=L1[1]+add(L1[j]*T[j-1],j=2..nops(L1));
```

$$gT := 881 - 238\,T_1 + 185\,T_2 + 297\,T_3 + 203\,T_4 - 689\,T_5 + 194\,T_6 \quad (8)$$

```
> hT:=mulT(fT,gT);
```

$$hT := 730732 + 771455\,T_1 - 1119715\,T_2 + 1551072\,T_3 - 1082373\,T_4 - 108833\,T_5 - 213049\,T_6 - 251893\,T_7 - 884002\,T_8 + 473111\,T_9 - 120055\,T_{10} - 543264\,T_{11} + 905537\,T_{12} - 237264\,T_{13} - 401559\,T_{14} + 163781\,T_{15} - 474502\,T_{16} - 475092\,T_{17} + 698555\,T_{18} - 13823\,T_{19} - 49858\,T_{20} \quad (9)$$

```
> MnT(1260);
```

$$-1 - T_{24} - T_{48} - T_{84} + T_{96} - T_{108} + T_{120} - T_{132} + T_{144} + T_{180} - T_{192} + 2\,T_{204} - T_{216} + T_{228} + T_{264} - T_{276} + T_{288} \quad (10)$$

FIGURE A.1 – Les commandes du `ChebUnit` : `ChebTn`, `ChebUn`, `p2t`, `t2p`, `T2x`, `x2T`, `mulT` et `MnT`

où n est le dénominateur commun de $r_1, \ldots, r_j, j = 1, \ldots, d$. Notre fonction `t2p` prend F pour renvoyer (f, n) où f est la forme de Chebyshev dans $\mathbb{Q}[x]$:

$$f = \widetilde{f}_0 + \sum_{j=1}^{\lfloor \frac{n-1}{2} \rfloor} \widetilde{f}_j T_j,$$

```
> with(ChebUnit);
```

$$[\,ChebTn,\ ChebUn,\ MnT,\ T2x,\ comT,\ mulT,\ p2t,\ quoT,\ t2p,\ x2T\,] \tag{1}$$

```
> M121:=MnT(121);
  h:=117*T[129]-3*T[49]-19*T[40]+99;
  f:=mulT(h,M121);
  F:=p2t(f,121);
```

$$M121 := -1 + T_{11} - T_{22} + T_{33} - T_{44} + T_{55}$$

$$h := 117\,T_{129} - 3\,T_{49} - 19\,T_{40} + 99$$

$$\begin{aligned} f := {} & -99 + 19\,T_4 - 3\,T_6 - 117\,T_{129} + 3\,T_{49} + 19\,T_{40} - 3\,T_{16} - 19\,T_{29} + 117\,T_{162} + 99\,T_{11} - 19\,T_{51} + 19\,T_{62} - 3\,T_{82} + 3\,T_5 - 19\,T_{95} - 117\,T_{107} \\ & - 19\,T_{15} + 19\,T_{18} - 99\,T_{44} - 3\,T_{60} + 3\,T_{71} + 99\,T_{55} - 19\,T_{73} + 117\,T_{74} - 117\,T_{85} - 99\,T_{22} + 3\,T_{27} + 99\,T_{33} - 3\,T_{38} - 117\,T_{151} \\ & + 117\,T_{140} + 19\,T_{84} + 117\,T_{96} + 3\,T_{93} - 3\,T_{104} + 117\,T_{118} - 19\,T_7 - 117\,T_{173} + 117\,T_{184} \end{aligned}$$

$$\begin{aligned} F := {} & -99 + 6\cos\left(\frac{27}{121}\pi\right) - 6\cos\left(\frac{38}{121}\pi\right) + 234\cos\left(\frac{58}{121}\pi\right) - 38\cos\left(\frac{15}{121}\pi\right) + 38\cos\left(\frac{18}{121}\pi\right) - 6\cos\left(\frac{60}{121}\pi\right) - 38\cos\left(\frac{59}{121}\pi\right) \\ & + 6\cos\left(\frac{39}{121}\pi\right) + 6\cos\left(\frac{5}{121}\pi\right) - 234\cos\left(\frac{25}{121}\pi\right) - 6\cos\left(\frac{28}{121}\pi\right) + 38\cos\left(\frac{26}{121}\pi\right) + 234\cos\left(\frac{14}{121}\pi\right) - 234\cos\left(\frac{41}{121}\pi\right) \\ & - 38\cos\left(\frac{51}{121}\pi\right) + 234\cos\left(\frac{52}{121}\pi\right) + 198\cos\left(\frac{1}{11}\pi\right) + 6\cos\left(\frac{17}{121}\pi\right) - 234\cos\left(\frac{3}{121}\pi\right) - 38\cos\left(\frac{7}{121}\pi\right) - 198\cos\left(\frac{2}{11}\pi\right) \\ & - 234\cos\left(\frac{47}{121}\pi\right) + 234\cos\left(\frac{36}{121}\pi\right) + 198\cos\left(\frac{3}{11}\pi\right) + 198\cos\left(\frac{5}{11}\pi\right) - 6\cos\left(\frac{50}{121}\pi\right) + 38\cos\left(\frac{48}{121}\pi\right) + 234\cos\left(\frac{8}{121}\pi\right) \\ & + 6\cos\left(\frac{49}{121}\pi\right) + 38\cos\left(\frac{40}{121}\pi\right) - 234\cos\left(\frac{19}{121}\pi\right) - 38\cos\left(\frac{37}{121}\pi\right) - 6\cos\left(\frac{16}{121}\pi\right) - 38\cos\left(\frac{29}{121}\pi\right) + 234\cos\left(\frac{30}{121}\pi\right) \\ & + 38\cos\left(\frac{4}{121}\pi\right) - 6\cos\left(\frac{6}{121}\pi\right) - 198\cos\left(\frac{4}{11}\pi\right) \end{aligned} \tag{2}$$

```
> is(F=0);
  is(F>0);
  is(F<0);
```

$$FAIL$$
$$FAIL$$
$$FAIL \tag{3}$$

```
> evalf(F,50);
```

$$-1.0\ 10^{-47} \tag{4}$$

Figure A.2 – Les commandes `MnT` et `mulT` appliquées dans l'exemple 4.5

ceci est illustrée dans la figure A.5.

A.3 Minorer une somme de cosinus

Pour une borne théorique : trouver la valeur maximale possible de $\|.\|_T$

Soit $\alpha = \frac{i\pi}{a}$, $\beta = \frac{j\pi}{b}$, $\gamma = \frac{k\pi}{c}$. Nous devons toujours minorer une expression $A = \omega(\alpha, \beta, \gamma)$ avec $\omega \in \mathbb{Z}[u, v, w]$. Avec Maple 18, il est possible d'exécuter successivement trois calculs ci-dessous :

1. Utilisons la commande `combine`$(\omega, $`trig`$)$ pour réécrire :

$$\omega(\alpha, \beta, \gamma) = \omega_0 + \sum_{h=1}^{d} \omega_h \cos(x_h\alpha + y_h\beta + z_h\gamma),$$

voir la figure A.6.

2. Nous avons écrit un procédure qui s'appelle `rec1`, qui permet de prélever les coefficients existents dans une expression cos ou sin :

$$\omega_h \cos(x_h\alpha + y_h\beta + z_h\gamma) \overset{\texttt{rec1}}{\longmapsto} [\omega_h, \cos, x_h\alpha + y_h\beta + z_h\gamma]$$

3. Listons tous les dénominateurs des $\frac{\omega_h}{2}$, on prend la commande `lcm` pour obtenir m, leurs ppcm

$$m = \text{ppcm}(\text{dénominateur}(\tfrac{\omega_h}{2}), h = 0, \ldots, d);$$

Sur ce point là, pour tout (α, β, γ), le nombre algébrique $m \cdot \omega(\alpha, \beta, \gamma)$ peut toujours être écrit sous la forme

$$m \cdot \omega(\alpha, \beta, \gamma) = f(2 \cos \tfrac{\pi}{abc}).$$

Pour un triplet précisé $(\alpha = \frac{i\pi}{a}, \frac{j\pi}{b}, \frac{k\pi}{c})$, on risque de perdre quelques coefficients s'il apparait des cosinus de valeurs en opposite, (par exemple $2 \cos \frac{\pi}{11} + 7 \cos \frac{10\pi}{11} = -5 \cos \frac{\pi}{11}$), de toute façon :

$$\|f\|_T \leq m \sum_{h=0}^{d} |\omega_h|,$$

cela nous permet d'appliquer le lemme 4.11 afin de minorer la valeur absolue de $\omega(\alpha, \beta, \gamma)$. Précisément :

$$|\omega(\alpha, \beta, \gamma)| \geq \tfrac{1}{2m} \Big(2m \sum_{h=0}^{d} |\omega_h|\Big)^{1-\frac{abc}{2}}.$$

Le processus est empaqueté dans la procédure `PourMinorer` :

$$\texttt{PourMinorer} : \omega(\alpha, \beta, \gamma) \mapsto \Big[M = m \sum_{h=0}^{d} |\omega_h|, m\Big]$$

Quelques calculs dans le chapitre 5 sont illustrés par les figures A.7, A.8, A.9, A.10, A.11, A.12, A.13.

Pour une borne en pratique : outil existent de Maple

La commande `shake` renvoie un intervalle qui borne la vraie valeur d'une expression. Elle a été utilisé pour minorer $|a_j|, j = 1, \ldots, 4$ dans l'exemple 4.18 illustré dans la figure A.5.

```
> XM1260:=T2x(MnT(1260));
```

$$
\begin{aligned}
XM1260 := {} & x^{288} - 288\,x^{286} + 41040\,x^{284} - 3857856\,x^{282} + 269105832\,x^{280} - 14856924096\,x^{278} + 676165660127\,x^{276} - 26090988105324\,x^{274} + 871275934013706\,x^{272} \\
& - 25576858479602208\,x^{270} + 668218428413856765\,x^{268} - 15692496830770217436\,x^{266} + 333986745386760827663\,x^{264} - 6486536412096159085368\,x^{262} \\
& + 115631473382947395664443\,x^{260} - 1901495332992504239109164\,x^{258} + 28970714344308812250623058\,x^{256} - 410507535534174335300405760\,x^{254} \\
& + 5427990724295585143552638816\,x^{252} - 67174968346178017770151719552\,x^{250} + 780157029185835915837197188575\,x^{248} \\
& - 8523149164694745220794956040200\,x^{246} + 87780276278496953762113157820900\,x^{244} - 853924171671658209007766212189200\,x^{242} \\
& + 7860252775761540307118788477751250\,x^{240} - 68572479510596198445956181866676960\,x^{238} + 567806630608355681106416595144854529\,x^{236} \\
& - 4468646294220124894199519441278429164\,x^{234} + 33466961752656487684172989711807628241\,x^{232} - 238791231281359999227166282484609116968\,x^{230} \\
& + 1624952197086146713060984581358204694206\,x^{228} - 10556168158045557291215534297545210419192\,x^{226} \\
& + 65525010169562601168459174926418206402121\,x^{224} - 388960584700738915360063982865022924851680\,x^{222} \\
& + 2209723270322390402194120345302726582380139\,x^{220} - 12023079920997659630106375565405887235643124\,x^{218} \\
& + 62694357719176720870265454956743115190107415\,x^{216} - 313505143753134943445895727852004984298212136\,x^{214} \\
& + 1504228405078294405128402926680990584594084366\,x^{212} - 6928949072812851586704401544407630392627346776\,x^{210} \\
& + 30656343917359040206969174228142891748913565814\,x^{208} - 130338329299091576472940886099978049636974840544\,x^{206} \\
& + 532730507204982236002107892180807474499149378871\,x^{204} - 2094101625196956969922741163942103936551190810324\,x^{202} \\
& + 7919558200640087253835735523995154814558290642772\,x^{200} - 28824596598210759469222934729889982060710673092320\,x^{198} \\
& + 100999480136418493527729757602928477229208091333908\,x^{196} - 340794236972520435036005913708100111032816249092688\,x^{194} \\
& + 1107632562244521755210708234744685904848848417227421\,x^{192} - 3468428770212167976062299283612014278417890996735424\,x^{190} \\
& + 10466395735307715566819746614805933955930017206766572\,x^{188} - 30442016936306381195410011148295636828657763356550992\,x^{186} \\
& + 85356662068517743407411699579670838166890270517783774\,x^{184} - 230757875357401766802124838514133251000045819895251600\,x^{182} \\
& + 601575258437920620640556859301801246243308794480581887\,x^{180} - 1512477911583558758848222098454123255030839471977757900\,x^{178} \\
& + 3667730308679155035497153263455821243227225825153412898\,x^{176} - 8579286964229684885209213908531919556263921925992585 92\,x^{174} \\
& + 19358839598484293072391512711327442140454406428773086095\,x^{172} - 42140872526854571763559608994749360389642100266275051284\,x^{170} \\
& + 88499039448036695914900180779328557439056731123894355893\,x^{168} - 179304960356667266039707599497057427331045555045280565000\,x^{166} \\
& + 350482124490522486733931671492090463660672750671637893107\,x^{164} - 660927278739401520916059422975281578605014026748308336492\,x^{162} \\
& + 1202384871526250553203691439157741949524520895793219414631\,x^{160} - 2110165686181621119971918730541849270872672403834920973152\,x^{158} \\
& + 3572296458449139671007815002195809703815986592259970743485\,x^{156} - 5833167335843387876787256204445912995546673329020722580012\,x^{154} \\
& + 9186476590600532060830900001891021571427107607051538078247\,x^{152} - 13951989698770072622131673403340029570078431504811341582184\,x^{150} \\
& + 20432166594346388355163281742995956441198220913552203034200\,x^{148} - 28848659238568015791973213437790552586884046336923778237088\,x^{146} \\
& + 39264940135399122687190689397206725075440010333074942315419\,x^{144} - 51508716862864843170922807355499271705396108006721407771440\,x^{142} \\
& + 65114025909442379865422893121365659959968613765616576063576\,x^{140} - 79305072633188542451334033679840536121269663491333157662112\,x^{138} \\
& + 93039523925896018742062591666481109861250905659569728655038\,x^{136} - 105117336102993302698035445988464595057125144900357524100848\,x^{134} \\
& + 114344279110255088648993781541379436415757926740476058776400\,x^{132} - 119721753271541505840360036950591437222135086175858942254528\,x^{130} \\
& + 120622345590607332246421331167728058876654606704868215975174\,x^{128} - 116090041963829347266527368716961693677297357107914892362496\,x^{126} \\
& + 108967101634007641276323183407001667248087683596550515148265\,x^{124} - 97638817747890317928104305607287783787187032200107840558300\,x^{122} \\
& + 84076285646007150653844593513957104083076168328594513597 41\,x^{120} - 69547801676084881451238146376040077538286685911637370457304\,x^{118} \\
& + 55243063056305135055554463501334870114593510789127993204677\,x^{116} - 42118444496192305283683077847447925965997616509329873160580\,x^{114} \\
& + 30808652623948042895603285311347196873586879267492623862819\,x^{112} - 21610949394626060513429644274914438870604617409494655397072\,x^{110} \\
& + 14529798581942340758141592120515256559251714335582174540767\,x^{108} - 9358412434003156958038097526575580335780613125150158122132\,x^{106} \\
& + 5771140799557791823570573948329123913722055082283036264261\,x^{104} - 3405535014626013113290457310531362219364471022577007884872\,x^{102} \\
& + 1921795101800198564495631219788698075155099174568512011640\,x^{100} - 1036445784547832237876640595707505410983090807676677135520\,x^{98} \\
& + 533837015752249926373527407719894367537526144242722071611\,x^{96} - 262411236036546428939754830663713879841188639254246112800\,x^{94} \\
& + 123009941999890981594901347549376901872919852536482759596\,x^{92} - 54946088094222415074923868907131504156814297011674689232\,x^{90} \\
& + 23367276860690374439228854250117919935095113052863062304\,x^{88} - 9453012643403839699679466429793260914280672208119845376\,x^{86} \\
& + 3634276715786121868379376615739335638720446535893073954\,x^{84} - 1326549658346537753319903603129595359737209739528831784\,x^{82} \\
& + 459234731987513437895445934604897059972726058956648623\,x^{80} - 150616870779581932614750233704151617804907976299834288\,x^{78} \\
& + 46744912221046840958190516813551431585403599874213187\,x^{76} - 13711379352993664885747467211479704107864688350142820\,x^{74} \\
& + 3796173249075149314798427405642706732626576036902975\,x^{72} - 990663606553452019597534435181574776372551412114424\,x^{70} \\
& + 243321958356782840335066886369163685773323611796322\,x^{68} - 56160391210859161239409475196656694794686765160520\,x^{66} \\
& + 12160380699908264255982216017748852755390183170749\,x^{64} - 2465804669311089875923551599192230985197067780928\,x^{62} \\
& + 467345574597279191417935577953101146677434755018\,x^{60} - 82623075679262282338712280412572542797753824600\,x^{58} \\
& + 13595674489955525524340729471909872755612531906\,x^{56} - 2077394398234615646635412883505857806901383408\,x^{54} \\
& + 294009564745232840525039045583216150370937175\,x^{52} - 38436894289240674520979141245223098136298540\,x^{50} \\
& + 4628117343019376629909138906287470601193419\,x^{48} - 511621006280790300281693808105419954445456\,x^{46} \\
& + 51746366889608011256879080565950193239242\,x^{44} - 4770503153868571545736175023627344829240\,x^{42} + 399222237929934842221582296016710370773\,x^{40} \\
& - 30190611718996719997279499140103595720\,x^{38} + 2052949134521983271740958944095504174\,x^{36} - 124838302461089261717953954583319096\,x^{34} \\
& + 6747306983434732062240755713796706\,x^{32} - 321933327388586659538894322123328\,x^{30} + 13456395449512186145796632312805\,x^{28} \\
& - 488492380416984991504401221388\,x^{26} + 15249896414378022807871956638\,x^{24} - 404788745877135098537461008\,x^{22} + 9016111948728461943888450\,x^{20} \\
& - 165924942927139623716008\,x^{18} + 2476737515475184799124\,x^{16} - 29320596366796081728\,x^{14} + 267701208900574320\,x^{12} - 1818326209216320\,x^{10} \\
& + 8749956775968\,x^{8} - 27750806784\,x^{6} + 51255936\,x^{4} - 41472\,x^{2} + 1
\end{aligned}
\tag{18}
$$

Figure A.3 – $[\mathcal{X}]M_{1260}$ calculé par `T2x(MnT(1260))`

```
> VoirLaTaille:=proc(L)
  local  m1,m2,j;
  ## L est une liste des entiers ##
  ## le procédure renvoie [a,b] où a est la taille binaire maximal, b est la somme de bits nécessaire
  #local L1,m1,m2,j;
  m1:=max(seq(ceil(log[2](max(2,abs(L[j])))),j=1..nops(L)));
  m2:=add(ceil(log[2](max(2,abs(L[j])))),j=1..nops(L));
  [m1,m2];
  end:
> M936:=MnT(936);
  M1260:=MnT(1260);
  M8640:=MnT(8640);
```

$$M936 := 1 - T_{48} - T_{72} + T_{120} + T_{144} - T_{192} - T_{216} + T_{264} + T_{288}$$

$$M1260 := -1 - T_{24} - T_{48} - T_{84} + T_{96} - T_{108} + T_{120} - T_{132} + T_{144} + T_{180} - T_{192} + 2\,T_{204} - T_{216} + T_{228} + T_{264} - T_{276} + T_{288}$$

$$M8640 := -1 - T_{576} + T_{1728} + T_{2304} \quad (16)$$

```
> K[1]:=PolynomialTools:-CoefficientList(T2x(M936),x):
  K[2]:=PolynomialTools:-CoefficientList(T2x(M1260),x):
  K[3]:=PolynomialTools:-CoefficientList(T2x(M8640),x):
> for h from 1 to 3 do VoirLaTaille(K[h]) od;
```

$$[197, 20484]$$
$$[197, 20475]$$
$$[1595, 1323350] \quad (17)$$

FIGURE A.4 – Calculer la taille binaire des coefficients de M_{1260}, M_{936}, M_{8640}.

```
> with(ChebUnit):
> a[1]:=16*sin(Pi/9)*sin(5*Pi/18)*sin(11*Pi/39)*sin(3*Pi/8)-3;
  a[2]:=16*sin(2*Pi/45)*sin(4*Pi/25)*sin(20*Pi/49)*sin(17*Pi/40)-1;
  a[3]:=48*cos(Pi/18)*cos(7*Pi/15)*cos(9*Pi/22)*cos(12*Pi/49)-1;
  a[4]:=16*cos(2*Pi/5)*cos(5*Pi/16)*cos(8*Pi/27)*cos(104*Pi/357)-1;
```

$$a_1 := 16 \sin\left(\frac{1}{9}\pi\right) \sin\left(\frac{5}{18}\pi\right) \sin\left(\frac{11}{39}\pi\right) \sin\left(\frac{3}{8}\pi\right) - 3$$

$$a_2 := 16 \sin\left(\frac{2}{45}\pi\right) \sin\left(\frac{4}{25}\pi\right) \sin\left(\frac{20}{49}\pi\right) \sin\left(\frac{17}{40}\pi\right) - 1$$

$$a_3 := 48 \cos\left(\frac{1}{18}\pi\right) \cos\left(\frac{7}{15}\pi\right) \cos\left(\frac{9}{22}\pi\right) \cos\left(\frac{12}{49}\pi\right) - 1$$

$$a_4 := 16 \cos\left(\frac{2}{5}\pi\right) \cos\left(\frac{5}{16}\pi\right) \cos\left(\frac{8}{27}\pi\right) \cos\left(\frac{104}{357}\pi\right) - 1 \quad (1)$$

```
> for i from 1 to 4 do t2p(combine(a[i])) od;
```

$$[T_{243} + T_{69} + T_{477} + T_{165} - T_{277} - T_{451} - T_{43} + T_{251} - 3,936]$$

$$[T_{8707} + T_{11677} + T_{4523} + T_{24907} - T_{16547} - T_{19517} - T_{3317} - T_{32747} - 1,88200]$$

$$[3\,T_{5989} - 3\,T_{10301} - 3\,T_{1579} + 3\,T_{5891} - 3\,T_{7606} + 3\,T_{8684} + 3\,T_{3196} - 3\,T_{4274} - 1,24255]$$

$$[T_{21211} + T_{23771} - T_{83509} - T_{128491} - T_{72619} - T_{75179} + T_{32101} - T_{77141} - 1,257040] \quad (2)$$

```
> for ii from 1 to 4 do shake(a[ii],12) od;
```

$$INTERVAL(-2.7067\ 10^{-9} ..-2.2379\ 10^{-9})$$

$$INTERVAL(5.3685\ 10^{-9} ..5.5133\ 10^{-9})$$

$$INTERVAL(3.8645\ 10^{-9} ..4.6696\ 10^{-9})$$

$$INTERVAL(8.889\ 10^{-10} ..1.2407\ 10^{-9}) \quad (3)$$

FIGURE A.5 – Expliquer $a_j, j = 1, \ldots, 4$ sous formes de Chebyshev et minorer leurs valeurs avec la commande `shake`.

```
> rec1((omega[h])*cos(x[h]*alpha+y[h]*beta+z[h]*gamma));
```

$$[\omega_h, \cos, \alpha\, x_h + \beta\, y_h + \gamma\, z_h] \quad (2)$$

FIGURE A.6 – Lire les coefficients d'une expression cosinus

```
> P1:=2*(cos(alpha[1])*cos(beta[1])-cos(alpha[2])*cos(beta[2]));
  P1ijk:=combine(subs(alpha[1]=i[1]*Pi/a,alpha[2]=i[2]*Pi/a,beta[1]=j[1]*
  Pi/b,beta[2]=j[2]*Pi/b,P1),trig);
  t2p(P1ijk);
  PourMinorer(P1ijk);
```

$$P1 := 2\cos(\alpha_1)\cos(\beta_1) - 2\cos(\alpha_2)\cos(\beta_2)$$

$$P1ijk := \cos\left(\frac{\pi\,(a\,j_1 - b\,i_1)}{a\,b}\right) + \cos\left(\frac{\pi\,(a\,j_1 + b\,i_1)}{a\,b}\right) - \cos\left(\frac{\pi\,(a\,j_2 - b\,i_2)}{a\,b}\right) - \cos\left(\frac{\pi\,(a\,j_2 + b\,i_2)}{a\,b}\right)$$

$$\left[\frac{1}{2}\,T_{a\,j_1 - b\,i_1} + \frac{1}{2}\,T_{a\,j_1 + b\,i_1} - \frac{1}{2}\,T_{a\,j_2 - b\,i_2} - \frac{1}{2}\,T_{a\,j_2 + b\,i_2},\, a\,b\right]$$

$$[8, 2] \qquad (4)$$

Figure A.7 – La forme de Chebyshev pour l'inégalité (5.22) et la valeur maximale de $\|.\|_T$, $m = 2$, $M = 8$

```
> P2:=(sin(gamma[2]))^2*(2*cos(alpha[1])*cos(beta[1])-2*cos(alpha[2])*cos
  (beta[2]))^2-4*(cos(gamma[2]))^2*((sin(gamma[2]))^2-(sin(alpha[2]))^2*(sin
  (beta[2]))^2);
  PourMinorer(P2);
```

$$P2 := \sin(\gamma_2)^2\,(2\cos(\alpha_1)\cos(\beta_1) - 2\cos(\alpha_2)\cos(\beta_2))^2 - 4\cos(\gamma_2)^2\,(\sin(\gamma_2)^2 - \sin(\alpha_2)^2\sin(\beta_2)^2)$$

$$[256, 16] \qquad (5)$$

Figure A.8 – La forme de Chebyshev pour l'inégalité (5.23), $m = 16$, $M = 256$

```
> polyP:=phi^2+4*u*v*phi+4*(u^2-w^2)*(v^2-w^2)/(1-w^2);
  P1:=subs(u=cos(alpha[1]),v=cos(beta[1]),w=cos(gamma[1]),PPP);
  P2:=subs(u=cos(alpha[2]),v=cos(beta[2]),w=cos(gamma[2]),PPP);
  RR:=(sin(gamma[1])*sin(gamma[2]))^4*resultant(P1,P2,phi):
  PourMinorer(RR);
```

$$polyP := \phi^2 + 4\,u\,v\,\phi + \frac{4\,(u^2 - w^2)\,(v^2 - w^2)}{-w^2 + 1}$$

$$P1 := \phi^2 + 4\cos(\alpha_1)\cos(\beta_1)\,\phi + \frac{4\,(\cos(\alpha_1)^2 - \cos(\gamma_1)^2)\,(\cos(\beta_1)^2 - \cos(\gamma_1)^2)}{-\cos(\gamma_1)^2 + 1}$$

$$P2 := \phi^2 + 4\cos(\alpha_2)\cos(\beta_2)\,\phi + \frac{4\,(\cos(\alpha_2)^2 - \cos(\gamma_2)^2)\,(\cos(\beta_2)^2 - \cos(\gamma_2)^2)}{-\cos(\gamma_2)^2 + 1}$$

$$[65552, 256] \qquad (6)$$

Figure A.9 – La forme de Chebyshev pour l'inégalité (5.26), $m = 256$, $M = 65552$

```
> vfoispolyP1:=v+2*v*cos(i*Pi/a)*cos(j*Pi/b);
  t2p(combine(vfoispolyP1,trig));
  PourMinorer(vfoispolyP1);
```

$$vfoispolyP1 := v + 2\,v\cos\left(\frac{\pi\,i}{a}\right)\cos\left(\frac{\pi\,j}{b}\right)$$

$$\left[v + \frac{1}{2}\,v\,T_{a\,j-b\,i} + \frac{1}{2}\,v\,T_{a\,j+b\,i},\ a\,b\right]$$

$$[6\,|v|,\ 2] \tag{7}$$

FIGURE A.10 – La forme de Chebyshev dans le lemme 5.20 où $\gamma = \frac{\pi}{2}$, $m = 2$, $M = 6|v|$

```
> PP1:=subs(u=cos(i*Pi/a),v=cos(j*Pi/b),w=cos(k*Pi/c),polyP);
  PP2:=simplify(4*v^2*(sin(k*Pi/c))^2*subs(phi=u/v,PP1));
  t2p(combine(PP2,trig));
  PourMinorer(PP2);
```

$$PP1 := \phi^2 + 4\cos\left(\frac{\pi\,i}{a}\right)\cos\left(\frac{\pi\,j}{b}\right)\phi + \frac{4\left(\cos\left(\frac{\pi\,i}{a}\right)^2 - \cos\left(\frac{\pi\,k}{c}\right)^2\right)\left(\cos\left(\frac{\pi\,j}{b}\right)^2 - \cos\left(\frac{\pi\,k}{c}\right)^2\right)}{-\cos\left(\frac{\pi\,k}{c}\right)^2 + 1}$$

$$\begin{aligned} PP2 := {} & 16\cos\left(\frac{\pi\,i}{a}\right)^2\cos\left(\frac{\pi\,j}{b}\right)^2 v^2 - 16\cos\left(\frac{\pi\,i}{a}\right)^2\cos\left(\frac{\pi\,k}{c}\right)^2 v^2 \\ & - 16\cos\left(\frac{\pi\,i}{a}\right)\cos\left(\frac{\pi\,j}{b}\right)\cos\left(\frac{\pi\,k}{c}\right)^2 u\,v - 16\cos\left(\frac{\pi\,j}{b}\right)^2\cos\left(\frac{\pi\,k}{c}\right)^2 v^2 + 16\cos\left(\frac{\pi\,k}{c}\right)^4 v^2 \\ & + 16\cos\left(\frac{\pi\,i}{a}\right)\cos\left(\frac{\pi\,j}{b}\right) u\,v - 4\cos\left(\frac{\pi\,k}{c}\right)^2 u^2 + 4\,u^2 \end{aligned}$$

$$\begin{aligned} & \Big[-u^2\,T_{2\,a\,b\,k} + 2\,u\,v\,T_{c\,(a\,j-b\,i)} + 2\,u\,v\,T_{c\,(a\,j+b\,i)} - u\,v\,T_{2\,a\,b\,k-a\,c\,j-b\,c\,i} - u\,v\,T_{2\,a\,b\,k-a\,c\,j+b\,c\,i} \\ & - u\,v\,T_{2\,a\,b\,k+a\,c\,j-b\,c\,i} - u\,v\,T_{2\,a\,b\,k+a\,c\,j+b\,c\,i} - v^2\,T_{2\,a\,(b\,k-c\,j)} - v^2\,T_{2\,a\,(b\,k+c\,j)} \\ & - v^2\,T_{2\,b\,(a\,k-c\,i)} - v^2\,T_{2\,b\,(a\,k+c\,i)} + v^2\,T_{2\,c\,(a\,j-b\,i)} + v^2\,T_{2\,c\,(a\,j+b\,i)} + v^2\,T_{4\,a\,b\,k} + 2\,u^2 + 2\,v^2, \\ & c\,a\,b\Big] \end{aligned}$$

$$\left[4\,|u|^2 + 16\,|u\,v| + 16\,|v|^2,\ 1\right] \tag{8}$$

FIGURE A.11 – La forme de Chebyshev dans le lemme 5.20 où $\gamma \neq \frac{\pi}{2}$, $m = 1$, $M = 4|u|^2 + 16|uv| + 16|v|^2$

```
> dt:=discrim(polyP,phi);
  dt1:=subs(u=cos(i*Pi/a),v=cos(j*Pi/b),w=cos(k*Pi/c),16*(u^2*v^2-u^2-v^2+
  w^2));
  t2p(combine(dt1,trig));
  PourMinorer(dt1);
```

$$dt := \frac{16\,w^2\left(u^2\,v^2 - u^2 - v^2 + w^2\right)}{w^2 - 1}$$

$$dt1 := 16\cos\left(\frac{\pi\,i}{a}\right)^2\cos\left(\frac{\pi\,j}{b}\right)^2 - 16\cos\left(\frac{\pi\,i}{a}\right)^2 - 16\cos\left(\frac{\pi\,j}{b}\right)^2 + 16\cos\left(\frac{\pi\,k}{c}\right)^2$$

$$\left[T_{2\,c\,(a\,j-b\,i)} + T_{2\,c\,(a\,j+b\,i)} - 2\,T_{2\,b\,c\,i} - 2\,T_{2\,a\,c\,j} - 4 + 4\,T_{2\,a\,b\,k},\ c\,a\,b\right]$$

$$[24,\ 1] \tag{9}$$

FIGURE A.12 – La forme de Chebyshev de $\delta_{\alpha,\beta,\gamma}$ dans l'inégalité (5.21), $m = 1$, $M = 24$

```
> polyPtilde1:=2*phi+4*cos(i*Pi/a)*cos(j*Pi/b);
  polyPtilde2:=(sin(k*Pi/c))^2*phi^2+4*(sin(k*Pi/c))^2*cos(i*Pi/a)*cos(j*Pi/b)
  *phi+(cos(2*i*Pi/a)-cos(2*k*Pi/c))*(cos(2*j*Pi/b)-cos(2*k*Pi/c));
```

$$polyPtilde1 := 2\,\phi + 4\cos\left(\frac{\pi\, i}{a}\right)\cos\left(\frac{\pi\, j}{b}\right)$$

$$polyPtilde2 := \sin\left(\frac{\pi\, k}{c}\right)^2 \phi^2 + 4\sin\left(\frac{\pi\, k}{c}\right)^2 \cos\left(\frac{\pi\, i}{a}\right)\cos\left(\frac{\pi\, j}{b}\right)\phi + \left(\cos\left(\frac{2\,\pi\, i}{a}\right) - \cos\left(\frac{2\,\pi\, k}{c}\right)\right)\left(\cos\left(\frac{2\,\pi\, j}{b}\right) - \cos\left(\frac{2\,\pi\, k}{c}\right)\right) \quad \textbf{(13)}$$

```
> for l from 0 to 1 do m[l]:=t2p(combine(coeff(polyPtilde1,phi,l),trig)) od;
  for l from 0 to 2 do n[l]:=t2p(combine(coeff(4*polyPtilde2,phi,l),trig)) od;
```

$$m_0 := \left[T_{-a\,j+b\,i} + T_{a\,j+b\,i},\, a\,b\right]$$

$$m_1 := 2$$

$$n_0 := \left[T_{-2\,c\,(a\,j-b\,i)} + T_{2\,c\,(a\,j+b\,i)} - T_{-2\,b\,(a\,k-c\,i)} - T_{2\,b\,(a\,k+c\,i)} - T_{-2\,a\,(b\,k-c\,j)} - T_{2\,a\,(b\,k+c\,j)} + T_{4\,a\,b\,k} + 2,\, c\,a\,b\right]$$

$$n_1 := \left[-T_{-2\,a\,b\,k-a\,c\,j+b\,c\,i} - T_{2\,a\,b\,k-a\,c\,j+b\,c\,i} - T_{-2\,a\,b\,k+a\,c\,j+b\,c\,i} - T_{2\,a\,b\,k+a\,c\,j+b\,c\,i} + 2\,T_{-c\,(a\,j-b\,i)} + 2\,T_{c\,(a\,j+b\,i)},\, c\,a\,b\right]$$

$$n_2 := \left[2 - T_{2\,k},\, c\right] \quad \textbf{(14)}$$

Figure A.13 – Les formes de Chebyshev dans $\widetilde{P}_{\alpha,\beta,\gamma}$, lemme 5.9

Table des figures

1 Plan de Thèse. 14

2.1 Stratégie "diviser pour régner" 45

5.1 Les nœuds $\overline{5}_2, 5_4, \overline{4}_1$ avec leurs diagrammes de Chebyshev et trajectoires de billiard 89
5.2 Deux types de croisement 89

A.1 Les commandes du `ChebUnit` : `ChebTn, ChebUn, p2t, t2p, T2x, x2T, mulT` et `MnT` 106
A.2 Les commandes `MnT` et `mulT` appliquées dans l'exemple 4.5 107
A.3 $[\mathcal{X}]M_{1260}$ calculé par `T2x(MnT(1260))` 109
A.4 Calculer la taille binaire des coefficients de M_{1260}, M_{936}, M_{8640}. 110
A.5 Expliquer $a_j, j = 1, \ldots, 4$ sous formes de Chebyshev et minorer leurs valeurs avec la commande `shake`. 110
A.6 Lire les coefficients d'une expression cosinus 110
A.7 La forme de Chebyshev pour l'inégalité (5.22) et la valeur maximale de $\|.\|_T$, $m = 2$, $M = 8$ 111
A.8 La forme de Chebyshev pour l'inégalité (5.23), $m = 16$, $M = 256$ 111
A.9 La forme de Chebyshev pour l'inégalité (5.26), $m = 256$, $M = 65552$ 111
A.10 La forme de Chebyshev dans le lemme 5.20 où $\gamma = \frac{\pi}{2}$, $m = 2$, $M = 6|v|$ . . 112
A.11 La forme de Chebyshev dans le lemme 5.20 où $\gamma \neq \frac{\pi}{2}$, $m = 1$, $M = 4|u|^2 + 16|uv| + 16|v|^2$ 112
A.12 La forme de Chebyshev de $\delta_{\alpha,\beta,\gamma}$ dans l'inégalité (5.21), $m = 1$, $M = 24$ . . 112
A.13 Les formes de Chebyshev dans $\widetilde{P}_{\alpha,\beta,\gamma}$, lemme 5.9 113

Liste des Algorithmes

2.10 Calculer T_n 37
2.15 Multiplication de formes de Chebyshev par le Doublage 40
2.19 Division euclidienne de formes de Chebyshev 42
2.31 Calculer la forme de Chebyshev 51
2.35 Calculer X^m 54
2.36 Développer une forme de Chebyshev 55

3.14 Calculer M_n dans la base des polynômes unitaires de Chebyshev 65

4.9 Déterminer le signe de F par l'isolation 72
4.15 `SIGNE` pour déterminer le signe de $F = f(2\cos\frac{k\pi}{n})$ 76
4.31 Calculer P_f 84

Bibliographie

[AM10] A Arnold et M Monagan : A high-performance algorithm for calculating cyclotomic polynomials. *In Proceedings of the 4th International Workshop on Parallel and Symbolic Computation*, pages 112–120, 2010.

[Bac03] G. Bachman : On the coefficients of ternary cyclotomic polynomials. *Journal of Number Theory*, 100:104–116, 2003.

[Ban95] A. S. Bang : Om lignigen $\phi_n(x) = 0$. *Tidsskrift for Math.*, 1895.

[Bas97] G. Baszenski : Fast polynomial multiplication and convolutions related to the discrete cosine transform. *Linear Algebra and its Applications*, 252 (1-3):–25, 1997.

[BC12] A. Bayad et I. N. Cangul : The minimal polynomial of $2\cos\frac{\pi}{q}$ and Dickson polynomials. *Applied Mathematics and Computation*, 218(13):7014–7022, Mars 2012.

[BCG+14] A. Bostan, F. Chyzak, M. Giusti, R. Lebreton, B. Salvy et É. Schost : Algorithmes efficaces en Calcul Formel. Notes du cours 2-22 du MPRI, 2014.

[Ben12] A. Benoît : *Algorithmique semi-numérique rapid des séries de Tchebychev*. Thèse de doctorat, Ecole polytechnique, 2012.

[Ber14] D. Bertrand : *Théorie des nombres*. UPMC, Cours de master de mathématiques (m1) édition, 2ème semestre 2013-2014 2014.

[BFSS05] A. Bostan, P. Flajolet, B. Salvy et É. Schost : Fast computation of special resultants. *Journal of Symbolic Computation*, 41:1–29, 2005.

[BGVPS05] A. Bostan, L. Gonzalez-Vega, H. Perdry et É. Schost : From Newton sums to coefficients : complexity issues in characteristic p. *In Proceedings MEGA'05*, 2005.

[BLPR15] Y. Bouzidi, S. Lazard, M. Pouget et F. Rouillier : Separating linear forms and Rational Univariate Representations of bivariate systems. *Journal of Symbolic Computation*, 68:84–119, 2015.

[Bos95] W. Bosma : *Computation of cyclotomic Polynomial with Magma*, volume 325 de *Mathematics and its applications*, chapitre 15, pages 213–225. Springer Netherlands, 1995.

[BPR06] S. Basu, R. Pollack et F-M. Roy : *Algorithms in Real Algebraic Geometry*, volume 10 de *Algorithms and Computation in Mathematics*. Springer-Verlag, 2006.

[Bre75] R. Brent : Multipleprecision zero-finding methods and the complexity of elementary function evaluation. *Analytic Computational Complexity (edited by J. F. Traub), Academic Press, New York*, pages 151–176, 1975.

[Bre76] R. BRENT : Fast multiple precision evaluation of elementary functions. *Journal of the Association for Computing Machinery*, 23(2):242–251, 1976.

[BSS08] A. BOSTAN, B. SALVY et É. SCHOST : Power series composition and chage of basis. *In ISSAC'08*, pages 269–276. ACM, 2008.

[BSS10] A. BOSTAN, B. SALVY et É. SCHOST : Fast conversion algorithms for orthogonal polynomials. *Linear Algebra and its Applications*, 431(1):249–258, 2010.

[CC12] Z. CAO et H. CAO : On fast division algorithm for polynomials using Newton iteration. *In* B. LIU, M. MA et J. CHANG, éditeurs : *Lecture Notes in Computer Science*, volume Volume 7473, pages 175–180. Third International Conference, ICICA 2012, Chengde, China, September 14-16, 2012, 2012.

[Czi12] S. CZIRBUSZ : Comparing the computation of Chebyshev polynomials in computer angebra systems. *Annales Univ. Sci. Budapest., Sect. Comp.*, 36:23–39, 2012.

[dB87] C. de BOOR : B-form basics. *In Geometric modeling : Algorithm and New Trends*, pages 131–148. SIAM, Philadenphia, 1987.

[Dem08] M. DEMAZURE : *Cours d'Algèbre.* Cassini, 2008.

[DHJR97] J.R. DRISCOLL, D.M. HEALY JR et D.N. ROCKMORE : Fast discrete polynomial transforms with applications to data analysis for distance transitive graphs. *SIAM J. Comput.*, 26(4):1066–1099, 1997.

[Erd46] P. ERDOS : On the coefficients of cyclotomic polynomials. *Bull. Amer. Math. Soc*, 1946.

[Fru95] M. FRUMKIN : A fast algorithm for expansion over spherical harmonics. *Appl. Algebra Eng. Commun. Comput.*, 6(6):333–343, 1995.

[GG13] J. von zur. GATHEN et J. GERHARD : *Modern Computer Algebra.* Cambridge University Press, 03 édition, juin 2013.

[Gio12] P. GIORGI : On polynomial multiplication in Chebyshev basis. *IEEE Transactions on Computers*, 61(6):780–789, 2012.

[GT91] A. GRYTCZUK et B. TROPAK : A numerical method for the determination of the cyclotomic polynomial coefficients. *In Computational Number Theory.* Proceedings of the Colloquium on Computational Number Theory Held at Kossuth Lajos University, Debrecen (Hungary), 1991.

[HN11] W. B. HART et A. NOVOCIN : Practical divide-and-conquer algorithms for polynomial arithmetic. *In Lecture Notes in Computer Science*, volume 6885, pages 200–214. CASC'11 Proceedings of the 13th international conference on Computer algebra in scientific computing, september 2011.

[Hsi84] H. J. HSIAO : On factorization of Chebyshev's polynomial of the first kind. *Bulletin of the Institute of Mathematics Academia Sinica*, 12(1):89–94, 1984.

[HW08] G. H. HARDY et E. M. WRIGHT : *An Introduction to the Theory of numbers.* Oxford University Press, 2008.

[Knu97] D. KNUTH : *The art of Computer Programming*, volume Volume 2 : Seminumeraical Algorithms. Addison-Wesley, Reading, 1997.

[Kos00] Y. KOSHIBA : On the calculations of the coefficients of the cyclotomic polynomials II. *Rep. Fac. Sci., Kagoshima Univ.*, 33:55–59, 2000.

[KP11] P.-V. KOSELEFF et D. PECKER : Chebyshev knot. *Journal of Knot theory and its ramifications*, 20(4):575–593, 2011.

[KPR10] P-V. KOSELEFF, D. PECKER et F. ROUILLIER : The first rational Chebyshev knots. *Journal of Symbolic Computation*, 45:1341–1358, 2010.

[KPRT] P-V. KOSELEFF, D. PECKER, F. ROUILLIER et C. TRAN : Computing Chebyshev knot diagrams. à soumettre bien tôt.

[KRT15] P-V. KOSELEFF, F. ROUILLIER et C. TRAN : On the sign of a trigonometric expression. pages 259–266. ISSAC'15 (Bath, UK), ACM New York, juillet 2015.

[Leh36] E. LEHMER : On the magnitude of coefficients of the cyclotomic polynomials. *Bull. Amer. Math. Soc 42*, 1936.

[LL96] T. Y. LAM et K. H. LEUNG : On the cyclotomic polynomial $\phi_{pq}(x)$. *Amer. Math. Monthly*, Aug.-Sept.:562–564, 1996.

[LRC07] G. LEIBON, D.N. ROCKMORE et G. CHIRIKJIAN : A fast Hermite transform with applications to protein structure determinaiton. *In SNC'07*, pages 117–124. ACM, 2007.

[McD99] I. G. MCDONALD : *Symmetric Functions and Hall Polynomials*, volume Oxford Mathematical Monographs. Oxford University Press, 1999.

[MH03] J. C. MASON et D. C. HANDSCOMB : *Chebyshev Polynomials.* Chapman and Hall/CRC, 2003.

[Mig83] A. MIGNOTTI : Aur theorie der koeffisienten des n-ten kreisteilungspolynome. *In Z. B. der Math. - Naturwiss, Classe der Kaiserlichen Akademie der Wissenschaften, Wien, 87, 7-14*, 1883.

[Mil76] G. L. MILLER : Riemann's hypothesis and tests for primality. *Journal of computer and system sciences*, 13:30–317, 1976.

[MSW14] K. MEHLHORN, M. SAGRALOFF et P. WANG : From approximate factorization to root isolation with application to cyclindrical algebraic decomposition. *Journal of Symbolic Computation*, 66, 2014.

[Mul06] J-M. MULLER : *Elemenary functions, algorithms and implementations.* Birkhäuser Boston, Basel, Berlin, 2006.

[Mur96] K. MURASUGI : *Knot Theory and its Applications.* Birkhäuser, 1996.

[Mye93] G. MYERSON : Rational products of cosine of rational angles. *Aequationes Mathematicae, Univ. of Waterloo*, 45:70–82, 1993.

[OLBC10] F. W. J. OLVER, D. W. LOZIER, R. F. BOISVERT et C. W. CLARK : *NIST Handbook of mathematical functions.* Cambridge University Press, 2010.

[Pan98] V. Y. PAN : New fast algorithms for polynomial interpolation and evaluation on the Chebyshev node set. *Comput. Math. Applic.*, 35(3):125–129, 1998.

[Pan00] V. Y. PAN : New techniques for the computation of linear recurrence coefficients. *Finite Fields and their Applications*, 6(1):93–118, 2000.

[Pan01] V. Y. PAN : *Structured matrices and polynomials : Unified superfast algorithms.* Birkhäuser, 2001.

[PST98] D. POTTS, G. STEIDL et M. TASCHE : Fast algorithms for discrete polynomial transforms. *Math. Comput.*, 67(224):1577–1590, 1998.

[Riv90] T. J. RIVLIN : *Chebyshev Methods in Numerical Approximation.* John Wiley & Sons, Inc., 1990.

[Rou99] F. ROUILLIER : Solving zero-dimensional systems through the Rational Univariate Representation. *Applicable Algebra in Engineering, Communication and Computing*, 9(5):433–461, Mai 1999.

[Sch] I. SCHUR : Discussion par courier avec Landau. 1935.

[SM16] M. SAGRALOFF et K. MEHLHORN : Computing real roots of real polynomials. *Journal of Symbolic Computation*, 73:46–86, March-April 2016.

[Val06] A. VALIBOUZE : Compute the minimal polynomial of $\cos\frac{2\pi}{n}$ as a resolvent. LSTA 2006-3, Mars 2006.

[Vas90] V. A. VASILIEV : Cohomology of knot spaces. *Theory of singularities and Its applications, Advance Soviet Mahts*, 1, 1990.

[WZ93] W. WATKINS et J. ZEITLIN : The minimal polynomial of $\cos\frac{2\pi}{n}$. *The American Mathematical Monthly*, 100(5):471–474, May 1993.

Index des notations

abréviation
- $\mathrm{Abr}_n(f)$, $\mathrm{Abr}(f)$, 78
- lc, 15, 34

application
- $\mathrm{Abr}_n(f)$, $\mathrm{Abr}(f)$, 78
- $\mathcal{D}$, 26
- REV, 23, 24

auteur
- Arnold & Monagan, 60
- Bézout, 20, 28
- Bang, 60
- Baszenski, 39
- Bayard & Cangul, 28
- Benoît, 39, 50
- Berstein, 47
- Bostan, 53
- Brent, 70, 100
- Dickson, 15
- Erdos, 60
- Euler, 92
- Fourier, 35
- Gauss, 23, 81
- Giorgi, 39
- Hölder, 25, 84
- Hart & Novocin, 47
- Horner, 47
- Hsiao, 28, 29
- Karatsuba, 35
- Lehmer, 60
- Mignotte, 65
- Migotti, 60
- Muller, 70
- Myerson, 77, 97
- Newton, 83
- Ramanujan, 83
- Rivlin, 28, 29
- Salvy, 49
- Schost, 49
- Schur, 60
- Sylvester, 74
- Valibouze, 63
- Watkin & Zeitlin, 28

base de $\mathbb{Z}[x]$, 16

base de Gröbner, 90

borne de séparation
- $\mathrm{Sep}(f)$, 98

complexité
- arithmétique, 35
- binaire, 35
- quasi-linaire, 35
- quasi-quadratique, 35

courbe de Chebyshev, $\mathcal{C}\ (a, b, c, \phi)$, 87

Diviser pour Régner, 44

famille
- $\mathcal{T}$,$\mathcal{U}$, 15
- $\mathcal{X}$, 16

fonction
- arithmétique, 19
- arithmétique multiplicative, 19
- asymptotiquement positive, 34
- d'Euler, $\varphi(n)$, 20, 28
- de Möbius, $\mu(n)$, 21
- nombre de diviseurs premiers, $\omega(n)$, 25, 59
- nombre de diviseurs, $d(n)$, 59

forme de Chebyshev, 34

formule
- d'Euler, 19, 30
- d'inversion de Möbius, 22
- de Newton, 24

Maple, 67, 71, 77, 78, 98

matrice de Sylvester, 74

nœud
- à deux ponts, 90
- de Chebyshev, 87

nombre dyadique, 95

polynôme
$\mathbf{T}_n, \mathbf{U}_n$, 15, 16
M_f, 81
$M_n(x)$, 28
P_f, 81
caractéristique, 90
cyclotomique inversé, $\Psi_n(x)$, 61
cyclotomique, $\Phi_n(x)$, 22, 28
de Chebyshev, 15
de Dickson, 15
unitaire de Chebyshev, 15
unitaire de Chebyshev de première espèce, T_n, 15
unitaire de Chebyshev de seconde espèce, U_n, 15

résultant, $\mathrm{Res}(A, B)$, 75
radical d'un entier, 59
RUR, 90

série formelle, 24
somme
de Newton, $S_m(P)$, 23
de Ramanujan, $c_n(m)$, $S_m(\Phi_n)$, 24
symbole
$[B_j]f$, 34
C_A, pour la complexité arithmétique, 35
C_B, pour la complexité binaire, 35
$\Delta_{\alpha,\beta,\gamma}$, 97
$\mathcal{M}$, temps de multiplication, 35
$\|f\|_\infty$, la norme infinie, 34
$\|f\|_T$, la norme Chebyshev, 34
$\|f\|_1$, la norme Manhattan, 34
$\mathcal{O}, \widetilde{\mathcal{O}}$, pour la complexité, 34
ord, 20
$\widetilde{P}_{\alpha,\beta,\gamma}$, $P_{\alpha,\beta,\gamma}$, 92
$\prec$, 96
Quo, Rem, 27
τ, la taille binaire, 33
$\tau_{\mathcal{B}}(f)$, $\tau(f)$, $\tau_{\mathcal{T}}(f)$, la taille binaire des coefficients, 34
$\mathrm{Newton}(P)$, 24
$\mathrm{Syl}(A, B)$, 74
$\widetilde{R}_{a,b,c}$, 90
$E_\eta(x, y)$, 91
$P_n(t, x)$, $Q_n(t, s)$, 88
$R_{a,b,c}$, 90

taille binaire, 33
trajectoire de billiard, 88
valeurs critiques, 90