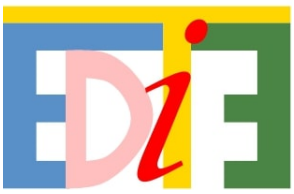
Certified Polynomial Optimization Based on Exact Sum of Squares Decompositions

Trung-Hieu Vu



THÈSE DE DOCTORAT DE
SORBONNE UNIVERSITÉ

Spécialité

Informatique

École Doctorale Informatique, Télécommunications et Électronique (Paris)

Présentée par

Trung Hieu VU

Pour obtenir le grade de

DOCTEUR de SORBONNE UNIVERSITÉ

Certified Polynomial Optimization Based on Exact Sum of Squares Decompositions

Thèse dirigée par

Victor MAGRON et Mohab SAFEY EL DIN

soutenue le 9 décembre 2022

après avis des **rapporteurs** :

M. Bernard MOURRAIN	Directeur de recherche, Inria Sophia Antipolis Méditerranée
M. Tien Son PHAM	Professor, Dalat University

devant le **jury** composé de :

M. Stef GRAILLAT	Professeur, Sorbonne Université
M. Victor MAGRON	Chercheur, Laboratoire d'Analyse et d'Architecture des Systèmes
M. Bernard MOURRAIN	Directeur de recherche, Inria Sophia Antipolis Méditerranée
M. Simone NALDI	Maître de conférences, Université de Limoges
M. Tien Son PHAM	Professor, Dalat University
M. Mohab SAFEY EL DIN	Professeur, Sorbonne Université
Mme. Lihong ZHI	Professor, Chinese Academy of Sciences

Contents

List of Acronyms and Symbols

SOS	sum of squares
SOHS	sum of Hermitian squares
SDP	semi-definite program
$\mathbb{N}$	$\{0, 1, 2, \dots\}$
$\mathbb{Z}$	the set of integer numbers
$\mathbb{Q}$	the field of rational numbers
$\mathbb{Q}_+$	the set of non-negative rational numbers
$\mathbb{R}$	the field of real numbers
$\mathbb{C}$	the field of complex numbers
$\boldsymbol{i}$	the imaginary unit
$\mathscr{C}$	the complex unit circle
$\boldsymbol{x}$	$(x_1, \dots, x_n)$
$\mathbb{K}[\boldsymbol{x}]$	the ring of polynomials with coefficients over $\mathbb{K}$
$\mathbb{K}(\boldsymbol{x})$	the field of rational functions with coefficients over $\mathbb{K}$
$\langle g_1, \dots, g_r \rangle$	the ideal generated by $g_1, \dots, g_r$
$\frac{\partial f}{\partial x_i}$	the partial derivative of f w.r.t. the variable x_i
$\mathcal{I}_{\text{grad}}(f)$	the gradient ideal of the polynomial f
$V(\mathcal{I})$	the algebraic variety associated to the ideal $\mathcal{I}$
$V_{\text{grad}}(f)$	the gradient variety of the polynomial f
$I(\mathcal{V})$	the ideal of all polynomials in $\mathbb{K}[\boldsymbol{x}]$ that vanish on the variety $\mathcal{V}$
$\sqrt{\mathcal{I}}$	the radical of the ideal $\mathcal{I}$
$\#S$	the cardinality of the set S
$\text{ht}(f)$	the height of polynomial f with (Gaussian) rational coefficients
I_d	the identity matrix of size d
$Q \succeq 0$	Q is a positive semi-definite matrix
$Q \succ 0$	Q is a positive definite matrix
$\text{tr}(Q)$	the trace of the matrix Q
$\mathscr{H}[z]$	the set of trigonometric polynomials in variable z with complex coefficients
$\mathscr{H}(\mathbb{Z})[z]$	the set of polynomials in $\mathscr{H}[z]$ with Gaussian integer coefficients

CHAPTER 1

Introduction en français

En optimisation polynomiale, il est bien connu que le calcul de l'infimum d'un polynôme sur un ensemble semi-algébrique est NP-dur. Pour contourner ce problème de calcul, on peut utiliser des hiérarchies de relaxations semi-définies et la théorie des sommes de carrés (SOS) de polynômes, dans laquelle chaque programme semi-défini peut être résolu, jusqu'à une précision arbitraire, en temps polynomial. Dans ce cadre, la non-négativité des polynômes est remplacée par la propriété d'être SOS. Par conséquent, l'évaluation de la non-négativité des polynômes basée sur les décompositions SOS est un problème d'actualité dans le domaine de l'optimisation polynomiale.

Les certificats de non-négativité sont généralement abordés par le biais du calcul de décompositions SOS qui reposent sur des solveurs numériques efficaces numériques efficaces pour la programmation semi-définie. Par conséquent, les certificats obtenus de cette manière sont *approximatifs* et donc non exacts. Pour certaines applications critiques, il est important de calculer réellement des certificats *exacts* de non-négativité.

Le but de cette thèse est de calculer des certificats exacts de non-négativité pour des polynômes basés sur des décompositions SOS avec des coefficients rationnels. Les certifications utilisant SOS font face aux deux difficultés suivantes. Premièrement, en raison du fait que l'ensemble des polynômes non-négatifs est significativement plus grand que celui des polynômes SOS, les certifications basées sur les décompositions SOS ne peuvent pas être appliquées à tous les polynômes non négatifs. Deuxièmement, il existe des polynômes à coefficients rationnels qui sont SOS à coefficients réels mais qui ne sont pas SOS à coefficients rationnels.

Dans cette thèse, nous fournissons des algorithmes symboliques pour calculer les décompositions SOS modulo l'idéal gradient des polynômes multivariés réels non négatifs sous une condition de généricité. Ces algorithmes peuvent traiter un large éventail de problèmes qui sont hors de portée des algorithmes les plus avancés. Nous calculons également les sommes des décompositions des carrés hermitiens pour les polynômes trigonométriques complexes univariés qui sont positifs sur le cercle unité avec des gaussiens. De plus, nous analysons la complexité binaire de ces algorithmes et déduisons les limites de taille binaire de ces certificats. Enfin, nous implémentons ces algorithmes dans le système de calcul formel MAPLE et l'environnement de programmation JULIA et évaluons leurs performances sur quelques benchmarks standards.

1.1 Thèmes de recherche et motivations

Nous désignons par $\mathbb{N}$ l'ensemble des nombres naturels, par $\mathbb{Z}$ l'ensemble des entiers, et par $\mathbb{Q}$, $\mathbb{Q}_+$, $\mathbb{R}$ et $\mathbb{C}$ les corps des nombres rationnels, rationnels non négatifs, réels et complexes, respectivement. Soit $\boldsymbol{x}$ le vecteur de n variables $(x_1, \ldots, x_n)$. Soit $\mathbb{K}$ un un corps, nous désignons par $\mathbb{K}[\boldsymbol{x}]$ l'anneau de polynômes avec le corps de base $\mathbb{K}$ et les variables $\boldsymbol{x}$.

1.1.1 Certificats exacts pour les polynômes multivariés réels

Un polynôme f dans $\mathbb{R}[\boldsymbol{x}]$ de degré d est *non-négatif* sur $\mathbb{R}^n$ s'il ne prend que des valeurs non négatives. Fournir des conditions vérifiables ou une procédure pour vérifier la non-négativité des polynômes est une question cruciale dans la théorie de l'optimisation polynomiale [64, 65]. Il existe plusieurs questions d'actualité concernant la non-négativité des polynômes telles que [65] : Comment *decider* de la non-négativité de f? Est-il possible de *certifier* la non-négativité de f? Quelle est la *complexité* de la décision de la non-négativité de f? Quelle est la *structure* de l'ensemble des polynômes non-négatifs? Nos recherches dans cette thèse se concentrent sur la certification et la et les problèmes de complexité.

Non-négativité et sommes des carrés. La non-négativité de f découle du fait que f peut être décomposé comme une *somme de carrés* (SOS) de polynômes, à savoir

$$f = q_1^2 + \cdots + q_r^2, \tag{1.1}$$

où $q_1, \ldots, q_r$ sont dans $\mathbb{R}[\boldsymbol{x}]$, pour un certain $r \in \mathbb{N} \setminus \{0\}$. Le membre de droit de (1.1) est appelé une *décomposition SOS* de f et fournit un *certificat* de non-négativité pour f.

La relation entre les propriétés de non-négativité et de SOS des polynômes a été étudiée depuis la fin du 19ème siècle, conduisant au 17ème problème de Hilbert. Le théorème de Hilbert, dans [34], dit que la non-négativité et la propriété SOS des polynômes homogènes sont équivalentes si et seulement si $n = 2$, $d = 2$, ou $(n, d) = (3, 4)$. Il découle du théorème de Hilbert que, pour les autres cas, l'ensemble des polynômes non-négatifs contient strictement l'ensemble des polynômes SOS. En 1967, Motzkin a donné le premier polynôme explicite p_M non négatif et pourtant non SOS. [60],

$$p_M := x_1^2 x_2^2 (x_1^2 + x_2^2 - 3) + 1. \tag{1.2}$$

Quantitativement, Blekherman [15] a prouvé que pour tout degré fixe $d \geq 4$, si le nombre de variables est suffisamment grand alors l'ensemble des polynômes non-négatifs est significativement plus grand que celui des polynômes SOS. Notons que, l'ensemble des polynômes non négatifs et celui des polynômes SOS sont significativement plus

grands. Notons que ces deux ensembles définissent tout deux des cônes convexes and que tout deux sont de dimension complète dans $\mathbb{R}_d[\boldsymbol{x}]$, l'espace de tous les polynômes de degré au plus égal à d [72].

Rappelons que la non-négativité de f peut également être certifiée si f peut être décomposé en une somme de carrés de *fonctions rationnelles*, c'est-à-dire que chaque q_ℓ dans (1.1) est une fraction de deux polynômes non triviaux dans $\mathbb{R}[\boldsymbol{x}]$,

$$f = \left(\frac{u_1}{v_1}\right)^2 + \cdots + \left(\frac{u_r}{v_r}\right)^2, \tag{1.3}$$

Le 17ème problème de Hilbert en 1900 [35] était le suivant : "Pour tout $f \in \mathbb{R}[\boldsymbol{x}]$, *est-il vrai que si f est non négatif sur $\mathbb{R}^n$ alors f est une somme de carrés de fonctions rationnelles?*" Artin a donné une réponse affirmative [2] en 1927. Par exemple, la non-négativité du polynôme de Motzkin p_M peut être certifiée en en l'écrivant comme une SOS de fonctions rationnelles:

$$p_M = \frac{1}{x_1^2 + x_2^2}(x_1^2(1-x_2)^2 + x_2^2(1-x_1)^2 + x_1^2x_2^2(x_1^2 + x_2^2 - 2)^2).$$

Nous nous intéressons également à la question du nombre maximal de carrés r et le degré maximal de u_ℓ, v_ℓ dans la décomposition (1.3). En 1967, Pfister [68] a prouvé que si $f \in \mathbb{R}[\boldsymbol{x}]$ est non-négatif alors f est une somme de 2^n carrés de fonctions rationnelles, ce nombre ne dépendant que du du nombre de variables. Lombardi, Perrucci et Roy ont prouvé en 2014 que si f est non-négatif alors f peut s'écrire comme une somme de carrés de fonctions rationnelles de degré au plus $2^{2^{2^{d^{4^n}}}}$ en [48].

La relation entre la non-négativité et the propriété SOS des polynômes peut être trouvée plus en détail dans les monographies [46, 58, 71], ou dans le article de synthèse [65].

Sommes de carrés à coefficients rationnels. Soit $f \in \mathbb{Q}[\boldsymbol{x}]$ un polynôme à coefficients rationnels. On dit que f est somme (pondérée) de carrés à coefficients rationnels si f peut s'écrire sous la forme

$$f = c_1 s_1^2 + \cdots + c_r q_r^2,$$

où $c_1, \ldots, c_r$ sont dans $\mathbb{Q}$, et $q_1, \ldots, q_r$ sont dans $\mathbb{Q}[\boldsymbol{x}]$, pour un certain nombre d'entiers positifs r.

Tout polynôme univarié non négatif $f \in \mathbb{Q}[x_1]$ à coefficients rationnels peut être décomposé en tant que somme pondérée de carrés à coefficients rationnels [43, 70]. Landau [43] a prouvé en 1905 que le nombre maximal de carrés est au plus égal à huit, c'est-à-dire que $r \leq 8$. Ce résultat a été amélioré par Pourchet [70] en 1971 par $r \leq 5$.

Il y a plus de dix ans, Sturmfels a soulevé la question de savoir si un polynôme à coefficients rationnels est nécessairement une SOS de polynômes à coefficients rationnels. Scheiderer a donné une réponse négative dans [79], où il a construit des familles de polynômes homogènes explicites à coefficients rationnels qui sont SOS avec des coefficients réels mais pas avec des coefficients rationnels. Il a souligné explicitement que le polynôme

$$p_S = x_1^4 + x_1x_2^3 + x_2^4 - 3x_1^2x_2x_3 - 4x_1x_2^2x_3 + 2x_1^2x_3^2 + x_1x_3^3 + x_2x_3^3 + x_3^4 \qquad (1.4)$$

a une décomposition SOS avec des coefficients réels mais qu'il n'existe pas de décompositions SOS avec des coefficients rationnels [79].

Pour les sommes de carrés de fonctions rationnelles avec des coefficients rationnels, Jannsen a fait remarquer dans [36] que, avec $n \geq 2$, si $f \in \mathbb{Q}[\boldsymbol{x}]$ est non-négatif alors f est une somme de 2^{n+1} carrés de fonctions rationnelles sur $\mathbb{Q}$.

Somme des carrés et programmes semi-définis. Il est intéressant de noter que le problème de l'expression d'un polynôme sous forme de SOS peut être examiné du point de vue de l'optimisation convexe. Nous sommes en mesure de décider si un polynôme peut être décomposé en SOS par le biais de la programmation semi-définie.

Un programme semi-défini (SDP en abrégé) est le problème d'optimisation suivant [90]

$$\begin{aligned} \text{minimize} \quad & \operatorname{tr}(CX) \\ \text{subject to} \quad & \operatorname{tr}(A_iX) = b_i,\ i = 1, \ldots, m, \\ & X \succeq 0, \end{aligned}$$

où X, C, A_i sont des matrices symétriques réelles, b_i dans $\mathbb{R}$; X est la variable matricielle variable, C, A_i, et $b_i \in \mathbb{R}$ sont des données, et $\operatorname{tr}(\cdot)$ représente l'opérateur de trace matricielle habituel. Par conséquent, la fonction objectif et les contraintes sont convexes. La caractéristique cruciale du SDP est sa convexité. Ce problème convexe peut être résolu par des méthodes de point intérieur (voir, par ex, [16, Chapitre 11]), des méthodes du premier ordre (voir, par ex, [11]) ou les méthodes de l'ellipsoïde (voir, par exemple [31, Chapitre 3]).

On désigne par $v_d(\boldsymbol{x}) = (1, x_1, x_2, \ldots, x_n, x_1x_2, \ldots, x_n^d)^T$ vecteur contenant tous les monômes de degré au plus égal à d. La longueur du vecteur $v_d(\boldsymbol{x})$ est égale à $\binom{n+d}{d}$. Choi, Lam, et Reznick [19] ont établi le fait que $f \in \mathbb{R}[\boldsymbol{x}]$ est SOS si et seulement s'il existe une matrice semi-définie positive Q, c'est-à-dire que Q est une matrice symétrique ayant des valeurs propres non négatives, telle que que $f = v_d^T Q v_d$. Une telle matrice Q est appelée une matrice de *Gram* associée à f. Puisque le calcul de telles matrices Gram se

réduit à la résolution d'inégalités matricielles linéaires, le calcul d'une décomposition SOS de f se résume à la résolution d'un problème de faisabilité SDP.

En pratique, les résolutions SDP fournissent des approximations numériques ; par conséquent les décompositions SOS obtenues par cette méthode sont approximatives et donc non exactes, voir, par exemple, [86] ou [49, Chapitre 2].

Sommes de carrés et optimisation polynomiale. Au cours des deux dernières décennies, inspirée par les travaux de Lasserre [44] et de Parrilo [63], la théorie SOS ainsi que la programmation semi-définie ont été des méthodologies très populaires pour attaquer un problème d'optimisation polynomiale.

Pour décrire brièvement cette idée, nous considérons le problème non contraint comme suit :

$$f_{\inf} := \inf_{x \in \mathbb{R}^n} f(x). \tag{1.5}$$

Ce problème est NP-dur [61] lorsque le degré de f est supérieur ou égal à 4. Clairement, ce problème peut être reformulé comme suit :

$$f_{\inf} = \sup \rho \ \text{ s.t. } \ f - \rho \geq 0 \ \text{ on } \ \mathbb{R}^n. \tag{1.6}$$

En remplaçant la condition de non-négativité non négative dans (2.6) par une condition plus facile à gérer est une idée naturelle pour traiter le le problème (2.6). Par la suite, le remplacement de la condition de non-négativité par la condition SOS proposée par Shor [81] mène à une relaxation SOS de (2.5) comme suit :

$$f_{\mathtt{sos}} := \sup \rho \ \text{ s.t. } \ f - \rho \text{ is } \text{SOS}. \tag{1.7}$$

Comme l'ensemble des polynômes non négatifs contient celui des polynômes SOS, $f_{\mathtt{sos}}$ est une borne inférieure de $f_{\inf}$. La relation entre la programmation semi-définie et la théorie SOS nous permet de calculer $f_{\mathtt{sos}}$ via un programme semi-défini qui peut être résolu en temps polynomial jusqu'à une précision prescrite [69, 21].

Problem 1. Le premier sujet de cette thèse porte sur les *certificats exacts de non-négativité pour polynômes multivariés réels basés sur des décompositions SOS à coefficients rationnels*. Nous sommes intéressés par les algorithmes de certification, leur complexité binaire ainsi que les implémentations.

Comme décrit ci-dessus, notre motivation principale vient du besoin de certificats exacts pour l'optimisation polynomiale certifiée. Nous rappelons et soulignons ici les deux difficultés auxquelles nous sommes confrontés :

- L'ensemble de polynômes non-négatifs est significativement plus grand que l'ensemble des polynômes SOS.

- Il existe des polynômes à coefficients rationnels qui sont SOS à coefficients réels, mais pas à coefficients rationnels.

1.1.2 Certificats exacts pour les polynômes complexes univariés

On désigne par $\boldsymbol{i}$ l'unité imaginaire. Pour une variable ou un nombre complexe v, nous notons $\bar{v}$ son conjugué. Soit z une variable complexe. Pour un polynôme complexe univarié

$$h(z) = h_0 + h_1 z + \ldots + h_d z^d$$

dans $\mathbb{C}[z]$, où $h_k \in \mathbb{C}$, nous définissons

$$h^\star(z) := \bar{h}_0 + \bar{h}_1 \frac{1}{z} + \cdots + \bar{h}_d \frac{1}{z^d}.$$

Le *cercle unité* $\mathscr{C}$ est défini par

$$\mathscr{C} := \{\zeta \in \mathbb{C} : |\zeta| = 1\}.$$

On voit que $\bar{\zeta} = \zeta^{-1}$ et $\zeta\bar{\zeta} = 1$, pour $\zeta \in \mathscr{C}$. Par conséquent, sur le cercle unité, le produit $hh^\star$ est le carré du module de h.

Polynômes trigonométriques univariés. Soit $\mathscr{H}[z]$ l'ensemble des *polynômes trigonométriques univariés* défini comme un sous-ensemble des polynômes de Laurent à coefficients complexes et en la variable complexe z comme suit :

$$f(z) = f_0 + \left(\frac{f_1}{z} + \bar{f}_1 z\right) + \cdots + \left(\frac{f_d}{z^d} + \bar{f}_d z^d\right),$$

avec $f_0 \in \mathbb{R}$ et $d \in \mathbb{N}$. Par convention, lorsque $f_d \neq 0$, d est le degré de f ; le degré du du polynôme zéro est $-\infty$. Clairement, si $f \in \mathscr{H}[z]$ alors f a des valeurs réelles sur le cercle unité. De plus, si α est une racine de f alors sa réciproque $1/\bar{\alpha}$ est aussi une racine de f.

Puisque nous travaillons avec des corps de base de caractéristique zéro, nous voyons les polynômes à travers les cartes d'évaluation qu'ils définissent plutôt que comme des objets algébriques. Notez que pour $f \in \mathscr{H}[z]$, la restriction de la carte $\zeta \mapsto f(\zeta)$ sur le cercle unité $\mathscr{C}$ coïncide avec l'application d'évaluation définie par le polynôme

$$g(z) = f_0 + \left(f_1 \bar{z} + \bar{f}_1 z\right) + \cdots + \left(f_d \bar{z}^d + \bar{f}_d z^d\right),$$

puisque $\bar{\zeta} = \zeta^{-1}$ pour $\zeta \in \mathscr{C}$. Remarquons également que pour tout $\zeta \in \mathscr{C}$, $g(\zeta) = g(\bar{\zeta}) \in \mathbb{R}$, de sorte que g est un polynôme *hermitien*. Enfin, notons que pour tout polynôme hermitien g, il existe $f \in \mathscr{H}[z]$ tel que les restrictions à $\mathscr{C}$ des applications $\zeta \mapsto g(\zeta)$ et $\zeta \mapsto f(\zeta)$ coïncident.

Somme des carrés hermitiens. On dit que f est un *somme de carrés hermitiens*, SOHS en abrégé, s'il existe un certain $r \in \mathbb{N} \setminus \{0\}$ et des polynômes $s_1, \ldots, s_r$ dans $\mathbb{C}[z]$ tels que

$$f(z) = s_1(z)s_1^\star(z) + \cdots + s_r(z)s_r^\star(z). \tag{1.8}$$

Cette terminologie des carrés hermitiens provient de la discussion ci-dessus sous la forme $s_j^\star(\zeta) = s_j(\bar{\zeta})$ pour tout $\zeta \in \mathscr{C}$. Clairement, si f est SOHS comme dans (1.8) alors, à cause de $s_j(z)s_j^\star(z) = |s_j(z)|^2$ sur $\mathscr{C}$, f est non-négatif sur $\mathscr{C}$.

Selon le théorème de factorisation spectrale de Riesz-Fejér (voir, par exemple, [22, Theorem 1.1]), tout polynôme trigonométrique univarié f qui est non négatif sur le cercle unité $\mathscr{C}$ peut être écrit comme un carré hermitien. De plus, d'après sa preuve [22, pp. 3–5], on a

$$f = a \times \prod_{k=1}^{d}(z - a_k) \times \left(\frac{1}{z} - \bar{a}_k\right),$$

où $(a_1, 1/\bar{a}_1) \ldots, (\alpha_d, 1/\bar{a}_d)$ sont d paires de racines de f, et a est un scalaire positif. Ceci nous permet de de concevoir un algorithme pour calculer les certificats de non-négativité de f dans lequel nous devons manipuler *exactement* toutes les $2d$ racines complexes de f. Normalement, cet algorithme est appliqué avec des calculs approximatifs, ce qui conduit à des certificats approcimatifs de non-négativité sur $\mathscr{C}$. Nous cherchons à calculer des certificats de non-négativité *exacts* de non-négativité des polynômes trigonométriques. En particulier, lorsque les coefficients sont des entiers *Gaussiens*, c'est-à-dire que les parties réelles et imaginaires sont des entiers, les décompositions SOHS exactes de f peuvent être calculées par des méthodes numériques-symboliques hybrides.

Problem 2. Le deuxième sujet de cette thèse porte sur les *Certificats exacts de non-négativité pour les polynômes trigonométriques univariés basés sur les décompositions SOHS avec coefficients gaussiens*. Nous sommes également intéressés par les algorithmes de certification, leur complexité en bits et leurs implémentations.

Notre motivation provient de problèmes de conception dans le traitement du signal en temps discret. En particulier, pour la conception de filtres à réponse impulsionnelle finie (FIR) dans le traitement du signal, la minimisation de l'énergie de la bande d'arrêt est une question cruciale [22, Chapitre 5]. Calculer des décompositions SOHS exactes pour des polynômes trigonométriques univariés polynômes univariés dans ce contexte semble donc être un problème de calcul naturel.

1.2 Travaux connexes pour les décompositions SOS exactes

1.2.1 Polynômes univariés

Il est bien connu que tout polynôme univarié non négatif $f \in \mathbb{R}[x_1]$ avec des coefficients réels peut être décomposée comme une somme d'au plus deux carrés de polynômes. De même, tout polynôme non négatif univarié non négatif $f \in \mathbb{Q}[x_1]$ est un somme pondérée de carrés à coefficients rationnels [43, 70].

Dans la littérature, nous connaissons deux algorithmes qui calculent les décompositions exactes SOS exactes d'un polynôme non négatif $f \in \mathbb{Q}[x_1]$ à coefficients rationnels. Le premier [80] a été soulevé par Schweighofer en 1999 et s'appuie sur l'isolement des racines réelles, l'approximation quadratique des polynômes positifs et la décomposition sans carré. Le second a été proposée par Chevillard, Harrison, Joldes, et Lauter [18] en 2011 et est basée sur l'isolation des racines complexes et la décomposition sans carré. Leurs complexités binaires et leurs benchmarks sont donnés dans [53].

Récemment, Krick, Mourrain, et Szanto dans [41] ont proposé une condition nécessaire et suffisante pour la non-négativité d'un polynôme $f \in \mathbb{Q}[x_1]$ sur les racines réelles d'un autre polynôme $g \in \mathbb{Q}[x_1]$. En particulier, sous une condition légère, f est non-négatif sur toutes les racines réelles de g si et seulement si f est un SOS modulo g. Dans leur article, ils fournissent également un algorithme pour calculer une décomposition SOS.

1.2.2 Polynômes multivariés

Pour le cas multivarié, c'est-à-dire $n \geq 3$, Hilbert [34] a prouvé que tout polynôme homogène non négatif de degré d dans $\mathbb{R}[\boldsymbol{x}]$ est SOS si et seulement si $d = 2$ ou $(n, d) = (3, 4)$. Dans ce travail, il a prouvé que, pour $n = 3$, tout polynôme non négatif de degré 4 est une somme de trois carrés.

À la suite de ces travaux fondateurs [44, 63], les hiérarchies de programmes semi-définis donnent des *approximations* des décompositions SOS pondérées de polynômes positifs. Plusieurs heuristiques ont été proposées pour élever ces approximations à des décompositions *exactes* SOS du polynôme d'entrée. En commençant par la méthode rouding-projection soulevée par Peyrl et Parrilo [67], cette méthode peut être appliquée pour les polynômes situés à l'intérieur du cône des polynômes SOS, et suivie par des méthodes hybrides numériques-symboliques [38, 39, 40]. Il convient de noter que les algorithmes de [38, 40] nous permettent de calculer des décompositions SOS sur certains exemples dégénérés ou de calculer des SOS de fractions rationnelles. Les problèmes de complexité sont étudiés sous l'angle des techniques de perturbation-

compensation afin de calculer des décompositions SOS à l'intérieur du cône SOS [50, 51, 52]. Des algorithmes généraux pour calculer de tels certificats exacts au moyen de décompositions SOS ont été conçu, soit pour le calcul des décompositions SOS à coefficients rationnels [78] ou avec des nombres algébriques en calculant des solutions exactes à des programmes semi-définis exactes de programmes semi-définis [32].

Des certificats exacts alternatifs de non-négativité, par exemple, des sommes de nombres non négatifs et les sommes d'exponentielles arithmétiques-géométriques [56, 87] peuvent également être utilisés. Cependant, ils sont confrontés à des problèmes similaires à ceux rencontrés par les techniques SOS en matière de généralité.

Décider de la non-négativité d'un polynôme $f \in \mathbb{Q}[\boldsymbol{x}]$ sur un ensemble semi-algébrique arbitraire peut être fait exactement en utilisant des algorithmes de calcul formel. Les meilleures complexités pour une telle procédure de décision sont obtenues par des algorithmes rendant efficace la méthode dite du point critique [30, 9]. D'autres développements pratiques sont présentés dans [5, 6, 7, 76] et leurs applications en optimisation polynomiale sont données dans [28, 29, 8]. Notez que, même si ces algorithmes sont exacts (c'est-à-dire que leurs résultats sont exacts à condition qu'aucun bug n'ait été rencontré), ils ne fournissent pas de certificat de non-négativité qui puisse être vérifié a posteriori puisqu'il s'agit d'algorithmes de recherche de racines. Leurs complexités sont exponentielles dans la dimension de l'espace ambiant, car ils réduisent le problème d'entrée au calcul d'un nombre fini de points critiques de certaines applications bien choisies. Par conséquent, l'idée de considérer les *idéaux gradients* est naturelle.

En résumé, de tels idéaux de gradient peuvent être utilisés pour réduire la dimension de l'ensemble sur lequel la certification de non-négativité peut être effectuée. l'ensemble sur lequel la certification de la non-négativité peut être faite. Sous certaines hypothèses, cette idée est traduite dans [66] dans un algorithme évaluant la non-négativité d'une rupture de ligne donnée $f \in \mathbb{R}[\boldsymbol{x}]$. Précisément, en supposant l'idéal de gradient idéal $\mathcal{I}_{\text{grad}}(f)$ (qui est l'ensemble de toutes les combinaisons combinaisons algébriques des dérivées partielles de f) est de dimension nulle et radicale, et que f atteint son infimum sur $\mathbb{R}^n$, cet algorithme calcule une décomposition SOS de f dans l'anneau quotient quotient $\mathbb{R}[\boldsymbol{x}]/\mathcal{I}_{\text{grad}}(f)$ (ou, en d'autres termes, une décomposition SOS de f modulo $\mathcal{I}_{\text{grad}}(f)$), c'est-à-dire que f s'écrit comme suit

$$c_1 s_1^2 + \cdots + c_k s_k^2 + \sum_{i=1}^{n} q_i \frac{\partial f}{\partial x_i} ,$$

où les s_i et les q_i se situent dans $\mathbb{R}[\boldsymbol{x}]$ et les c_i sont positifs dans $\mathbb{R}$. Un résultat similaire en relâchant légèrement les hypothèses ci-dessus est donné dans [62]. Notez que lorsque f a des coefficents dans $\mathbb{Q}$, il n'y a pas de garantie donnée qu'une décomposition SOS décomposée de celui-ci dans $\mathbb{Q}[\boldsymbol{x}]/\mathcal{I}_{\text{grad}}(f)$ aura également des coefficients rationnels.

En appliquant le résultat de Parrilo dans [66], on peut conclure qu'un polynôme $f \in \mathbb{R}[\boldsymbol{x}]$ est non négatif sur la variété réelle d'un idéal $\mathcal{I}$ si et seulement si f est SOS sur l'anneau quotient $\mathbb{R}[\boldsymbol{x}]/\mathcal{I}$. Nie, Demmel et Sturmfels dans [62] ont montré qu'un polynôme $f \in \mathbb{R}[\boldsymbol{x}]$ non négatif sur sa variété réelle gradiente est SOS modulo l'idéal gradient de f à condition que cet idéal soit radical ou que f soit strictement positif sur la variété réelle de gradient [62].

1.3 Contributions

1.3.1 Décompositions SOS exactes de polynômes réels multivariés

Nous considérons le problème du calcul d'une décomposition SOS exacte d'un polynôme multivarié réel à coefficients rationnels, $f \in \mathbb{Q}[\boldsymbol{x}]$. Nous résumons nos contributions comme suit :

Existence de certificats de non-négativité avec des coefficients rationnels. Nous fournissons une condition nécessaire et suffisante pour la non-négativité de $f \in \mathbb{Q}[\boldsymbol{x}]$ sous une condition de généricité.

- Supposons que l'idéal de gradient associé à f est radical et de dimension nulle et que f atteint son infimum sur $\mathbb{R}^n$. Nous prouvons dans le théorème 5.1.1 que *f est non-négatif sur $\mathbb{R}^n$ si et seulement si f est un SOS de polynômes à coefficients rationnels sur l'anneau quotient $\mathbb{Q}[\boldsymbol{x}]/\mathcal{I}_{\text{grad}}(f)$*. Nous nous appuyons sur une procédure d'élimination algébrique basée sur le Lemme de Forme (voir Lemma 3.3.1) pour réduire le problème au cas univarié. Il est intéressant de noter que le théorème 5.1.1 peut être appliqué au polynôme de Robinson (voir Exemple 5.1.1), ainsi qu'au polynôme de Scheiderer (voir Exemple 5.1.2) qui n'ont pas de décomposition SOS à coefficients rationnels.

Algorithmes et complexités des bits. Le problème suivant auquel nous nous attaquons est de concevoir des algorithmes qui calculent de tels certificats de non-négativité ainsi que d'estimer leur complexité binaire. Pour mesurer la *taille du bit* d'un polynôme à coefficients rationnels, nous utiliserons son *hauteur*, défini comme dans la section 4.1.

- De la preuve du théorème 5.1.1, nous dérivons l'algorithme `sosgradientshape` (Algorithme 3) pour calculer une décomposition SOS de polynômes modulo l'idéal gradient de f. Nous prouvons que, étant donné en entrée un polynôme de n-variétés $f \in \mathbb{Q}[\boldsymbol{x}]$ de degré d avec un maximum de maximale de ses coefficients τ, cet algorithme utilise $\widetilde{O}((\tau+n+d)^2 d^{6n} + (\tau+n+d)d^{6n+4})$ boolean opérations

booléennes. Cette estimation est meilleure que l'estimation de la complexité donnée dans [52, Theorem 12], où le nombre rapporté est $\widetilde{O}(\tau^2(4d+2)^{15n+6})$.

- Nous concevons une variante de l'algorithme précédent, nommée `sosgradient`. (Algorithme 4). Sur entrée $f \in \mathbb{Q}[\boldsymbol{x}]$ comme ci-dessus, cet algorithme décompose f en une somme de *fractions rationnelles fractions rationnelles* modulo l'idéal gradient associé à f. Nous prouvons que cette variante utilise $\widetilde{O}\left((\tau+n+d)d^{4n+4}\right)$ opérations booléennes et présente donc une meilleure complexité que Algorithme `sosgradientshape`.

Nous soulignons que que ces estimations de complexité sont intéressantes pour la communauté de l'optimisation polynomiale, car elles donnent des limites de degré pour les multiplicateurs SOS requis lorsque l'utilisation de la variante de ce que l'on appelle la "hiérarchie Moment-SOS" ou la hiérarchie de Lasserre [44] pour minimiser les polynômes sur leurs idéaux gradients [62]. En effet, de telles limites de degré se traduisent par des taux de convergence pour le schéma d'optimisation sous-jacent et nous permettent d'estimer la complexité globale du coût de calcul.

Résultats expérimentaux. Nos deux algorithmes ont été mis en œuvre à l'aide du système algébrique informatique MAPLE. Nous présentons des expériences pratiques qui montrent que nos algorithmes peuvent déjà évaluer la non-négativité de nombreux polynômes qui sont hors de portée, par exemple, des méthodes hybrides calculant des sommes de carrés des décompositions telles que [50].

- Nos expériences pratiques montrent que *Algorithme* `sosgradient` *peut évaluer la non-négativité des polynômes multivariés d'un large ensemble d'exemples qui sont hors de portée de l'état de l'art* lorsque le nombre de variables et le degré augmentent.

1.3.2 Décompositions SOHS exactes de polynômes complexes univariés

Nous concevons trois algorithmes pour calculer les décompositions SOHS exactes de polynômes dans $\mathscr{H}(\mathbb{Z})[z]$ qui sont *positifs* sur le cercle unité $\mathscr{C}$. Ces algorithmes sont basés sur des techniques de compensation de perturbation ou de projection d'arrondi. De plus, nous analysons leur complexité binaire et la taille de leur sortie.

Le premier algorithme est appelé `csos1` (Algorithme 5). L'algorithme `csos1` utilise une méthode de perturbation-compensation dans laquelle l'étape numérique calcule une décomposition approximative de SOHS approximative pour une perturbation bien choisie du polynôme d'entrée avec isolation des racines complexes isolement. Nous obtenons la complexité binaire de l'algorithme comme suit :

- *Sur l'entrée f, où f n'a pas de racines multiples,* `csos1` *calcule une décomposition SOHS de f avec des coefficients (modulo de) gaussiens en utilisant au maximum $\left(d^6(d+\tau)\right)$ opérations sur les bits. De plus, la taille maximale des bits des coefficients de sortie est bornée dans $\widetilde{O}(d^5(d+\tau))$.*

Les deux autres algorithmes sont appelés `csos2` et `csos3`. (Algorithmes 6 et 7, respectivement). Nous concevons deux algorithmes `csos2` et `csos3` qui sont basés sur la programmation semi-définie complexe. Dans l'algorithme `csos2`, nous calculons une décomposition SOHS approximative pour la perturbation à l'aide de la résolution SDP complexe. L'algorithme `csos3` est une adaptation de l'algorithme d'arrondissement-projection d'arrondi-projection soulevé par Peyrl et Parrilo [67]. Les complexités binaires de ces deux algorithmes sont similaires car nous utilisons la résolution SDP complexe pour les deux :

- *Sur l'entrée f,* `csos2` *et* `csos3` *calculent les décompositions SOHS de f avec (modulus de) coefficients gaussiens en utilisant au maximum $\widetilde{O}(d^{13}(d+\tau)^2)$ opérations sur bits. De plus, la taille binaire maximale des coefficients de sortie est limitée par dessus par $\widetilde{O}(d^6(\tau+d))$.*

Ces deux algorithmes sont plus coûteux que l'algorithme `csos1` parce que nous remplaçons l'isolation de la racine complexe par la résolution complexe de SDP. Malgré leur plus grande complexité, ils permettent de traiter des problèmes d'optimisation contraints et de concevoir des filtres.

Résultats expérimentaux. Ces algorithmes ont été implémentés en utilisant le langage de programmation JULIA [13].

- Nous présentons des expériences pratiques montrant que l'algorithme `csos1` fonctionne plus rapidement que les autres algorithmes, ce qui coïncide avec nos résultats théoriques de complexité. De plus, nous nous appuyons sur `csos3` pour concevoir des filtres de manière certifiée.

1.4 Organisation de la thèse

Cette thèse comprend trois parties. La partie I est la partie préliminaire qui contient les chapitres 3 et 4. Dans ces chapitres, nous rappelons des notions de base et des résultats fondamentaux de géométrie algébrique et d'algèbre commutative effective, ainsi que des résultats de complexité binaire pour la résolution de systèmes polynomiaux.

Nos principales contributions apparaissent dans la partie II qui comprend les chapitres:

- Le chapitre 5 est consacré aux certificats exacts de de non-négativité pour les polynômes multivariés réels. Le contenu de ce chapitre est tiré de l'article intitulé "*Sum of squares decompositions of polynomials over their gradient ideals with rational coefficients*" par Victor Magron, Mohab Safey El Din, et Trung Hieu Vu [55] qui a été accepté pour publication dans le SIAM Journal of Optimization, 2022.

- Le chapitre 6 présente des résultats sur les certificats exacts de positivité pour les polynômes complexes univariés. Ces résultats ont été publiés dans l'article intitulé "*Exact SOHS decompositions of trigonometric univariate polynomials with Gaussian coefficients*" [54] par Victor Magron, Mohab Safey El Din, Markus Schweighofer et Trung Hieu Vu dans les "Proceedings of the 2022 International Symposium on Symbolic and Algebraic Computation", Villeneuve-d'Ascq, France, 2022.

Enfin, dans la partie III qui ne contient que le chapitre 7, nous donnons quelques conclusions et décrivons les futures directions de recherche.

CHAPTER 2

Introduction

In polynomial optimization, it is well-known that computing the infimum of a polynomial on a semi-algebraic set is NP-hard. To bypass this computational issue, one can use hierarchies of semi-definite relaxations and the theory of sums of squares (SOS for short) of polynomials, in which every semi-definite program can be solved, up to arbitrary precision, in polynomial time. Within this framework, non-negativity of polynomials is replaced by the property of being SOS. Hence, assessing non-negativity of polynomials based on SOS decompositions is a topical issue in polynomial optimization.

Certificates of non-negativity are usually tackled through the computation of SOS decompositions which rely on efficient numerical solvers for semi-definite programming. Consequently, certificates obtained in this way are *approximate* and therefore non-exact. For some critical applications, it is important to actually compute *exact* certificates of non-negativity.

The aim of this thesis is to compute exact certificates of non-negativity for polynomials based on SOS decompositions with rational coefficients. Certifications using SOS face the following two difficulties. Firstly, due to the fact that the set of non-negative polynomials is significantly larger than that of SOS polynomials, certifications based on SOS decompositions cannot be applied to all non-negative polynomials. Secondly, there exist polynomials with rational coefficients that are SOS with real coefficients but are not SOS with rational coefficients.

In this thesis, we provide symbolic algorithms to compute SOS decompositions modulo the gradient ideal of non-negative real multivariate polynomials under a genericity condition. These algorithms can tackle a large range of problems which are out of reach for state-of-the-art algorithms. We also compute sums of Hermitian squares decompositions for complex trigonometric univariate polynomials that are positive on the unit circle with Gaussian coefficients. Moreover, we analyze the bit complexity of these algorithms and deduce bitsize bounds of such certificates. Finally, we implement these algorithms in the computer algebra system MAPLE and the programming environment JULIA and evaluate their performance on some standard benchmarks.

Contents

2.1 Research topics and motivations

We denote by $\mathbb{N}$ the set of natural numbers, by $\mathbb{Z}$ the set of integers, and by $\mathbb{Q}$, $\mathbb{Q}_+$, $\mathbb{R}$ and $\mathbb{C}$ the fields of rational, non-negative rational, real and complex numbers, respectively. Let $\boldsymbol{x}$ be the n-tuple of variables $(x_1, \ldots, x_n)$. Let $\mathbb{K}$ be a field, we denote by $\mathbb{K}[\boldsymbol{x}]$ the polynomial ring with base field $\mathbb{K}$ and variables $\boldsymbol{x}$.

2.1.1 Exact certificates for real multivariate polynomials

A polynomial f in $\mathbb{R}[\boldsymbol{x}]$ of degree d is *non-negative* over $\mathbb{R}^n$ if it takes only non-negative values. Providing checkable conditions or a procedure for verifying non-negativity of polynomials is a crucial issue in polynomial optimization theory [64, 65]. There are several topical questions concerning non-negativity of polynomials such as [65]: How do we *decide* non-negativity of f? Is it possible to *certify* the non-negativity of f? What is the *complexity* of deciding non-negativity of f? What is the *structure* of the set of non-negative polynomials? Our research in this thesis focuses on the certification and complexity issues.

Non-negativity and sums-of-squares. Clearly, the non-negativity of f follows from the fact that f can be decomposed as a *sum of squares* (SOS) of polynomials, namely

$$f = q_1^2 + \cdots + q_r^2, \tag{2.1}$$

where $q_1, \ldots, q_r$ are in $\mathbb{R}[\boldsymbol{x}]$, for some $r \in \mathbb{N} \setminus \{0\}$. The right-hand side of (2.1) is called an *SOS decomposition* of f and provides a *certificate* of non-negativity for f.

The relationship between the properties of non-negativity and SOS of polynomials has been investigated since the late 19th century, raising Hilbert's 17th problem. Hilbert's theorem, in [34], says that non-negativity and SOS property of homogeneous polynomials are equivalent if and only if $n = 2$, or $d = 2$, or $(n, d) = (3, 4)$. It follows from Hilbert's theorem that, for other cases, the set of non-negative polynomials strictly contains the set of SOS polynomials. In 1967, Motzkin gave the first explicit polynomial p_M that was non-negative but not SOS [60],

$$p_M := x_1^2 x_2^2 (x_1^2 + x_2^2 - 3) + 1. \tag{2.2}$$

Quantitatively, Blekherman [15] proved that for any fixed degree $d \geq 4$, if the number of variables is large enough then the set of non-negative polynomials is significantly larger than that of SOS polynomials. Note that, the set of non-negative polynomials and that of SOS polynomials are convex cones and both of them are full-dimensional in $\mathbb{R}_d[\boldsymbol{x}]$ the space of all polynomials of degree at most d [72].

Recall that non-negativity of f can also be certified if f can be decomposed as a sum of squares of *rational functions*, i.e., every q_ℓ in (2.1) is a fraction of two nontrivial polynomials in $\mathbb{R}[\boldsymbol{x}]$,

$$f = \left(\frac{u_1}{v_1}\right)^2 + \cdots + \left(\frac{u_r}{v_r}\right)^2, \tag{2.3}$$

Hilbert's 17-th problem in 1900 [35] was the following: For any $f \in \mathbb{R}[\boldsymbol{x}]$, *is it true that if f is non-negative over $\mathbb{R}^n$ then f is a sum of squares of rational functions?* Artin gave an affirmative answer [2] in 1927. For example, the non-negativity of Motzkin's polynomial p_M can be certified by writing it as an SOS of rational functions as follows:

$$p_M = \frac{1}{x_1^2 + x_2^2}(x_1^2(1 - x_2)^2 + x_2^2(1 - x_1)^2 + x_1^2 x_2^2 (x_1^2 + x_2^2 - 2)^2).$$

We are also interested in the question about the maximal number of squares r and the maximal degree of u_ℓ, v_ℓ in the decomposition (2.3). In 1967, Pfister [68] proved that if $f \in \mathbb{R}[\boldsymbol{x}]$ is non-negative then f is a sum of 2^n squares of rational functions, that number depending only on the number of variables. Lombardi, Perrucci and Roy proved in 2014 that if f is non-negative then f can be written as a sum of squares of rational functions of degree at most $2^{2^{d^{4^n}}}$ in [48].

The relationship between non-negativity and the SOS property of polynomials can be found in more detail in the monographs [46, 58, 71], or in the survey article [65].

Sums of squares with rational coefficients. Let $f \in \mathbb{Q}[\boldsymbol{x}]$ be a polynomial with rational coefficients. One says that f is (weighted) sum of squares with rational coefficients

if it can be written

$$f = c_1 s_1^2 + \cdots + c_r q_r^2,$$

where $c_1, \ldots, c_r$ are in $\mathbb{Q}_+$, and $q_1, \ldots, q_r$ are in $\mathbb{Q}[\boldsymbol{x}]$, for some positive integer r.

Any non-negative univariate polynomial $f \in \mathbb{Q}[x_1]$ with rational coefficients can be decomposed as a weighted sum of squares with rational coefficients [43, 70]. Landau [43] proved in 1905 that the maximal number of squares is at most eight, i.e., $r \leq 8$. This result was improved by Pourchet [70] in 1971 by $r \leq 5$.

More than ten years ago, Sturmfels raised the question whether a polynomial with rational coefficients is necessarily an SOS of polynomials with rational coefficients. Scheiderer gave a negative answer in [79], where he constructed families of explicit homogeneous polynomials with rational coefficients that are SOS with real coefficients but not with rational coefficients. He pointed out explicitly that the polynomial

$$p_S = x_1^4 + x_1 x_2^3 + x_2^4 - 3x_1^2 x_2 x_3 - 4x_1 x_2^2 x_3 + 2x_1^2 x_3^2 + x_1 x_3^3 + x_2 x_3^3 + x_3^4 \tag{2.4}$$

has an SOS decomposition with real coefficients but there are no SOS decompositions with rational coefficients [79].

For sums of squares of rational functions with rational coefficients, Jannsen pointed out in [36] that, with $n \geq 2$, if $f \in \mathbb{Q}[\boldsymbol{x}]$ is non-negative then f is a sum of 2^{n+1} squares of rational functions over $\mathbb{Q}$.

Sums of squares and semi-definite programs. Interestingly, the problem of expressing a polynomial as an SOS can be examined from the viewpoint of convex optimization. We are able to decide whether a polynomial can be decomposed as an SOS decomposition through semi-definite programming.

A semi-definite program (SDP for short) is the following optimization problem [90]

$$\begin{aligned} \text{minimize} \quad & \operatorname{tr}(CX) \\ \text{subject to} \quad & \operatorname{tr}(A_i X) = b_i,\ i = 1, \ldots, m, \\ & X \succeq 0, \end{aligned}$$

where X, C, A_i are real symmetric matrices, b_i in $\mathbb{R}$; X is the matrix variable, C, A_i, and $b_i \in \mathbb{R}$ are given data, and $\operatorname{tr}(\cdot)$ stands for the usual matrix trace operator. Consequently, the objective function and the constraints are convex. The crucial feature of SDP is its convexity. This convex problem can be solved by interior-point methods (see, e.g., [16, Chapter 11]), first-order methods (see, e.g., [11]) or ellipsoid methods (see, e.g., [31, Chapter 3]).

Denote by $v_d(\boldsymbol{x}) = (1, x_1, x_2, \ldots, x_n, x_1x_2, \ldots, x_n^d)^T$ the vector containing all monomials of degree at most d. The length of the vector $v_d(\boldsymbol{x})$ is equal to $\binom{n+d}{d}$. Choi, Lam, and Reznick [19] established the fact that $f \in \mathbb{R}[\boldsymbol{x}]$ is SOS if and only if there exists a positive semi-definite matrix Q, that is Q is a symmetric matrix having non-negative eigenvalues, such that $f = v_d^T Q v_d$. Such a matrix Q is called a *Gram matrix* associated to f. Since computing such Gram matricies reduces to solving linear matrix inequalities, computing an SOS decomposition of f boils down to solving an SDP feasibility problem.

In practice, SDP solvers provide numerical approximations; therefore SOS decompositions obtained by this method are approximate and hence not exact, see e.g., [86] or [49, Chapter 2].

Sums of squares and polynomial optimization. In the last two decades, inspired by the work by Lasserre [44] and Parrilo [63], SOS theory along with semi-definite programming have been very popular methodologies with which to attack a polynomial optimization problem.

To briefly describe this idea, we consider the unconstrained problem as follows:

$$f_{\inf} := \inf_{x \in \mathbb{R}^n} f(x). \tag{2.5}$$

This problem is NP-hard [61] when the degree of f is greater than or equal 4. Clearly, this problem can be reformulated as follows:

$$f_{\inf} = \sup \rho \ \text{ s.t. } \ f - \rho \geq 0 \ \text{ on } \ \mathbb{R}^n. \tag{2.6}$$

Replacing the non-negativity condition in (2.6) by a more tractable one is a natural idea to handle the problem (2.6). Hereafter, replacing the non-negativity condition by the SOS condition proposed by Shor [81], we obtain an SOS relaxation of (2.5) as follows:

$$f_{\mathtt{sos}} := \sup \rho \ \text{ s.t. } \ f - \rho \ \text{ is SOS.} \tag{2.7}$$

As the set of non-negative polynomials contains that of SOS polynomials, $f_{\mathtt{sos}}$ is a lower bound of $f_{\inf}$. The relationship between semi-definite programming and SOS theory allows us to compute $f_{\mathtt{sos}}$ via a semi-definite program which can be solved in polynomial time up to prescribed accuracy [69, 21].

Problem 1. The first topic in this thesis is about *exact certificates of non-negativity for real multivariate polynomials based on SOS decompositions with rational coefficients*. We are interested in certification algorithms, their bit complexity as well as implementations.

As described above, our main motivation comes from the need for exact certificates for certified polynomial optimization. We recall and emphasize here the two difficulties we are facing:

- The set of non-negative polynomials is significantly larger than the set of SOS polynomials.
- There exist polynomials with rational coefficients that are SOS with real coefficients, but not with rational coefficients.

2.1.2 Exact certificates for complex univariate polynomials

Denote by $\boldsymbol{i}$ the imaginary unit. For a complex variable or number v, we denote by $\bar{v}$ its conjugate. Let z be a complex variable. For a complex univariate polynomial

$$h(z) = h_0 + h_1 z + \ldots + h_d z^d$$

in $\mathbb{C}[z]$, where $h_k \in \mathbb{C}$, we define

$$h^\star(z) := \bar{h}_0 + \bar{h}_1 \frac{1}{z} + \cdots + \bar{h}_d \frac{1}{z^d}.$$

The *unit circle* $\mathscr{C}$ is defined by

$$\mathscr{C} := \{\zeta \in \mathbb{C} : |\zeta| = 1\}.$$

One can see that $\bar{\zeta} = \zeta^{-1}$ and $\zeta\bar{\zeta} = 1$, for $\zeta \in \mathscr{C}$. Hence, on the unit circle, the product $hh^\star$ is the square of the modulus of h.

Trigonometric univariate polynomials. Let $\mathscr{H}[z]$ be the set of *trigonometric univariate polynomials* defined as a subset of Laurent polynomials with complex coefficients and complex variable z as follows:

$$f(z) = f_0 + \left(\frac{f_1}{z} + \bar{f}_1 z\right) + \cdots + \left(\frac{f_d}{z^d} + \bar{f}_d z^d\right),$$

with $f_0 \in \mathbb{R}$ and $d \in \mathbb{N}$. By convention, when $f_d \neq 0$, d is the degree of f; the degree of the zero polynomial is $-\infty$. Clearly, if $f \in \mathscr{H}[z]$ then f has real values on the unit circle. Furthermore, if α is a root of f then its reciprocal $1/\bar{\alpha}$ is also a root of f.

Since we work with base fields of characteristic zero, we see polynomials through the evaluation maps they define rather than as algebraic objects. Note that for $f \in \mathscr{H}[z]$, the restriction of the map $\zeta \mapsto f(\zeta)$ over the unit circle $\mathscr{C}$ coincides with the evaluation map defined by the polynomial

$$g(z) = f_0 + (f_1 \bar{z} + \bar{f}_1 z) + \cdots + \left(f_d \bar{z}^d + \bar{f}_d z^d\right),$$

since $\bar{\zeta} = \zeta^{-1}$ for $\zeta \in \mathscr{C}$. Note also that for any $\zeta \in \mathscr{C}$, $g(\zeta) = g(\bar{\zeta}) \in \mathbb{R}$, so that g is a *Hermitian* polynomial. Finally, note that for any Hermitian polynomial g, there exists $f \in \mathscr{H}[z]$ such that the restrictions to $\mathscr{C}$ of the maps $\zeta \mapsto g(\zeta)$ and $\zeta \mapsto f(\zeta)$ coincide.

Sum of Hermitian squares. One says that f is a *sum of Hermitian squares*, SOHS for short, if there exists some $r \in \mathbb{N} \setminus \{0\}$ and polynomials $s_1, \ldots, s_r$ in $\mathbb{C}[z]$ such that

$$f(z) = s_1(z)s_1^\star(z) + \cdots + s_r(z)s_r^\star(z). \tag{2.8}$$

This terminology of Hermitian squares comes from the above discussion as $s_j^\star(\zeta) = s_j(\bar{\zeta})$ for all $\zeta \in \mathscr{C}$. Clearly, if f is SOHS as in (2.8) then, because of $s_j(z)s_j^\star(z) = |s_j(z)|^2$ over $\mathscr{C}$, f is non-negative over $\mathscr{C}$.

According to the Riesz-Fejér spectral factorization theorem (see, e.g., [22, Theorem 1.1]), any trigonometric univariate polynomial f which is non-negative over the unit circle $\mathscr{C}$ can be written as a Hermitian square. Moreover, from its proof [22, pp. 3–5], one has

$$f = a \times \prod_{k=1}^{d} (z - a_k) \times \Big(\frac{1}{z} - \bar{a}_k\Big),$$

where $(a_1, 1/\bar{a}_1) \ldots, (\alpha_d, 1/\bar{a}_d)$ are d pairs of roots of f, and a is a positive scalar. This allows us to design an algorithm to compute certificates of non-negativity for f in which we are required to manipulate *exactly* all $2d$ complex roots of f. Normally, this algorithm is applied with approximate computations, leading to approximate certificates of non-negativity over $\mathscr{C}$. We aim to compute *exact* certificates of non-negativity of trigonometric polynomials. In particular, when the coefficients are *Gaussian integers*, i.e., with real and imaginary parts being integers, exact SOHS decompositions of f can be computed by hybrid numeric-symbolic methods.

Problem 2. The second topic in this thesis is about *exact certificates of non-negativity for trigonometric univariate polynomials based on SOHS decompositions with Gaussian coefficients*. We are also interested in certification algorithms, their bit complexity, and implementations.

Our motivation comes from design problems in discrete-time signal processing. In particular, for the design of finite impulse response (FIR) filters in signal processing, minimizing the stop-band energy is a crucial issue [22, Chapter 5]. Computing exact SOHS decompositions of trigonometric univariate polynomials in this context appears to be a natural computational issue.

2.2 Related works for exact SOS decompositions

2.2.1 Univariate polynomials

It is well-known that every non-negative univariate polynomial $f \in \mathbb{R}[x_1]$ with real coefficients can be decomposed as a sum of at most two squares of polynomials. Also,

any non-negative univariate polynomial $f \in \mathbb{Q}[x_1]$ is a weighted sum of squares with rational coefficients [43, 70].

In the literature, we know of two algorithms that compute exact SOS decompositions of a non-negative polynomial $f \in \mathbb{Q}[x_1]$ with rational coefficients. The first one [80] was raised by Schweighofer in 1999 and relies on real root isolation, quadratic approximation of positive polynomials, and square-free decomposition. The second one was proposed by Chevillard, Harrison, Joldes, and Lauter [18] in 2011 and is based on complex root isolation and square-free decomposition. Their bit complexities and benchmarks are given in [53].

Recently, Krick, Mourrain, and Szanto in [41] have proposed a necessary and sufficient condition for the non-negativity of a polynomial $f \in \mathbb{Q}[x_1]$ over the real roots of another polynomial $g \in \mathbb{Q}[x_1]$. In particular, under a mild condition, f is non-negative on all the real roots of g if and only if f is an SOS modulo g. In their paper, they also provide an algorithm to compute an SOS decomposition.

2.2.2 Multivariate polynomials

For the multivariate case, i.e., $n \geq 3$, Hilbert [34] proved that every non-negative homogeneous polynomial of degree d in $\mathbb{R}[\boldsymbol{x}]$ is SOS if and only if $d = 2$ or $n = 3$ and $d = 4$. In this work, he proved that, for $n = 3$, every non-negative polynomial of degree 4 is a sum of three squares.

Following the seminal works [44, 63], hierarchies of semi-definite programs yield *approximations* of weighted SOS decompositions of positive polynomials. Several heuristics have been proposed to lift such approximations to *exact* SOS decompositions of the input polynomial. Starting with the rouding-projection method raised by Peyrl and Parrilo [67], this method can be applied for polynomials lying on the interior of the cone of SOS polynomials, and followed by hybrid numerical-symbolic methods [38, 39, 40]. Note that the algorithms in [38, 40] allow us to compute SOS decompositions on some degenerate examples or compute SOS of rational fractions. Complexity issues are studied through the lens of perturbation-compensation techniques to compute SOS decompositions in the interior of the SOS cone [50, 51, 52]. General algorithms for computing such exact certificates by means of SOS decompositions have been designed, either for computing SOS decompositions with rational coefficients [78] or with algebraic numbers by computing exact solutions to semi-definite programs [32].

Alternative exact certificates of non-negativity, for instance, sums of non-negative circuits and sums of arithmetic-geometric-exponentials [56, 87] can also be used. However, they face similar issues to the ones met by SOS techniques when it comes

with generality.

Deciding non-negativity if a polynomial $f \in \mathbb{Q}[\boldsymbol{x}]$ over an arbitrary semi-algebraic set can be done exactly using computer algebra algorithms. The best complexities for such a decision procedure are achieved by algorithms making effective the so-called critical point method [30, 9]. Further practical developments are given in [5, 6, 7, 76] and their applications in polynomial optimization are given in [28, 29, 8]. Note that, even if these algorithms are exact (i.e., their results are exact provided that no bug has been encountered), they do not provide a certificate assessing non-negativity which can be checked a posteriori since these are root-finding algorithms. Their complexities are exponential in the dimension of the ambient space as they reduce the input problem to computing finitely many critical points of some well-chosen maps. Therefore, the idea of considering *gradient ideals* is natural.

In summary, such gradient ideals can be used to reduce the dimension of the set over which certifying non-negativity can be done. Under some assumptions, this idea is translated in [66] to an algorithm assessing the non-negativity of a given $f \in \mathbb{R}[\boldsymbol{x}]$. Precisely, assuming the gradient ideal $\mathcal{I}_{\text{grad}}(f)$ (which is the set of all algebraic combinations of the partial derivatives of f) is zero-dimensional and radical, and that f reaches its infimum over $\mathbb{R}^n$, this algorithm computes an SOS decomposition of f in the quotient ring $\mathbb{R}[\boldsymbol{x}]/\mathcal{I}_{\text{grad}}(f)$ (or, in other words, an SOS decomposition of f modulo $\mathcal{I}_{\text{grad}}(f)$), i.e., f is written as

$$c_1 s_1^2 + \cdots + c_k s_k^2 + \sum_{i=1}^{n} q_i \frac{\partial f}{\partial x_i} ,$$

where the s_i's and the q_i's lie in $\mathbb{R}[\boldsymbol{x}]$ and the c_i's are positive in $\mathbb{R}$. A similar result slightly relaxing the above assumptions is given in [62]. Note that when f has coefficients in $\mathbb{Q}$, there is no given guarantee that an SOS decomposition of it in $\mathbb{Q}[\boldsymbol{x}]/\mathcal{I}_{\text{grad}}(f)$ will have rational coefficients too.

Applying Parrilo's result in [66], one can conclude that a polynomial $f \in \mathbb{R}[\boldsymbol{x}]$ is non-negative of over the real variety of an ideal $\mathcal{I}$ if and only if f is SOS over the quotient ring $\mathbb{R}[\boldsymbol{x}]/\mathcal{I}$. Nie, Demmel, and Sturmfels in [62] showed that a polynomial $f \in \mathbb{R}[\boldsymbol{x}]$ which is non-negative over its real gradient variety is SOS modulo the gradient ideal of f provided the gradient ideal is radical or f is strictly positive over the real gradient variety [62].

2.3 Contributions

2.3.1 Exact SOS decompositions of real multivariate polynomials

We consider the problem of computing an exact SOS decomposition of a real multivariate polynomial with rational coefficients, $f \in \mathbb{Q}[\boldsymbol{x}]$. We summarize our contributions as follows.

Existence of certificates of non-negativity with rational coefficients. We provide a necessary and sufficient condition for the non-negativity of $f \in \mathbb{Q}[\boldsymbol{x}]$ under a genericity condition.

- Assume that the gradient ideal associated to f is zero-dimensional and radical and that f reaches its infimum over $\mathbb{R}^n$. We prove in Theorem 5.1.1 that *f is non-negative over $\mathbb{R}^n$ if and only if f is an SOS of polynomials with rational coefficients over the quotient ring $\mathbb{Q}[\boldsymbol{x}]/\mathcal{I}_{\text{grad}}(f)$*. We rely on an algebraic elimination procedure based on the Shape Lemma (see Lemma 3.3.1) to reduce the problem to the univariate case. Interestingly, Theorem 5.1.1 can be applied to Robinson's polynomial (see Example 5.1.1), as well as Scheiderer's polynomial (see Example 5.1.2) which do not have an SOS decomposition with rational coefficients.

Algorithms and bit complexities. The next problem we tackle is to design algorithms that compute such certificates of non-negativity as well as to estimate their bit complexity for non-negative polynomial f that satisfies the conditions in Theorem 5.1.1. To measure the *bitsize* of a polynomial with rational coefficients, we will use its *height*, defined as in Section 4.1.

- From the proof of Theorem 5.1.1, we derive the algorithm `sosgradientshape` (Algorithm 3) to compute an SOS decomposition of polynomials modulo the gradient ideal of f. We prove that, given as input an n-variate polynomial f in $\mathbb{Q}[\boldsymbol{x}]$ of degree d with maximum bitsize of its coefficients τ, this algorithm uses
$$\widetilde{O}((\tau+n+d)^2 d^{6n} + (\tau+n+d)d^{6n+4})$$
boolean operations. This estimate is better than the complexity estimate given in [52, Theorem 12], where the reported number is $\widetilde{O}(\tau^2(4d+2)^{15n+6})$.
- We design a variant of the previous algorithm, named `sosgradient` (Algorithm 4). On input $f \in \mathbb{Q}[\boldsymbol{x}]$ as above, this algorithm decomposes f as a sum of *rational fractions* modulo the gradient ideal associated to f. We prove that this variant uses
$$\widetilde{O}\left((\tau+n+d)d^{4n+4}\right)$$
boolean operations and hence has a better complexity than `sosgradientshape`.

We emphasize that such complexity estimates are of interest to the polynomial optimization community as they give degree bounds for the SOS multipliers required when using the variant of the so-called "Moment-SOS hierarchy" or Lasserre's hierarchy [44] to minimize polynomials over their gradient ideals [62]. Indeed, such degree bounds translate to convergence rates for the underlying optimization scheme and allow us to estimate the overall computational cost complexity.

Experimental results. Both our algorithms have been implemented using the computer algebra system MAPLE. We report on practical experiments, showing that our algorithms can already assess the non-negativity of numerous polynomials that are out of reach of, e.g., hybrid methods computing sums of squares decompositions such as [50].

- Our practical experiments show that *Algorithm* `sosgradient` *can assess the non-negativity of multivariate polynomials of a large set of examples which are out of reach of the state of the art* when both the number of variables and degree increase.

2.3.2 Exact SOHS decompositions of complex univariate polynomials

We design three algorithms to compute exact SOHS decompositions of polynomials in $\mathscr{H}(\mathbb{Z})[z]$ that are *positive* over the unit circle $\mathscr{C}$. These algorithms are based on perturbation-compensation or rounding-projection techniques. Additionally, we analyze their bit complexities and output size as.

The first algorithm is called `csos1` (Algorithm 5). Algorithm `csos1` uses a perturbation-compensation methodology in which the numerical step computes an approximate SOHS decomposition for a well-chosen perturbation of the input polynomial with complex root isolation. We obtain the bit complexity of the algorithm as follows:

- *On input f, where f has no multiple roots,* `csos1` *computes an SOHS decomposition of f with (modulus of) Gaussian coefficients using at most $\widetilde{O}\left(d^6(d+\tau)\right)$ bit operations. In addition, the maximum bitsize of the output coefficients is bounded from above by $\widetilde{O}(d^5(d+\tau))$.*

The two other algorithms are called `csos2` and `csos3` (Algorithms 6 and 7, respectively). We design two algorithms `csos2` and `csos3` which are based on complex semi-definite programming. In Algorithm `csos2`, we compute an approximate SOHS decomposition for the perturbation by using complex SDP solving. Algorithm `csos3` is an adaptation of the rounding-projection algorithm raised by Peyrl and Parrilo [67]. The bit complexities of these two algorithms are similar because we use complex SDP solving for both:

- *On input f,* `csos2` *and* `csos3` *compute SOHS decompositions of f with (modulus of) Gaussian coefficients using at most $\widetilde{O}(d^{13}(d+\tau)^2)$ bit operations. In addition, the maximal bitsize of the output coefficients is bounded from above by $\widetilde{O}(d^6(\tau+d))$.*

These two algorithms are more expensive than Algorithm `csos1` because we replace complex root isolation by complex SDP solving. Despite their worse complexity, they allow one to handle constrained optimization problems and to design filters.

Experimental results. These algorithms have been implemented using the programming environment JULIA [13].

- We report on practical experiments showing that Algorithm `csos1` runs faster than the other algorithms, coinciding with our theoretical complexity results. Furthermore, we rely on `csos3` to design filters in a certified way.

2.4 Organization of the thesis

This thesis includes three parts. Part I is the preliminary part which contains Chapters 3 and 4. In these chapters, we recall basic notions and fundamental results from algebraic geometry and computational commutative algebra, as well as bit complexity results for polynomial system solving.

Our main contributions appear in Part II which includes the following chapters:

- Chapter 5 is dedicated to exact certificates of non-negativity for real multivariate polynomials. The content of this chapter is from the paper entitled "*Sum of squares decompositions of polynomials over their gradient ideals with rational coefficients*" by Victor Magron, Mohab Safey El Din, and Trung Hieu Vu [55] which has been accepted for publication in SIAM Journal of Optimization, 2022.
- Chapter 6 presents results on exact certificates of positivity for complex univariate polynomials. These results have been published in the paper entitled "*Exact SOHS decompositions of trigonometric univariate polynomials with Gaussian coefficients*" [54] by Victor Magron, Mohab Safey El Din, Markus Schweighofer, and Trung Hieu Vu in the Proceedings of the 2022 International Symposium on Symbolic and Algebraic Computation, Villeneuve-d'Ascq, France, 2022.

Finally, in Part III which contains only Chapter 7, we give some conclusions and outline future research directions.

PART I

PRELIMINARIES

CHAPTER 3

Basic notions of algebra and geometry

Contents

This chapter recalls basic notions and results from algebraic geometry and computational commutative algebra such as gradient ideals and varieties, Gröbner bases, and the Shape lemma that are essential for the contribution part. The first two sections are inspired by the monograph by Cox, Little, and O'Shea [20].

Here and subsequently, $\mathbb{K}$ is a field and $\boldsymbol{x} = (x_1, \ldots, x_n)$ is a tuple of n variables. We denote by $\mathbb{K}[\boldsymbol{x}]$ the ring of polynomials in $\boldsymbol{x}$ over $\mathbb{K}$. Every monomial in $\mathbb{K}[\boldsymbol{x}]$ can be written as $\boldsymbol{x}^\alpha = x_1^{\alpha_1} \ldots x_n^{\alpha_n}$, where $\alpha = (\alpha_1, \ldots, \alpha_n) \in \mathbb{N}^n$. One writes a polynomial f as the finite sum of terms $a_\alpha \boldsymbol{x}^\alpha$:

$$f = \sum_\alpha a_\alpha \boldsymbol{x}^\alpha,$$

where $a_\alpha \in \mathbb{K}$ is the coefficient of $\boldsymbol{x}^\alpha$. We denote by $\deg(f)$ the degree of f, i.e., $\deg(f) = \max\{|\alpha| = \alpha_1 + \cdots + \alpha_n : a_\alpha \neq 0\}$.

3.1 Gradient ideals and varieties

This section aims to introduce the main ingredients for Chapter 5 that are gradient ideals and gradient varieties.

Definition 3.1.1 (Ideal)**.** An additive subgroup $\mathcal{I}$ of $\mathbb{K}[\boldsymbol{x}]$ is an *ideal* of $\mathbb{K}[\boldsymbol{x}]$ if $hg \in \mathcal{I}$ for any $h \in \mathcal{I}$ and $g \in \mathbb{K}[\boldsymbol{x}]$.

Given a system of polynomials $g_1, \ldots, g_r$ in $\mathbb{K}[\boldsymbol{x}]$, we denote by $\langle g_1, \ldots, g_r \rangle$ the ideal generated by this system, i.e.,

$$\langle g_1, \ldots, g_r \rangle = \{q_1 g_1 + \cdots + q_r g_r : q_i \in \mathbb{K}[\boldsymbol{x}]\}.$$

According to Hilbert's basis theorem (see, e.g., [20, Ch.2, §5, Thm. 4]), every ideal in $\mathbb{R}[\boldsymbol{x}]$ is a finitely generated ideal, i.e., if $\mathcal{I}$ is an ideal of $\mathbb{K}[\boldsymbol{x}]$ then there exist $g_1, \ldots, g_r \in \mathbb{K}[\boldsymbol{x}]$ such that $\mathcal{I} = \langle g_1, \ldots, g_r \rangle$.

Definition 3.1.2 (Gradient ideal)**.** Let f be a polynomial in $\mathbb{K}[\boldsymbol{x}]$. The *gradient ideal* of f, denoted by $\mathcal{I}_{\text{grad}}(f)$, is the ideal generated by all partial derivatives of f in $\mathbb{K}[\boldsymbol{x}]$, i.e.,

$$\mathcal{I}_{\text{grad}}(f) := \left\langle \frac{\partial f}{\partial x_1}, \ldots, \frac{\partial f}{\partial x_n} \right\rangle.$$

Example 3.1.1. We consider the polynomial f_E in two variables x_1, x_2 of degree 4 with real coefficients

$$f_E = 2x_1^2 + 4x_1x_2 + x_2^4 + 3. \tag{3.1}$$

The gradient ideal $\mathcal{I}_{\text{grad}}(f_E)$ is

$$\mathcal{I}_{\text{grad}}(f_E) = \left\langle 4x_1 + 4x_2, 4x_1 + 4x_2^3 \right\rangle. \tag{3.2}$$

Definition 3.1.3. (Algebraic variety) Let $\mathcal{I}$ be an ideal of $\mathbb{R}[\boldsymbol{x}]$. The *algebraic variety* associated to $\mathcal{I}$ is defined as

$$V(\mathcal{I}) := \{a \in \mathbb{C}^n : \forall g \in \mathcal{I},\ g(a) = 0\}. \tag{3.3}$$

From the definition, the algebraic variety associated to $\mathcal{I}$ is the set of complex points on which the polynomials in $\mathcal{I}$ simultaneously vanish.

Definition 3.1.4 (Gradient variety)**.** Let f be a polynomial in $\mathbb{K}[\boldsymbol{x}]$. The *gradient variety* of f is respectively the algebraic variety associated to $\mathcal{I}_{\text{grad}}(f)$. We denote $V_{\text{grad}}(f)$ by the gradient variety associated to f.

Example 3.1.2. An easy computation shows that the gradient variety of the polynomial f_E in Example 3.1.1 is the following set that has only real points

$$V_{\text{grad}}(f_E) = \{(0,0), (1,1), (1,-1)\}.$$

Definition 3.1.5 (Zero-dimensional ideal)**.** The ideal $\mathcal{I}$ is *zero-dimensional* if $V(\mathcal{I})$ is finite and non-empty.

The dimension of the empty set is conventionally -1. It is worth noting here that if $\mathcal{I}$ is zero-dimensional then we can get a bound on the expected cardinality of $V(\mathcal{I})$, denoted by $\#V(\mathcal{I})$, from Bezout's theorem [20, Ch.8, §7, Thm. 10]. In particular, if the gradient ideal $\mathcal{I}_{\text{grad}}(f)$ is zero-dimensional, then $\#V(\mathcal{I}_{\text{grad}}(f))$ is bounded from above by $(d-1)^n$, where d is the degree of f.

We now recall some useful terminology concerning quotients of polynomial rings.

Definition 3.1.6 (Congruent modulo)**.** Let $\mathcal{I}$ be an ideal, and let p, q in $\mathbb{K}[\boldsymbol{x}]$. One says that p and q are *congruent modulo* $\mathcal{I}$, denoted by $p \equiv a \mod \mathcal{I}$, if $p - q$ belongs to $\mathcal{I}$.

Note that congruence modulo $\mathcal{I}$ is an equivalence relation on $\mathbb{K}[\boldsymbol{x}]$. The quotient of $\mathbb{K}[\boldsymbol{x}]$ modulo $\mathcal{I}$, denoted by $\mathbb{K}[\boldsymbol{x}]/\mathcal{I}$, is the set of equivalence classes of the equivalence relation modulo $\mathcal{I}$:

$$\mathbb{K}[\boldsymbol{x}]/\mathcal{I} = \{[q] : q \in \mathbb{K}[\boldsymbol{x}]\},$$

where $[q]$ is the equivalence class of q, i.e., $[q] = \{p \in \mathbb{R}[\boldsymbol{x}] : p \equiv q \mod \mathcal{I}\}$. On the quotient $\mathbb{K}[\boldsymbol{x}]/\mathcal{I}$, we can define sum and product operations as follows:

$$[p] + [q] := [p + q],\ [p] \cdot [q] := [p \cdot q],$$

for any p, q in $\mathbb{R}[\boldsymbol{x}]$. The quotient $\mathbb{K}[\boldsymbol{x}]/\mathcal{I}$ is a commutative ring under the operations given above.

Let $\mathcal{V} \subset \mathbb{C}^n$ be a variety. The set of all polynomials that vanish on $\mathcal{V}$ is an ideal of $\mathbb{K}[\boldsymbol{x}]$

$$I(\mathcal{V}) := \{q \in \mathbb{R}[\boldsymbol{x}] : q(x) = 0 \text{ for all } x \in \mathcal{V}\}. \tag{3.4}$$

Given an ideal $\mathcal{I}$ of $\mathbb{R}[\boldsymbol{x}]$, its *radical* is the ideal

$$\sqrt{\mathcal{I}} := \{q \in \mathbb{R}[\boldsymbol{x}] : q^m \in \mathcal{I} \text{ for some } m \in \mathbb{N}\}.$$

Clearly, $\mathcal{I} \subset \sqrt{\mathcal{I}}$. One has the following definition for the case that the converse inclusion holds.

Definition 3.1.7 (Radical ideal)**.** The ideal $\mathcal{I}$ is *radical* if $\mathcal{I} = \sqrt{\mathcal{I}}$.

Example 3.1.3. Consider the ideal $\mathcal{I} = \langle x_1^3 x_2 \rangle$ in $\mathbb{R}[x_1, x_2]$. Its radical is $\sqrt{\mathcal{I}} = \langle x_1 x_2 \rangle$. Clearly, $x_1^2 x_2$ belongs to $\sqrt{\mathcal{I}}$ but it does not belong to $\mathcal{I}$. This implies $\mathcal{I} \neq \sqrt{\mathcal{I}}$ and therefore $\mathcal{I}$ is not radical.

Hilbert's Strong Nullstellensatz [20, Ch. 4, §2, Thm. 6]) tells us that if a polynomial g vanishes at all points of the variety $V(\mathcal{I})$ then some power of g itself must belong to $\sqrt{\mathcal{I}}$. This property will be used in the proof of Theorem 5.2.1 in Chapter 5.

Theorem 3.1.1 (Hilbert's Strong Nullstellensatz)**.** *Let $\mathcal{I}$ be an ideal in $\mathbb{R}[\boldsymbol{x}]$. Then, $I(V(\mathcal{I})) = \sqrt{\mathcal{I}}$.*

Let $\mathcal{I}$ be radical. According to Hilbert's Strong Nullstellensatz, $I(V(\mathcal{I})) = \mathcal{I}$. This implies that if a polynomial g vanishes at all points of $V(\mathcal{I})$ then g must belong to $\mathcal{I}$.

The most important consequence of the Nullstellensatz is that it allows us to build a strong relationship between algebra and geometry. In particular, if we restrict to radical ideals of $\mathbb{R}[\boldsymbol{x}]$, then the maps V and I, defined respectively in (3.3) and (3.4), are inverses of each other, and they define bijections between the set of radical ideals and varieties [20, Ch. 4, §2, Thm. 7].

3.2 Gröbner bases of ideals

We give the definition of a Gröbner basis of an ideal, their elementary properties, and a criterion to check whether a given generating set of an ideal is a Gröbner basis. These will be ingredients in the next section (Section 3.3) and will be used to prove Theorems 5.1.1 and 5.2.1 in Chapter 5.

Encoding a monomial $\boldsymbol{x}^{\alpha}$ by a n-tuple α in $\mathbb{N}^n$ allows us to construct a one-to-one correspondence between the monomials and $\mathbb{N}^n$. Hence, if we have an ordering $<$ on $\mathbb{N}^n$, then we can build an ordering $<$ on the monomials of $\mathbb{K}[\boldsymbol{x}]$.

Definition 3.2.1. A *monomial ordering* $<$ on $\mathbb{K}[\boldsymbol{x}]$ is a relation $<$ on $\mathbb{N}^n$ satisfying the three following conditions:

1. $<$ is a total ordering on $\mathbb{N}^n$, i.e., $<$ is transitive, and for every pair α, β in $\mathbb{N}^n$, exactly one of the three statements $\alpha < \beta$, $\beta < \alpha$, and $\alpha = \beta$ holds;
2. $<$ is a well-ordering on $\mathbb{N}^n$, i.e., every non-empty subset of $\mathbb{N}^n$ has a smallest element under the relation $<$;
3. If $\alpha < \beta$ and $\gamma \in \mathbb{N}^n$ then $\alpha + \gamma < \beta + \gamma$.

We write $x^{\alpha} < x^{\beta}$ if $\alpha < \beta$.

Example 3.2.1. We consider two important monomial orderings on $\mathbb{K}[\boldsymbol{x}]$ that will be used in the thesis. First, one says $\alpha <_{\text{lex}} \beta$ if the leftmost non-zero entry of $\beta - \alpha$ is positive; This relation is a monomial ordering on $\mathbb{N}^n$ called *lexicographic order*. Second, one says $\alpha <_{\text{grlex}} \beta$ if $|\alpha| < |\beta|$, or $|\alpha| = |\beta|$ and $\alpha <_{\text{lex}} \beta$; in this case, $<_{\text{grlex}}$ is a *graded lexicographic order* on $\mathbb{N}^n$.

Let $<$ be a monomial ordering on $\mathbb{K}[\boldsymbol{x}]$ and $\mathcal{I} \neq \{0\}$ be an ideal. We denote by $\text{LT}(f)$ the *leading term* of $f \in \mathcal{I}$ with respect to $<$, i.e., if $f = \sum_{\alpha} a_{\alpha} x^{\alpha}$ then $\text{LT}(f) = a_{\beta} x^{\beta}$ such that $a_{\beta} \neq 0$ and $\alpha < \beta$ for any α with $a_{\alpha} \neq 0$. In this case, $\text{mdeg}(f) = \beta$ is called the *multi-degree* of f, and a_{β} is called the *leading coefficient* of f. Denote by $\text{LT}(\mathcal{I})$ the set of all leading terms $\text{LT}(g)$ of $g \in \mathcal{I}$ with respect to $<$, and by $\langle \text{LT}(\mathcal{I}) \rangle$ the ideal generated by the elements of $\text{LT}(\mathcal{I})$.

Definition 3.2.2 (Gröbner basis)**.** Fix a monomial order $<$ on $\mathbb{K}[\boldsymbol{x}]$. A finite subset $G = \{g_1, \ldots, g_r\}$ of $\mathcal{I}$ is said to be a *Gröbner basis* of $\mathcal{I}$ with respect to the order $<$ if

$$\langle \text{LT}(g_1), \ldots, \text{LT}(g_r) \rangle = \langle \text{LT}(\mathcal{I}) \rangle .$$

Using the convention that $\langle \emptyset \rangle = \{0\}$, we define the empty set to be the Gröbner basis of the zero ideal.

In the following example, we will give a subset that is not a Gröbner basis . A subset that is a Gröbner basis will be shown later.

Example 3.2.2. ([20, Ch. 2 §5, Example 2]) We consider the ideal $\mathcal{I} = \langle g_1, g_2 \rangle$, where

$$g_1 = x_1^3 - 2x_1x_2,\ g_2 = x_1^2x_2 - 2x_2^2 + x_1,$$

and use the graded lexicographic ordering on monomials in $\mathbb{R}[x_1, x_2]$. We can see that $x_1^2 = x_1g_2 - x_2g_1 \in \mathcal{I}$, hence $x_1^2 = \mathrm{LT}(x_1^2) \in \langle \mathrm{LT}(\mathcal{I}) \rangle$. However, x_1^2 is not divisible by $\mathrm{LT}(g_1) = x_1^3$ or $\mathrm{LT}(g_2) = x_1^2x_2$. According to [20, Ch. 2 §2, Lemma 2], x_1^2 does not belong to $\langle \mathrm{LT}(g_1), \mathrm{LT}(g_2) \rangle$. This implies that $\{g_1, g_2\}$ is not a Gröbner basis for $\mathcal{I}$.

Assume that G is a Gröbner basis of $\mathcal{I}$ with respect to some monomial order $<$ and that p is a polynomial in $\mathbb{K}[\boldsymbol{x}]$. The remainder of the division of p by G does not depend on how the elements of G are listed. Hence, the remainder is unique. Furthermore, the remainder is zero if and only if p belongs to the ideal.

Note that every ideal $\mathcal{I}$ in $\mathbb{K}[\boldsymbol{x}]$ has a Gröbner basis. If $G = \{g_1, \ldots, g_r\}$ is a given basis of $\mathcal{I}$ and $<$ is a given monomial order, Buchberger's Algorithm [20, Ch. 2 §7] allows us to compute a Gröbner basis with respect to $<$ based on G. Moreover, Buchberger's Criterion tells us whether a given generating set of an ideal is a Gröbner basis.

We now recall Buchberger's Criterion that will be used in the proof of Theorems 4.3.2 and 5.2.1.

Suppose that polynomials $p, q \in \mathbb{K}[\boldsymbol{x}]$ have multi-degrees $\mathrm{mdeg}(p) = \alpha = (\alpha_1, \ldots, \alpha_n)$ and $\mathrm{mdeg}(p) = \beta = (\beta_1, \ldots, \beta_n)$ with respect to the monomial order $<$. We denote $\gamma_i := \max\{\alpha_i, \beta_i\}$ and $\boldsymbol{x}^\gamma$ is the least common multiple of $\mathrm{LT}(p)$ and $\mathrm{LT}(q)$, written $\boldsymbol{x}^\gamma = \mathrm{lcm}(\mathrm{LT}(p), \mathrm{LT}(q))$. The S-polynomial of p and q is the combination

$$S(p, q) = \frac{\boldsymbol{x}^\gamma}{\mathrm{LT}(p)} \times p - \frac{\boldsymbol{x}^\gamma}{\mathrm{LT}(q)} \times q.$$

Theorem 3.2.1 (Buchberger's Criterion)**.** *Let $G = \{g_1, \ldots, g_r\}$ be a basis of $\mathcal{I}$. Then, G is a Gröbner basis if and only if, for any pair $i \neq j$, the remainder of the division of $S(g_i, g_j)$ by G is zero.*

The following example illustrates Buchberger's Criterion.

Example 3.2.3. We consider the ideal $\mathcal{I}_{\mathrm{grad}}(f_E) = \langle g_1, g_2 \rangle$ defined in Example 3.1.1, where $g_1 = x_1 + x_2, g_2 = x_1 + x_2^3$ with the graded lexicographical order $x_1 < x_2$. We see that $\mathrm{LT}(g_1) = x_2$, $\alpha = \mathrm{mdeg}(g_1) = (0, 1)$, $\mathrm{LT}(g_2) = x_2^3$, $\beta = \mathrm{mdeg}(g_2) = (0, 3)$; thus, $\gamma = (0, 3)$. The S-polynomial of g_1 and g_2 is

$$S(g_1, g_2) = \frac{x_2^3}{x_2} \times (x_1 + x_2) - \frac{x_2^3}{x_2^3} \times (x_1 + x_2^3) = x_1x_2^2 - x_1.$$

According to Buchberger's Criterion, $\{g_1, g_2\}$ is not a Gröbner basis of $\mathcal{I}_{\text{grad}}(f_E)$.

Definition 3.2.3. A Gröbner basis G is *reduced* if the leading coefficient of g is 1, for all $g \in G$, and there are no monomials of g lying in $\langle \text{LT}(G) \setminus \{g\} \rangle$.

For a given monomial ordering, every ideal $\mathcal{I}$ has a unique reduced Gröbner basis.

Example 3.2.4. We consider the ideal $\mathcal{I} = \langle g_1, g_2 \rangle$, where $g_1 = x_1^3 - x_1$ and $g_2 = x_2 + x_1$ with the graded lexicographical order $x_1 < x_2$. Using Buchberger's Criterion, we can show that $\{g_1, g_2\}$ is a Gröbner basis of $\mathcal{I}$. Furthermore, we can check that this is the reduced Gröbner basis.

3.3 Shape position of a zero-dimensional and radical ideal

This section introduces notions and results concerning the so-called Shape Lemma that is the key to reduce a multivariate problem to a univariate one in our research. We will exploit these results in the proofs of the existence of SOS decompositions in Theorems 5.1.1 and 5.2.1 in Chapter 5.

Assume that $\mathcal{I}$ is a zero-dimensional and radical ideal in $\mathbb{Q}[\boldsymbol{x}]$ and that G is the reduced Gröbner basis of $\mathcal{I}$ with respect to the lexicographical order $x_1 < \cdots < x_n$.

Definition 3.3.1. One says that $\mathcal{I}$ is in *shape position* if G has the form:

$$G = [w, x_2 - v_2, \ldots, x_n - v_n], \tag{3.5}$$

where $w, v_2, \ldots, v_n$ are polynomials in $\mathbb{K}[x_1]$ and $\deg w = \#V(\mathcal{I})$.

The following lemma, named Shape Lemma, gives us a criterion for being in shape position of an ideal.

Lemma 3.3.1 (Shape Lemma, [26]). *Let $\mathcal{I}$ be a zero-dimensional and radical ideal and $<$ be a lexicographic monomial order in $\mathbb{Q}[\boldsymbol{x}]$. If $V(\mathcal{I})$ is the union of δ points in $\mathbb{C}^n$ with distinct x_1-coordinates, then $\mathcal{I}$ is in shape position as in (3.5), where $v_2, \ldots, v_n$ are polynomials in $\mathbb{Q}[x_1]$ of degrees at most $\delta - 1$.*

Example 3.3.1. Consider the ideal $\mathcal{I}_{\text{grad}}(f_E)$ defined in (3.2) which is zero-dimensional and radical with the graded lexicographical order $x_1 < x_2$. An easy computation shows that

$$\mathcal{I}_{\text{grad}}(f_E) = \left\langle x_1 + x_2, x_1 + x_2^3 \right\rangle = \left\langle x_1^3 - x_1, x_2 + x_1 \right\rangle.$$

We see that $\mathcal{I}_{\text{grad}}(f_E)$ is in shape position as in (3.5), where $w(x_1) = x_1^3 - x_1$, $\delta = 3$ and $v_2 = -x_1$ of degree 1.

Definition 3.3.2. Let $\mathcal{V}$ be a zero-dimensional algebraic subset of $\mathbb{C}^n$. A *zero-dimensional rational parametrization* $\mathcal{Q} = ((w, \kappa_1, \ldots, \kappa_n), \lambda)$ of $\mathcal{V}$ consists of $n+1$ univariate polynomials $w, \kappa_1, \ldots, \kappa_n$ in $\mathbb{Q}[t]$ such that w is monic and square-free, $\deg \kappa_i < \deg w$, for $i = 1, \ldots, n$, and a $\mathbb{Q}$-linear form λ in n variables satisfying $\lambda(\kappa_1, \ldots, \kappa_n) = tw' \mod w$, where w' is the derivative of w, such that

$$\mathcal{V} = \left\{ \left(\frac{\kappa_1(t)}{w'(t)}, \ldots, \frac{\kappa_n(t)}{w'(t)} \right) : w(t) = 0 \right\}.$$

The condition on the linear form λ states that the roots of w are precisely the values taken by λ on $\mathcal{V}$, and that λ separates $\mathcal{V}$, i.e., $\lambda(x) \neq \lambda(y)$ for any distinct pair x, y in $\mathcal{V}$.

Note that there exist algorithms to compute a zero-dimensional rational parametrization of a zero-dimensional algebraic subset given by a polynomial sequence with rational coefficients [75, 27].

Example 3.3.2. Consider the ideal $\mathcal{I}_{\text{grad}}(f_E)$ given in (3.2) which is zero-dimensional. Its variety has a zero-dimensional rational parametrization $((w, \kappa_1, \kappa_2), \lambda)$ given by

$$\lambda = x_1,\ w = t^3 - t,\ \kappa_1 = -t_1,\ \kappa_2 = 2t. \tag{3.6}$$

One has $w' = 3t^2 - 1$ and

$$V_{\text{grad}}(f_E) = \left\{ \left(\frac{-2t}{3t^2-1}, \frac{2t}{3t^2-1} \right) : t^3 - t = 0 \right\}. \tag{3.7}$$

The polynomial w has three real roots which are $-1, 0$, and 1. By replacing t with these values in (3.7), we obtain the variety $V_{\text{grad}}(f_E)$ as follows:

$$V_{\text{grad}}(f_E) = \{(0,0), (1,1), (1,-1)\}.$$

This set coincides with the set computed in Example 3.1.2.

The following lemma points out the explicit shape position of a zero-dimensional and radical ideal $\mathcal{I}$ through a zero-dimensional rational parametrization of its variety. Importantly, the proof shows us how to compute the reduced Gröbner basis of $\mathcal{I}$ with respect to a given lexicographical order.

Lemma 3.3.2. *Let $\mathcal{I}$ be a zero-dimensional and radical ideal, and $<$ be a lexicographic monomial order in $\mathbb{Q}[\boldsymbol{x}]$. Assume that $\mathcal{Q} = ((w, \kappa_1, \ldots, \kappa_n), x_1)$ is a zero-dimensional rational parametrization of $\mathcal{V} = V(\mathcal{I})$. Then, there exist polynomials $w, v_2, \ldots, v_n$ in $\mathbb{Q}[x_1]$ satisfying $\deg v_i < \deg w$, for $i = 2, \ldots, n$, such that $\mathcal{I} = \langle w, x_2 - v_2, \ldots, x_n - v_n \rangle$.*

Proof. Because w is square-free and w' is the derivative of w, one sees that the gcd of w and w' is 1. From the extended Euclidean algorithm [85, Algorithm 3.14], there exist two

Bézout coefficients of w and w', namely a, b in $\mathbb{Q}[x_1]$, with $aw + bw' = 1$. For $i = 2, \ldots, n$, we see that $w'x_i(t) = \kappa_i(t)$ for any t satisfying $w(t) = 0$. As $\deg \kappa_i < \deg w$ and the linear form $\lambda = x_1$ separates $\mathcal{V}$, we have $w'x_i = \kappa_i$. This yields $bw'x_i = b\kappa_i$. Since $bw' = 1 - aw$, we observe that $x_i - awx_i = b\kappa_i$ and, hence, $x_i = b\kappa_i \mod w$. By denoting $v_i := b\kappa_i \mod w$, we obtain $w, v_2, \ldots, v_n$ which are the desired polynomials. □

Example 3.3.3. Consider the ideal $\mathcal{I}_{\text{grad}}(f_E)$ in Example 3.3.2. The zero-dimensional rational parametrization of $V_{\text{grad}}(f_E)$ is given in (3.6). Apply the procedure in the proof of Lemma 3.3.2, we obtain $w = x_1^3 - x_1$ and $v_2 = -x_1$.

CHAPTER 4

Bit complexity results for polynomial system solving

Contents

To measure the complexity of algorithms we use bit complexity. In this chapter, we provide fundamental results on bit complexity analysis of algorithms concerning univariate and multivariate divisions, computing zero-dimensional rational parametrizations, solving semi-definite programs, and computing lower bounds of the minimum of a polynomial on the unit circle. These are tools we rely on to investigate the bit complexity of our new algorithms.

4.1 Bitsize of polynomials with rational coefficients

We use the *height* of a polynomial with rational coefficients to measure its *bitsize* that is defined as follows. The bitsize of an integer b is denoted by $\mathrm{ht}(b) := \lfloor \log_2(|b|) \rfloor + 1$ when $b \neq 0$ and by $\mathrm{ht}(0) := 1$, where $\log_2$ is the binary logarithm. Given $a \in \mathbb{Z}$ and $b \in \mathbb{Z}$ with $b \neq 0$ and $\gcd(a, b) = 1$, we define $\mathrm{ht}(a/b) = \max(\mathrm{ht}(a), \mathrm{ht}(b))$. If $a, b \in \mathbb{Q}$, we define the bitsize of a Gaussian rational number as $\mathrm{ht}(a + \boldsymbol{i}b) = \max(\mathrm{ht}(a), \mathrm{ht}(b))$, where $\boldsymbol{i}$ is the imaginary unit. For a non-zero polynomial f with (Gaussian) rational coefficients, we define the height of the polynomial f, denoted by $\mathrm{ht}(f)$, as the maximum bitsize of the non-zero coefficients of f.

For two maps $p, q : \mathbb{N}^m \to \mathbb{R}$, one writes "$p(v) = O(q(v))$" when there exists $b \in \mathbb{N}$ such that $p(v) \leq bq(v)$, for all $v \in \mathbb{N}^m$. We use the notation $p(v) = \widetilde{O}(q(v))$ when $p(v) = O(q(v) \log^k q(v))$ for some $k \in \mathbb{N}$.

4.2 SOS decomposition of non-negative univariate polynomials

In our study, the key idea to handle the multivariate case of computing SOS decompositions is reducing the problem to the univariate case. Hence, we need to recall known results related to SOS decompositions of univariate polynomials which will be used in the proof of Theorems 5.1.1 and 5.2.1 in Chapter 5.

It is well-known that $f \in \mathbb{R}[t]$ is non-negative over $\mathbb{R}$ if and only if f is SOS. This property holds also for polynomials with coefficients in a subfield $\mathbb{K}$ of $\mathbb{R}$. More precisely, we have the following theorem:

Theorem 4.2.1 ([43, 70]). *Let $\mathbb{K}$ be a subfield of $\mathbb{R}$ and $f \in \mathbb{K}[t]$. Then, f is non-negative over $\mathbb{R}$ if and only if f admits a weighted SOS decomposition of polynomials in $\mathbb{K}[t]$, i.e., there exists a positive integer s, non-negative numbers $c_1, \ldots, c_s \in \mathbb{K}$ and polynomials $g_1, \ldots, g_s \in \mathbb{K}[t]$, such that $f = \sum_{j=1}^{s} c_j g_j^2$.*

Example 4.2.1. Consider the univariate polynomial $h(t) = 2t^4 - t^2 + 10$ in $\mathbb{Q}[t]$. One has the following two weighted rational SOS decompositions of h as follows:

$$h = 2(t - \frac{1}{4})^2 + \frac{79}{8} = \frac{1}{2}t^4 + \frac{3}{2}(t^2 - \frac{5}{2})^2 + \frac{13}{2}t^2 + \frac{5}{8}.$$

To compute an SOS decomposition of a non-negative univariate polynomial with rational coefficients, we use the algorithm by Schweighofer [80] named `univsos1` or the algorithm by Chevillard, Harrison, Joldes, and Lauter [18] named `univsos2`. Here, we need to recall their bit complexities. From the theoretical results below, it is worth mentioning that `univsos2` runs faster than `univsos1`.

Theorem 4.2.2 ([53], Theorems 16–17). *Let $h \in \mathbb{Z}[t]$ be a non-negative univariate polynomial of degree d and bitsize τ. Then, on input h, `univsos1` runs in*

$$\widetilde{O}\Big(\Big(\frac{d}{2}\Big)^{\frac{3d}{2}}\tau\Big)$$

boolean operations and the maximum bitsize of the coefficients involved in the SOS decomposition is bounded from above by

$$O\Big(\Big(\frac{d}{2}\Big)^{\frac{3d}{2}}\tau\Big).$$

Theorem 4.2.3 ([53], Theorems 23–24). *Let $h \in \mathbb{Z}[t]$ be a non-negative univariate polynomial of degree d and bitsize τ. Then, on input h, `univsos2` runs in $\widetilde{O}\left(d^3 + d^2\tau\right)$ boolean operations and the maximum bitsize of the coefficients involved in the SOS decomposition is bounded from above by $O\left(d^3 + d^2\tau\right)$.*

4.3 Univariate and multivariate division algorithms

We now establish the bit complexity of the Euclidean division algorithm and a multivariate division algorithm which will be used later on to investigate the bit complexity of our algorithms.

4.3.1 Euclidean division algorithm

Lemma 4.3.1. *Let a, b be polynomials in $\mathbb{Z}[t]$, with $\deg a = d \geq m = \deg b$, and τ be an upper bound of $\mathrm{ht}(a)$ and $\mathrm{ht}(b)$. To compute the quotient q and the remainder r of the division of a by b, we use the Euclidean division algorithm [85, Algorithm 2.5]. This algorithm uses $O\left(m\tau(d-m)^2\right)$ boolean operations. Furthermore, both bitsizes of q and r are bounded from above by $O\left(\tau(d-m)\right)$.*

We recall the Euclidean division algorithm [85, Algorithm 2.5] in Algorithm 1 to compute the quotient q and the remainder r of the division of a by b, i.e., $a = qb + r$ with $\deg r < \deg b$.

Algorithm 1 Euclidean division algorithm

Input: polynomials $a, b \in \mathbb{Z}[t]$
Output: polynomials $q, r \in \mathbb{Q}[t]$ such that $a = qb + r$ and $\deg r < \deg b$

1: Set $q := 0$ and $r := a$
2: **while** $\deg r \geq \deg b$ **do**
3: Set $h := \mathtt{lc}(r)/\mathtt{lc}(b)t^{\deg r - \deg b}$
4: Set $q := q + h$
5: Set $r := r - hb$
6: **done**
7: **return** q and r

Proof. Assume that a, b are polynomials in $\mathbb{Z}[t]$ with $\deg a = d \geq \deg b = m$ and that $\mathrm{ht}(a)$, $\mathrm{ht}(b)$ are bounded from above by τ. We denote by r_i (resp. q_i, h_i) the value of r (resp. q, h) after the i-th iteration of the while loop from Line 2. The initial values are $q_0 = 0$ and $r_0 = a$. After each iteration of the while loop, the degree of r is strictly decreasing. Hence, the while loop will terminate after k iterations, where $k \leq d - m$.

We now compute the numbers of boolean operations to perform the operations in Lines 3–5. From $h_i = \mathtt{lc}(r_{i-1})/\mathtt{lc}(b)t^{\deg r_{i-1} - \deg b}$ in Line 3, we observe that

$$\mathrm{ht}(h_i) = \max\{\mathrm{ht}(b), \mathrm{ht}(r_{i-1})\} \leq \max\{\tau, \mathrm{ht}(r_{i-1})\} \leq \tau + \mathrm{ht}(r_{i-1}), \tag{4.1}$$

and the number of boolean operations to perform Line 3 is bounded by $\tau + \mathrm{ht}(r_{i-1})$. Note that the number of boolean operations to perform the operation in Line 4 is bounded by $O(1)$. We consider the operation in Line 5, i.e., $r_i = r_{i-1} - h_i b$. The estimate in (4.1) implies $\mathrm{ht}(h_i b) \leq 2\tau + \mathrm{ht}(r_{i-1})$; then the bitsize of r_i is bounded by $2\tau + \mathrm{ht}(r_{i-1})$. We get the recurrence formula $\mathrm{ht}(r_{i+1}) \leq \mathrm{ht}(r_i) + 2\tau$, for each $i = 0, \ldots, k$, with $\mathrm{ht}(r_0) = \tau$. It follows that $\mathrm{ht}(r_i) \leq 2i\tau + \tau$, for each $i = 0, \ldots, k$. This yields

$$\mathrm{ht}(r) = \mathrm{ht}(r_k) \leq 2(d-m)\tau + \tau = O((d-m)\tau).$$

In Line 5, the number of boolean operations to compute $h_i b$ is $O(m(\tau + \mathrm{ht}(r_{i-1})))$, so r_i is also computed in $O(m(\tau + \mathrm{ht}(r_{i-1})))$ boolean operations.

From above, to compute every iteration in Line 2, we need $O(m\tau(d-m))$ the boolean operations. Since the algorithm has at most $d-m$ iterations, the number of boolean operations to perform the algorithm is $O(m\tau(d-m)^2)$.

To complete the proof, we estimate the bitsize of q. Since $q_i = q_{i-1} + h_i$, from (4.1), one has

$$\mathrm{ht}(q_i) \leq \max\{\mathrm{ht}(q_{i-1}), \mathrm{ht}(h_i)\} \leq \mathrm{ht}(q_{i-1}) + \tau + \mathrm{ht}(r_{i-1}).$$

This yields $\mathrm{ht}(q) \leq (d-m)\tau + \mathrm{ht}(r) = O((d-m)\tau)$. This is the desired estimate. □

4.3.2 Multivariate division algorithm `Eliminate`

Denote by $\mathbb{Q}(x_1)$ the field of rational fractions in variable x_1 with coefficients in $\mathbb{Q}$. With the lexicographic monomial order $x_2 < \cdots < x_n$, we consider the standard multivariate division [20, Ch. 2, Sec 3.] of $g \in \mathbb{Q}[x_1][x_2, \ldots, x_n]$ by the list

$$[x_2 - \frac{a_2}{a_0}, \ldots, x_n - \frac{a_n}{a_0}],$$

where $a_0, a_2, \ldots, a_n \in \mathbb{Q}[x_1]$. To compute the quotients $\phi_2, \ldots, \phi_n \in \mathbb{Q}(x_1)[x_2, \ldots, x_n]$ and remainder $r \in \mathbb{Q}(x_1)$, here

$$g = \sum_{i=2}^{n} \phi_i (x_i - \frac{a_i}{a_0}) + r, \tag{4.2}$$

we iterate classical univariate divisions by $x_i - \frac{a_i}{a_0}$ for $i = n, \ldots, 2$ considering them as univariate in x_i so that we eliminate step by step the variables $x_n, \ldots, x_2$ in g. The details of this algorithm, which we name `Eliminate`, are given in Algorithm 2. The inputs of `Eliminate` are $g, a_0, a_2, \ldots, a_n$ and its outputs are the quotients $[\phi_2, \ldots, \phi_n]$ and the remainder r.

Theorem 4.3.2. *Let g be in $\mathbb{Q}[x_1][x_2, \ldots, x_n]$, and $a_0, a_2, \ldots, a_n$ be in $\mathbb{Q}[x_1]$. We consider the lexicographic monomial order $x_2 < \cdots < x_n$ on $\mathbb{Q}(x_1)[x_2, \ldots, x_n]$. On input $g, a_0, a_2, \ldots, a_n$,* `Eliminate` *terminates and outputs quotients $\phi_2, \ldots, \phi_n \in \mathbb{Q}(x_1)[x_2, \ldots, x_n]$ and remainder $r \in \mathbb{Q}(x_1)$ satisfying (4.2).*

Algorithm 2 Elimination algorithm

`Eliminate` := proc$(g, a_0, a_2, \ldots, a_n)$

Input: $n+1$ polynomials $g \in \mathbb{Q}[x_1][x_2, \ldots, x_n]$, $a_0, a_2, \ldots, a_n \in \mathbb{Q}[x_1]$

Output: $\phi_2, \ldots, \phi_n$ in $\mathbb{Q}(x_1)[x_2, \ldots, x_n]$ and $r \in \mathbb{Q}(x_1)$ satisfying (4.2)

1: Set $r_{n+1} := g$
2: **for** $i = n$ to 2 **do**
3: Compute $\phi_i := \text{quo}(r_{i+1}, x_i - \frac{a_i}{a_0}, x_i)$
4: Substitute x_i by $\frac{a_i}{a_0}$ in r_{i+1} to define $r_i := r_{i+1}(x_1, \ldots, x_{i-1}, \frac{a_i}{a_0})$
5: **done**
6: Set $r := r_2$
7: **return** $[\phi_2, \ldots, \phi_n]$, and r

Proof. Let us consider the list of polynomials in $\mathbb{Q}(x_1)[x_2, \ldots, x_n]$:

$$G = [x_2 - \frac{a_2}{a_0}, \ldots, x_n - \frac{a_n}{a_0}],$$

where $a_0, a_2, \ldots, a_n$ are polynomials in $\mathbb{Q}[x_1]$, with $a_0 \neq 0$. Algorithm `Eliminate` outputs $[\phi_2, \ldots, \phi_n] \subset \mathbb{Q}(x_1)[x_2, \ldots, x_n]$ and $r \in \mathbb{Q}(x_1)$. We now prove that these polynomials are the quotients and remainder of the multivariate division of g by the list G, i.e., they satisfy (4.2).

In Line 3, ϕ_i is the quotient of the univariate division (in the variable x_i) of r_{i+1} by $x_i - \frac{a_i}{a_0}$. Since the degree of x_i in $x_i - \frac{a_i}{a_0}$ is 1, ϕ_i belongs to $\mathbb{Q}(x_1)[x_2, \ldots, x_i]$. The remainder r_i of the division in Line 3 is given in Line 4 after replacing x_i by $\frac{a_i}{a_0}$ in r_{i+1}. Hence one has $r_i \in \mathbb{Q}(x_1)[x_2, \ldots, x_{i-1}]$. After Lines 3-4, we obtain

$$r_{i+1} = \phi_i\Big(x_i - \frac{a_i}{a_0}\Big) + r_i. \tag{4.3}$$

Therefore, after Line 6, we get $g = \sum_{i=2}^{n} \phi_i(x_i - \frac{a_i}{a_0}) + r$, with $r \in \mathbb{Q}(x_1)$. Based on Buchberger's Criterion (Theorem 3.2.1), we can see that the system of $n-1$ polynomials

$$[x_2 - \frac{a_2}{a_0}, \ldots, x_n - \frac{a_n}{a_0}]$$

is a Gröbner basis of the ideal generated by this system with respect to the lexicographic monomial order $x_2 < \cdots < x_n$ in $\mathbb{Q}(x_1)[x_2, \ldots, x_n]$. Hence, $\phi_2, \ldots, \phi_n$ and r are uniquely defined. The correctness of the algorithm is proved. □

To illustrate how `Eliminate` works, we consider a simple example.

Example 4.3.1. Consider polynomials $g = x_1^2 + x_1x_2 + 2x_2^2$ in $\mathbb{Q}[x_1, x_2]$ and $a_0 = x_1^2 + 1$, and $a_2 = x_1$ in $\mathbb{Q}[x_1]$. We will perform the division of g by $x_2 - a_2/a_0$. Since $n = 2$, we

only need to compute the quotient ϕ_2 and remainder r,

$$g = \phi_2\Big(x_2 - \frac{a_2}{a_0}\Big) + r.$$

By performing Line 3, we obtain

$$\phi_2 = \mathrm{quo}(g, x_2 - \frac{a_2}{a_0}, x_2) = x_1 + 2x_2 + \frac{2x_1}{x_1^2+1}.$$

The remainder r is computed by performing Line 4. One has that

$$r = g(x_1, \frac{a_2}{a_0}) = x_1 + \frac{x_1^2}{x_1^2+1} + \frac{2x_1^2}{(x_1^2+1)^2}.$$

The bit complexity of Algorithm `Eliminate` is given in the following lemma.

Lemma 4.3.3. *Assume that $g \in \mathbb{Q}[x_1][x_2, \ldots, x_n]$ has degree d in $x_2, \ldots, x_n$ and bitsize τ_g, and that the polynomials $a_0, a_2, \ldots, a_n \in \mathbb{Q}[x_1]$ have bitsizes at most τ_a. Then,* `Eliminate` *runs in*

$$\widetilde{O}\left(n\tau_g + n^2 d\tau_a\right)$$

boolean operations and the bitsizes of the outputs $\phi_2, \ldots, \phi_n$ are in $\widetilde{O}\left(\tau_g + nd\tau_a\right)$.

Proof. Firstly, we estimate the bitsizes of ϕ_i, for $i = 2, \ldots, n$. From the definition of r_i in Line 4, one sees that $\mathrm{ht}(r_i) \leq \mathrm{ht}(r_{i+1}) + 2d\tau_a$. Since $\mathrm{ht}(r_{n+1}) = \tau_g$, the bitsize of r_i is bounded from above by $\tau_g + 2(n-1)d\tau_a$. The relation (4.3) leads to

$$\mathrm{ht}(\phi_i) \leq \mathrm{ht}(r_{i+1} - r_i) + \mathrm{ht}(x_i - \frac{a_i}{a_0}).$$

Because $\mathrm{ht}(r_{i+1} - r_i) \leq \max\{\mathrm{ht}(r_{i+1}), \mathrm{ht}(r_i)\}$ and $\mathrm{ht}(\frac{a_i}{a_0}) \leq 2\tau_a$, we get

$$\mathrm{ht}(\phi_i) \leq \tau_g + 2(nd - d + 1)\tau_a.$$

It follows that $\mathrm{ht}(\phi_i) = \widetilde{O}(\tau_g + nd\tau_a)$.

Clearly, the number of boolean operations to perform Lines 3 and 4 are $\widetilde{O}(\tau_g + nd\tau_a)$ and $O(1)$ respectively. The for loop in Line 2 has $n-1$ steps. Therefore, the number of boolean operations to perform the loop is $\widetilde{O}(n\tau_g + n^2 d\tau_a)$. This is also the number of boolean operations that Algorithm `Eliminate` uses. □

4.4 Computing zero-dimensional rational parametrizations

In our new algorithms (Algorithms 3 and 4) in Chapter 5, we need to compute a zero-dimensional rational parametrization of the algebraic variety associated to a zero-dimensional and radical ideal. The bit complexity of this procedure has been pointed out in

[77, Corollary 2]. In this section, we estimate the bit complexity of an algorithm (in the proof of Lemma 3.3.2) which computes a shape position from a zero-dimensional parametrization.

Let f be in $\mathbb{Q}[x]$ of degree d and bitsize τ. Assume that $V_{\text{grad}}(f)$ is finite. By applying [77, Corollary 2] to the system of partial derivatives of f, we obtain the following corollary which states that there exists an algorithm computing a zero-dimensional rational parametrization of $V_{\text{grad}}(f)$.

Corollary 4.4.1. *Assume that $V_{\text{grad}}(f)$ is finite. There exists a probabilistic algorithm that takes f as in input, and that produces one of the following outputs:*

a) *either a zero-dimensional rational parametrization of $V_{\text{grad}}(f)$,*

b) *a zero-dimensional rational parametrization of degree less than that of $V_{\text{grad}}(f)$,*

c) *or fails.*

In any case, the algorithm uses

$$\widetilde{O}\left(n^2(d+\tau)d^{2n+1}\binom{n+d}{d}\right) \tag{4.4}$$

boolean operations. Moreover, the polynomials $w, \kappa_1, \ldots, \kappa_n$ involved in the parametrization output have degrees at most $(d-1)^n$ and bitsize $\widetilde{O}\left((d+\tau+n)(d-1)^n\right)$.

Proof. Assume that the sequence of partial derivatives

$$\frac{\partial f}{\partial x_1}, \ldots, \frac{\partial f}{\partial x_n} \tag{4.5}$$

is given by a straight-line program Γ of size L, i.e., the program uses L elementary operations $+, -, \times$ to evaluate the sequence (4.5) from variables $x_1, \ldots, x_n$ and integers with bitsizes at most $\max_{i=1}^{n}\{\text{ht}\left(\frac{\partial f}{\partial x_i}\right)\}$.

We claim that L is $O(d\binom{n+d}{d})$. Indeed, f has at most $\binom{n+d}{d}$ terms and each term in f is defined by at most $d+1$ multiplications. Hence, the size of a straight-line program Γ_f which defines f does not exceed $(d+1)\binom{n+d}{d}$. By applying Baur-Strassen Theorem [10, Theorem 1], the size L is $O(d\binom{n+d}{d})$.

Recall that

$$\text{ht}\left(\frac{\partial f}{\partial x_i}\right) \leq \log d + \text{ht}(f) = \log d + \tau,$$

for $i = 1, \ldots, n$. By applying [77, Corollary 2] for the system 4.5 and a single group of variables, there exists an algorithm that takes the system as input and that produces

one of the outputs given as in items a)– c) of Corollary 4.4.1. The number of boolean operations of the algorithm is

$$\widetilde{O}\left(n^2d^{2n}(\log d+\tau+(d-1))(d\binom{n+d}{d}+n(d-1)+n^2)\right).$$

After reducing this formula, we get (4.4). Furthermore, the polynomials in the output have degrees at most $(d-1)^n$ and bitsize

$$\widetilde{O}\Big((d-1)^n(\log d+\tau+n+(d-1))\Big)=\widetilde{O}\Big((\tau+n+d)(d-1)^n\Big),$$

as claimed. □

Lemma 4.4.2. *Let $\mathcal{I}$ be a zero-dimensional and radical ideal, and $<$ be a lexicographic monomial order in $\mathbb{Q}[\boldsymbol{x}]$. To compute the reduced Gröbner basis $[w, x_2-v_2,\ldots,x_n-v_n]$ from the zero-dimensional rational parametrization $\mathcal{Q}=((w,\kappa_1,\ldots,\kappa_n),x_1)$ as in the proof of Lemma 3.3.2, we use*

$$\widetilde{O}\left((\tau+n+d)^2d^{6n}\right) \tag{4.6}$$

boolean operations. Moreover, the polynomials $w, v_2,\ldots,v_n$ have degrees at most $(d-1)^n$ and their maximum bitsizes are bounded from above by $\widetilde{O}((\tau+n+d)d^{3n})$.

Proof. By Corollary 4.4.1, the degree of w is at most $(d-1)^n$ and so $\deg w'$ is at most $(d-1)^n-1$. Assume that β is the positive minimum common denominator of all non-zero coefficients of w. Then, βw and $\beta w'$ belong to $\mathbb{Z}[t]$. Clearly,

$$\deg(\beta w')=\deg(\beta w)-1,\ \deg(\beta w)\le(d-1)^n,$$

and the bitsizes of βw and $\beta w'$ are bounded from above by $\widetilde{O}\left((\tau+n+d)(d-1)^n\right)$. We can apply [85, Theorem 6.52] to βw and $\beta w'$. The extended Euclidean algorithm computes the Bézout coefficient of $\beta w'$, denoted by b, using

$$\widetilde{O}(\tau+n+d)^2(d-1)^{6n}) \tag{4.7}$$

boolean operations. The bitsize of b is bounded by

$$O\left((\tau+n+d)(d-1)^{2n}\right). \tag{4.8}$$

Furthermore, one sees that the degree of b satisfies

$$\deg b\le\deg w-\deg\gcd(w,w')=\deg w\le(d-1)^n. \tag{4.9}$$

For every $i=2,\ldots,n$, we estimate the bitsize of the polynomial $b\kappa_i$. Recall from Corollary 4.4.1 that $\deg\kappa_i\le(d-1)^n$, hence from (4.9) one has $\deg b\kappa_i\le 2(d-1)^n$. From (4.8), we obtain

$$\mathrm{ht}(b\kappa_i)\le\mathrm{ht}(b)+\mathrm{ht}(\kappa_i)=\widetilde{O}((\tau+n+d)(d-1)^{2n})+\widetilde{O}((\tau+n+d)(d-1)^n).$$

After simplifying the last estimate, the bitsize of $b\kappa_i$ is bounded from above by $\widetilde{O}((\tau+n+d)(d-1)^{2n})$. Hence, the bitsize of $\eta b\kappa_i$, where η is the minimum common denominator of all non-zero coefficients of $b\kappa_i$, can be estimated as follows

$$\mathrm{ht}(\eta b\kappa_i) \leq 2\,\mathrm{ht}(b\kappa_i) \leq \widetilde{O}((\tau+n+d)(d-1)^{2n}).$$

In the proof of Lemma 3.3.2, we considered the division of $b\kappa_i$ by w and defined $v_i = b\kappa_i \mod w$. Thus, the degree of v_i is at most $\deg w \leq (d-1)^n$. From Lemma 4.3.1, the Euclidean division algorithm computes v_i using at most

$$\widetilde{O}((\tau+n+d)(d-1)^{5n}) \tag{4.10}$$

boolean operations. Thus, the bitsize of v_i is $\widetilde{O}((\tau+n+d)(d-1)^{3n})$, for $i = 2,\ldots,n$. Therefore, computing $[w, v_2, \ldots, v_n]$ from the zero-dimensional rational parametrization $\mathcal{Q}$ of $V_{\mathrm{grad}}(f)$, requires

$$\widetilde{O}((\tau+n+d)^2(d-1)^{6n} + (n-1)(\tau+n+d)(d-1)^{5n})$$

boolean operations, as a consequence of (4.7) and (4.10). By applying further simplifications, we obtain the desired result (4.6). □

4.5 Solving semi-definite programs

Our two algorithms (Algorithms 6 and 7) in Chapter 6 are based on complex semi-definite programming. Thus, we need to recall some notions related to semi-definite programming together complexity considerations.

4.5.1 Background on semi-definite matrices

One says that $Q \in \mathbb{C}^{n\times n}$ is a *Hermitian matrix* if Q is equal to its own conjugate transpose $\bar{Q}^T$, also denoted by $Q^\star$. That is, if its entries satisfy $q_{ij} = \bar{q}_{ji}$ for all $1 \leq i, j \leq n$. Clearly, all entries lying on the diagonal of a Hermitian matrix are real numbers. We denote by $\mathcal{H}^n$ the set of all Hermitian $n \times n$-matrices. We denote by $\mathcal{S}^n$ the set of all real symmetric $n \times n$-matrices.

A matrix $Q \in \mathcal{H}^n$ is said to be *positive semi-definite* (resp. definite) if Q has only non-negative (resp. positive) eigenvalues, and in this case we use the notation $Q \succeq 0$ (resp. $Q \succ 0$). The *Cholesky decomposition* of a Hermitian positive-definite matrix Q is the product $LL^\star$, where L is a lower triangular matrix with real and positive diagonal entries, and $L^\star$ is the conjugate transpose of the matrix L. A related variant of the Cholesky decomposition is the *LDL decomposition*, $Q = LDL^\star$, where L is a lower

unit triangular matrix, i.e. the diagonal elements of L are required to be 1, and D is a diagonal matrix.

The following result is obtained by applying directly the argument in the proofs of [4, Lemma 2.1 & Theorem 3.2] in the complex case, and will be used to investigate the bit complexity of Algorithms `csos2` and `csos3` based on SDP solving in Chapter 6.

Lemma 4.5.1 ([4]). *Let $Q \in \mathcal{H}^n$ be positive definite with Gaussian entries. Assume that L is the factor of Q computed by Cholesky's decomposition with finite precision δ_c. Then, we have $LL^\star = Q + H$, where*

$$|H_{ij}| \leq \frac{(n+1)2^{-\delta_c}\sqrt{|Q_{ii}Q_{jj}|}}{1-(n+1)2^{-\delta_c}}. \tag{4.11}$$

In addition, if the smallest eigenvalue $\tilde{\lambda}$ of Q satisfies the inequality

$$2^{-\delta_c} < \frac{\tilde{\lambda}}{n^2+n+(n-1)\tilde{\lambda}}\,, \tag{4.12}$$

Cholesky's decomposition returns a nonsingular factor L with Gaussian entries.

4.5.2 Bit complexity of solving semi-definite programs

A *complex semi-definite program* (SDP for short) is defined as the following optimization problem:

$$\begin{array}{ll} \text{minimize} & \operatorname{tr}(CX) \\ \text{subject to} & \operatorname{tr}(A_i X) = b_i,\ i = 1,\ldots,m, \\ & X \succeq 0 \end{array}$$

where $X \in \mathcal{H}^n$ is the matrix variable, the matrices $C, A_i \in \mathcal{H}^n$ and $b_i \in \mathbb{R}^n$ are given data and $\operatorname{tr}(\cdot)$ stands for the usual matrix trace operator. Consequently, the objective function and the constraints are convex.

The previous problem becomes a *real* SDP if the data is real, i.e., the matrices X, C and A_i are real symmetric matrices. When the SDP problem is given by rational data, we use the bit complexity analysis of the ellipsoid method by Khachiyan and Porkolab [69].

Theorem 4.5.2 ([69]). *We consider the real semi-definite feasibility problem*

$$\operatorname{tr}(A_i X) \leq b_i,\ i = 1,\ldots,m,\ X \succeq 0,$$

where $A_i \in \mathcal{S}^n$ with rational entries, $b_i \in \mathbb{Q}^n$ are given and $X \in \mathcal{S}^n$ is the variable. Assume that the maximal bitsize of their entries is τ, the accuracy δ and the radius bound R of computation are given. Then, to compute an approximate solution within the accuracy δ of this SDP, we need to perform $O(n^4 \log_2(2^\tau n\, R\, 2^\delta))$ iterations of the ellipsoid method, where each iteration requires $O(n^2(m+n))$ arithmetic operations over $\log_2(2^\tau n\, R\, 2^\delta)$-bit numbers.

4.6 Other estimates

This section provides a lower bound for a complex univariate polynomial with Gaussian integers on the unit circle. This lower bound allows us to estimate bit complexities of the two algorithms `csos1` and `csos2` in Chapter 6.

4.6.1 Distance between the roots of a complex univariate polynomial

For a polynomial $f = f_0 + \cdots + f_d z^d \in \mathbb{C}[z]$ of degree d, the *minimal distance* between the roots $\alpha_1, \ldots, \alpha_d$ of f is defined by

$$\operatorname{sep}(f) := \min\{|\alpha_i - \alpha_j|,\ \alpha_i \neq \alpha_j\}.$$

The *norm* of f is defined as $\|f\| := |f_d| + \cdots + |f_0|$. The following lemma is an immediate consequence of [57, Theorem 2].

Lemma 4.6.1. *Let $f \in \mathbb{Z}[\boldsymbol{i}][z]$ of degree d and τ be the maximum bitsize of its coefficients. Assume that f has no multiple root. The minimal distance between the roots of f satisfies*

$$\operatorname{sep}(f) \geq \frac{\sqrt{3}}{d^{\frac{d}{2}+1}\|f\|^{d-1}}.$$

Therefore, one needs an accuracy of $\delta = \widetilde{O}(\tau d)$ to compute distinct approximations of the roots of f with complex root isolation.

Proof. By [57, Theorem 2], one has:

$$\operatorname{sep}(f) \geq \frac{\sqrt{3|\operatorname{Disc}(f)|}}{d^{\frac{d}{2}+1}\|f\|^{d-1}}, \tag{4.13}$$

where

$$\operatorname{Disc}(f) = f_d^{2d-2} \prod_{j<k} (\alpha_j - \alpha_k)^2$$

is the discriminant of f. Note that $\operatorname{Disc}(f)$ can be written as a polynomial in $f_0, \ldots, f_d$ with integer coefficients, thus $\operatorname{Disc}(f) \in \mathbb{Z}[\boldsymbol{i}]$ and one has $|\operatorname{Disc}(f)| \geq 1$ which from (4.13), implies

$$\operatorname{sep}(f) \geq \frac{\sqrt{3|\operatorname{Disc}(f)|}}{d^{\frac{d}{2}+1}\|f\|^{d-1}} \geq \frac{\sqrt{3}}{d^{\frac{d}{2}+1}\|f\|^{d-1}},$$

the desired inequality. □

4.6.2 The minimum of a real bivariate polynomial on the unit circle

The following lemma provides a lower bound on the minimum of a real bivariate polynomial over the unit circle in $\mathbb{R}^2$.

Lemma 4.6.2. *Let $p \in \mathbb{Z}[x,y]$ be a real bivariate polynomial of degree d and τ be the maximum bitsize of its coefficients. Assume that p is positive on the unit circle $\mathscr{C}$. Then, the minimum of p on $\mathscr{C}$ satisfies the following inequality:*

$$p_{\min} := \min\{p(x,y) : x^2+y^2=1\} \geq 2^{-\widetilde{O}(d^3(d+\tau))}.$$

Proof. We consider the following algebraic set:

$$V := \left\{(x,y,m) \in \mathbb{C}^3 : p(x,y) - m = \; y\frac{\partial p}{\partial x} - x\frac{\partial p}{\partial y} = 0, x^2+y^2=1\right\}.$$

Note that the projection of V on the m-axis defines the critical values of the restriction of the evaluation map $z \mapsto p(z)$ to $\mathscr{C}$ which contains $p_{\min}$.

Assume that V is finite. By [77, Corollary 2], there is a zero-dimensional parametrization of V defined by univariate polynomials with bitsizes upper bounded by $\widetilde{O}(d^3(d+\tau))$. Since there exists (x_0,y_0) on $\mathscr{C}$ such that $(x_0,y_0,p_{\min})$ belongs to V, $p_{\min}$ is a (non-zero) root of a univariate polynomial of degree at most $O(d^3\tau)$. Hence, the Cauchy bound [17] yields:

$$|p_{\min}| \geq 2^{-\widetilde{O}(d^3(d+\tau))}.$$

Assume now that V is not finite. By Krull's theorem [42], this implies that $\mathscr{C}$ is contained in the complex zero set defined by

$$y\frac{\partial p}{\partial x} - x\frac{\partial p}{\partial y} = 0,$$

whence is a factor of this polynomial. This implies that there exists a factorization $p = p_1p_2$ where p_1 is a power of x^2+y^2-c (where c is a constant) and the zero set of the polynomial

$$y\frac{\partial p_2}{\partial x} - x\frac{\partial p_2}{\partial y}$$

has a zero-dimensional intersection with $\mathscr{C}$. This yields the following analysis. The set V is the union of a 1-dimensional component containing points (m,x,y) where (x,y) ranges over $\mathscr{C}$ and $m = c-1$, and a 0-dimensional component containing points (m,x,y) which are solutions to

$$p(x,y) = m,\; y\frac{\partial p_2}{\partial x} - x\frac{\partial p_2}{\partial y} = 0,\; x^2+y^2=1.$$

Applying the argument in the second paragraph of the proof to the above system ends the proof. □

PART II

CONTRIBUTIONS

CHAPTER 5

Exact SOS decompositions over gradient ideals with rational coefficients

Contents

Here, we consider the problem of computing exact certificates for non-negativity of real multivariate polynomials. We build on previous works by Parrilo, Nie, Demmel and Sturmfels who introduced certificates of non-negativity modulo gradient ideals [66, 62]. We prove that if the polynomial under consideration has rational coefficients then such certificates can be obtained exactly over the rationals and we provide exact algorithms to compute them. We analyze the bit complexity of these algorithms and deduce bitsize bounds of such certificates.

This chapter contains three sections. In Section 5.1, we prove the existence of an SOS of *polynomials* modulo the gradient ideal of f, we introduce Algorithm `sosgradientshape` and analyze its bit complexity. Our results towards decomposing f as an SOS of *rational fractions* modulo the gradient ideal along with Algorithm `sosgradient` are presented in Section 5.2. Practical experiments are given in the last Section 5.3.

Most of the content of this chapter is from the paper [55] entitled "*Sum of squares decompositions of polynomials over their gradient ideals with rational coefficients*" by Victor Magron, Mohab Safey El Din, and Trung Hieu Vu.

5.1 SOS of polynomials modulo gradient ideals

We recall that f is an SOS of polynomials over the quotient ring $\mathbb{Q}[\boldsymbol{x}]/\mathcal{I}_{\text{grad}}(f)$ if there exists $g \in \mathcal{I}_{\text{grad}}(f)$ such that $f - g$ is SOS in $\mathbb{Q}[\boldsymbol{x}]$, i.e., f can be decomposed

$$f = \sum_{j=1}^{s} c_j q_j^2 + \sum_{i=1}^{n} \phi_i \frac{\partial f}{\partial x_i},$$

for some polynomials $q_1, \ldots, q_s, \phi_1, \ldots, \phi_s$ in $\mathbb{Q}[\boldsymbol{x}]$ and positive numbers $c_1, \ldots, c_s$ in $\mathbb{Q}$.

5.1.1 The existence of an SOS decomposition over the rationals

The main result of this subsection is stated below. This result provides a necessary and sufficient condition for the non-negativity of $f \in \mathbb{Q}[\boldsymbol{x}]$ under a generic condition. Its proof is constructive, in that we show explicitly how to compute an SOS decomposition over gradient ideals with rational coefficients.

Theorem 5.1.1. *Let $f \in \mathbb{Q}[\boldsymbol{x}]$ be such that the following conditions hold:*

a) *The infimum $f_{\text{inf}} = \inf\{f(x) : x \in \mathbb{R}^n\}$ is attained.*

b) *The gradient ideal $\mathcal{I}_{\text{grad}}(f)$ is zero-dimensional and radical.*

Then, f is non-negative over $\mathbb{R}^n$ if and only if f is an SOS of polynomials over the quotient ring $\mathbb{Q}[\boldsymbol{x}]/\mathcal{I}_{\text{grad}}(f)$.

Proof. Suppose that f is non-negative over $\mathbb{R}^n$ and $\#V_{\text{grad}}(f) = \delta$. We prove that f is an SOS of polynomials over the quotient ring $\mathbb{Q}[\boldsymbol{x}]/\mathcal{I}_{\text{grad}}(f)$. We consider the two following cases:

CASE 1. Distinct points in $V_{\text{grad}}(f)$ have distinct x_1-coordinates. We now consider the lexicographic monomial order $x_1 < x_2 < \cdots < x_n$ on $\mathbb{Q}[\boldsymbol{x}]$. Since the gradient ideal is zero-dimensional and radical, according to the Shape Lemma (Lemma 3.3.1), the reduced Gröbner basis of $\mathcal{I}_{\text{grad}}(f)$ has the following form:

$$[w, x_2 - v_2, \ldots, x_n - v_n], \tag{5.1}$$

where $v_2, \ldots, v_n$ are polynomials in $\mathbb{Q}[x_1]$ of degree at most $\delta - 1$. We denote

$$h(x_1) := f(x_1, v_2, \ldots, v_n), \tag{5.2}$$

where x_i is replaced by v_i in f for $i = 2, \ldots, n$. With the order $<$, we divide $f - h$ by the system in (5.1) using the division algorithm in [20, Ch. 2, Sec 3.]. Then, there exist

$\phi_1, \ldots, \phi_n$ in $\mathbb{Q}[\boldsymbol{x}]$, and r in $\mathbb{Q}[x_1]$ such that

$$f - h = \phi_1 w + \sum_{i=2}^{n} \phi_i (x_i - v_i) + r, \tag{5.3}$$

with $\deg r < \delta$. Let x be in $V_{\text{grad}}(f)$. From (5.2) and (5.3), one sees that $f(x) = h(x)$. Hence, $f - h$ vanishes on $V_{\text{grad}}(f)$. Clearly, the value of $\phi_1 w + \sum_{i=2}^{n} \phi_i (x_i - v_i)$ is zero on $V_{\text{grad}}(f)$. This implies that r also vanishes on the image set $\pi(V_{\text{grad}}(f))$, where $\pi(x_1, \ldots, x_n) = x_1$. Since distinct points in $V_{\text{grad}}(f)$ have distinct x_1-coordinates, it holds that

$$\#\pi(V_{\text{grad}}(f)) = \#V_{\text{grad}}(f) = \delta.$$

As $\deg r < \delta$, we conclude that $r \equiv 0$. Hence, from (5.3), we obtain the following representation:

$$f = h + \phi_1 w + \sum_{i=2}^{n} \phi_i (x_i - v_i). \tag{5.4}$$

The set $\{(x_1, x_2, \ldots, x_n) \in \mathbb{R}^n : x_2 = v_2, \ldots, x_n = v_n\}$ defines a curve which is parametrized by x_1. Recall that f is non-negative over $\mathbb{R}^n$. Hence f is non-negative over this curve. Since f takes the same values over this curve as h takes over x_1 when x_1 ranges in $\mathbb{R}$, one can conclude that the univariate polynomial h is also non-negative over $\mathbb{R}$. According to the results on SOS decompositions of univariate polynomials with rational coefficients in Theorem 4.2.1, h is a sum of s squares in $\mathbb{Q}[x_1]$, i.e., there exist $q_1, \ldots, q_s \in \mathbb{Q}[x_1]$ and $c_1, \ldots, c_s$ in $\mathbb{Q}_+$ such that $h = c_1 q_1^2 + \cdots + c_s q_s^2$. Therefore, from (5.4), we assert that f is an SOS of polynomials over $\mathbb{Q}[\boldsymbol{x}]/\mathcal{I}_{\text{grad}}(f)$.

CASE 2. There are two distinct points in $V_{\text{grad}}(f)$ such that their x_1-coordinates are equal. According to [75, Lemma 2.1], there is $j \in \{1, \ldots, (n-1)\delta(\delta-1)/2\}$ such that the linear function

$$u := x_1 + j x_2 + \cdots + j^{n-1} x_n$$

separates $V_{\text{grad}}(f)$, i.e., $u(x) \neq u(y)$ for any distinct points x, y in $V_{\text{grad}}(f)$. We consider the change of variables $\boldsymbol{y} = T\boldsymbol{x}$, where

$$T = \begin{bmatrix} 1 & j & j^2 & \cdots & j^{n-1} \\ 0 & 1 & 0 & \cdots & 0 \\ 0 & 0 & 1 & \cdots & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & 0 & \cdots & 1 \end{bmatrix}. \tag{5.5}$$

We note that T is an invertible matrix. Therefore, we can obtain a polynomial $g(\boldsymbol{y}) = f(T^{-1}\boldsymbol{y}) \in \mathbb{Q}[\boldsymbol{y}]$ in new variables $y_1, y_2, \ldots, y_n$ such that g is non-negative and the infimum $g_{\text{inf}} = \inf\{g(y) : y \in \mathbb{R}^n\}$ is attained. By the chain rule $\nabla g = \nabla f \circ T^{-1}$, we have

$$V_{\text{grad}}(g) = \{y \in \mathbb{C}^n : y = Tx, x \in V_{\text{grad}}(f)\}.$$

Thus, the gradient ideal $\mathcal{I}_{\text{grad}}(g)$ is zero-dimensional and radical. Since $y_1 = u(\boldsymbol{x})$ separates $V_{\text{grad}}(f)$, distinct points in $V_{\text{grad}}(g)$ have distinct y_1-coordinates.

We now apply Case 1 for g. There exists an SOS decomposition of g modulo $\mathcal{I}_{\text{grad}}(g)$

$$g(\boldsymbol{y}) = \sum_{j=1}^{s} c_j \bar{q}_j^2(\boldsymbol{y}) + \sum_{i=1}^{n} \bar{\phi}_i(\boldsymbol{y}) \frac{\partial g}{\partial y_i}, \tag{5.6}$$

where $\bar{q}_1, \ldots, \bar{q}_s, \bar{\phi}_1, \ldots, \bar{\phi}_n \in \mathbb{Q}[\boldsymbol{y}]$ and $c_1, \ldots, c_s \in \mathbb{Q}_+$. By replacing $\boldsymbol{y}$ by $T\boldsymbol{x}$ and $\frac{\partial g}{\partial y_i}$ by $\frac{\partial f}{\partial x_i} \circ T^{-1}$ in (5.6), we obtain a decomposition of f as follows:

$$f(\boldsymbol{x}) = g(T\boldsymbol{x}) = \sum_{j=1}^{s} c_j \bar{q}_j^2(T\boldsymbol{x}) + \sum_{i=1}^{n} \bar{\phi}_i(T\boldsymbol{x}) \frac{\partial f}{\partial x_i} \circ T^{-1}. \tag{5.7}$$

Since $(\frac{\partial f}{\partial x_i} \circ T^{-1})(Tx) = \frac{\partial f}{\partial x_i}(x)$, (5.7) is an SOS decomposition of f modulo $\mathcal{I}_{\text{grad}}(f)$.

To complete the proof, we need to prove the reverse conclusion. Suppose that f is SOS over the quotient ring $\mathbb{Q}[\boldsymbol{x}]/\mathcal{I}_{\text{grad}}(f)$, i.e., f can be decomposed as follows:

$$f = \sum_{j=1}^{s} c_j q_j^2 + \sum_{i=1}^{n} \phi_i \frac{\partial f}{\partial x_i}, \tag{5.8}$$

for some polynomials $q_1, \ldots, q_s, \phi_1, \ldots, \phi_n \in \mathbb{Q}[\boldsymbol{x}]$, and $c_1, \ldots, c_s$ in $\mathbb{Q}_+$. Let $x_{\text{inf}} \in \mathbb{R}^n$ be such that $f(x_{\text{inf}}) = f_{\text{inf}}$. Then, x_{inf} is a critical point of f over $\mathbb{R}^n$, i.e., x_{inf} belongs to the variety $V_{\text{grad}}(f)$. Thus, we have

$$\sum_{i=1}^{n} \phi_i(x_{\text{inf}}) \frac{\partial f}{\partial x_i}(x_{\text{inf}}) = 0.$$

From (5.8), we see that $f(x_{\text{inf}}) = \sum_{j=1}^{s} c_j q_j^2(x_{\text{inf}})$ and so this value is non-negative. By assumption, for all x in $\mathbb{R}^n$, $f(x) \geq f(x_{\text{inf}})$. Hence, f is non-negative over $\mathbb{R}^n$. □

Remark 5.1.1. Assume that Q is a real field and that R is the real closure of Q. All arguments in the proof of Theorem 5.1.1 can be applied for f in $\mathrm{Q}[\boldsymbol{x}]$. Hence, the conclusion of Theorem 5.1.1 holds for the case $\mathrm{Q}[\boldsymbol{x}]$, i.e., f is non-negative over R^n if and only if f is an SOS of polynomials over the quotient ring $\mathrm{Q}[\boldsymbol{x}]/\mathcal{I}_{\text{grad}}(f)$ provided that the infimum $f_{\text{inf}} = \inf\{f(x) : x \in \mathrm{R}^n\}$ is attained and that the gradient ideal $\mathcal{I}_{\text{grad}}(f)$ is zero-dimensional and radical.

Remark 5.1.2. In the proof of Theorem 5.1.1, one can see that $f - h$ vanishes not only on $V_{\text{grad}}(f)$ but also on the variety defined by $\langle x_2 - v_2, \ldots, x_n - v_n \rangle$. Hence, ϕ_1 in (5.4) is zero and (5.4) becomes $f = c_1 q_1^2 + \cdots + c_s q_s^2 + \sum_{i=2}^{n} \phi_i(x_i - v_i)$.

Remark 5.1.3. Note that if f does not attain its infimum, it could be SOS modulo the gradient ideal but fail to be non-negative, as it may be negative at points where the

gradient does not vanish. This is illustrated by the example

$$f = x^2 + (xy - 1)^2 - \frac{1}{2}$$

whose gradient ideal is generated by x, y. Hence, f is $\frac{1}{2}$ modulo its gradient ideal while it can have negative values (e.g. along the sequence of points $\left(\frac{1}{k}, k\right)$ for $k \geq 1$). Hence, condition a) in Theorem 5.1.1 is used only to prove the reverse conclusion. Therefore, even without this condition, the following assertion still holds: Assume that $\mathcal{I}_{\text{grad}}(f)$ is zero-dimensional and radical. If f is non-negative over $\mathbb{R}^n$, then f is SOS modulo $\mathcal{I}_{\text{grad}}(f)$.

Theorem 5.1.1 provides certificates of non-negativity for polynomials in $\mathbb{Q}[\boldsymbol{x}]$ which satisfy its assumptions and which are not SOS of polynomials with real (or rational) coefficients. We illustrate this with two examples.

Example 5.1.1. We consider the Robinson polynomial [74],

$$p_R = x_1^6 + x_2^6 + x_3^6 - x_1^4x_2^2 - x_1^4x_3^2 - x_2^4x_1^2 - x_2^4x_3^2 - x_3^4x_1^2 - x_3^4x_2^2 + 3x_1^2x_2^2x_3^2,$$

that is non-negative but cannot be represented as an SOS of polynomials. By substituting the third variable x_3 by 1 in p_R, we get the following non-negative polynomial:

$$f_R = x_1^6 + x_2^6 - x_1^4x_2^2 + 3x_1^2x_2^2 - x_1^2x_2^4 - x_1^4 - x_2^4 - x_1^2 - x_2^2 + 1.$$

Because p_R is the homogenization of f_R, f_R cannot be represented as an SOS of polynomials [58, Proposition 1.2.4]. The gradient ideal $\mathcal{I}_{\text{grad}}(f_R)$ is zero-dimensional and radical. Thus, Theorem 5.1.1 tells us that f_R is an SOS of polynomials modulo $\mathcal{I}_{\text{grad}}(f_R)$.

Example 5.1.2. We consider the Scheiderer polynomial given in (2.4) that can be decomposed as an SOS of polynomials with algebraic coefficients but cannot be decomposed as an SOS of polynomials with rational coefficients. By replacing the third variable x_3 by -1, we obtain the non-negative polynomial

$$f_S = x_1^4 + x_1x_2^3 + x_2^4 + 3x_1^2x_2 + 4x_1x_2^2 + 2x_1^2 - x_1 - x_2 + 1.$$

Note that the conclusion in [58, Proposition 1.2.4] holds for polynomials with rational coefficients, i.e., $g \in \mathbb{Q}[\boldsymbol{x}]$ is SOS in $\mathbb{Q}[\boldsymbol{x}]$ if and only if its homogenization is in $\mathbb{Q}[\boldsymbol{x}]$. Hence, the polynomial f_S is also SOS with algebraic coefficients but not SOS with rational ones. The gradient ideal $\mathcal{I}_{\text{grad}}(f_S)$ satisfies the zero-dimensional and radical condition. Hence, according to Theorem 5.1.1, f_S is an SOS of polynomials over the quotient ring $\mathbb{Q}[\boldsymbol{x}]/\mathcal{I}_{\text{grad}}(f_S)$.

An explicit SOS decomposition of f_S will be given later in Example 5.2.2.

Example 5.1.3. The gradient ideal of the Motzkin polynomial p_M given in (2.2) is neither radical nor zero-dimensional. We consider the positive polynomial $g = x_1^2 + x_2^2$ and see that the gradient ideal of the product polynomial $f_M := gp_M$ is radical and zero-dimensional. According to Theorem 5.1.1, f_M is an SOS of polynomials modulo the ideal $\mathcal{I}_{\text{grad}}(f_M)$. This provides indirectly a certificate of non-negativity of the Motzkin polynomial.

5.1.2 Description of the algorithm

The proof of Theorem 5.1.1 allows us to design an algorithm to compute a rational SOS decomposition of polynomials modulo the gradient ideal of a non-negative polynomial.

The input of `sosgradientshape` is a non-negative polynomial $f \in \mathbb{Q}[\boldsymbol{x}]$ whose gradient ideal $\mathcal{I}_{\text{grad}}(f)$ is zero-dimensional, radical, and satisfies Shape Lemma's assumption, i.e., all points in $V_{\text{grad}}(f)$ have distinct x_1-coordinates. Our software implementation first checks that the gradient ideal is zero-dimensional and radical, and returns an error if the assumption is not satisfied. To do so, we rely on the procedures `IsZeroDimensional` and `IsRadical` from the Maple package `PolynomialIdeals`. These are all based on Gröbner bases computations (see e.g. [20]).

The output includes the cardinality $\delta = \#V_{\text{grad}}(f)$, the lists of polynomials and numbers

$$[w, v_2, \ldots, v_n],\ [q_1, \ldots, q_s],\ [\phi_2, \ldots, \phi_n] \subset \mathbb{Q}[\boldsymbol{x}],\ \text{and}\ [c_1, \ldots, c_s] \subset \mathbb{Q}_+$$

satisfying the relation

$$f = \sum_{j=1}^{s} c_j q_j^2 + \sum_{i=2}^{n} \phi_i (x_i - v_i)$$

From Remark 5.1.2, we do not need to compute ϕ_1 because it always equals zero. In Line 1, we compute the reduced Gröbner basis G for $\mathcal{I}_{\text{grad}}(f)$ by relying on the zero-dimensional rational parametrization of $V_{\text{grad}}(f)$ mentioned in Lemma 3.3.2. In Line 2, we compute the quotients $\phi_2, \ldots, \phi_n$ and the remainder r of the division of f by G. In Line 3, we compute a rational weighted SOS decomposition of the non-negative univariate polynomial h by using Algorithm `univsos1` or Algorithm `univsos2` described in [53, Fig. 1] or [53, Fig. 2], respectively.

Remark 5.1.4. Suppose that the Shape Lemma assumption does not hold for $\mathcal{I}_{\text{grad}}(f)$, i.e., there are two distinct points in $V_{\text{grad}}(f)$ such that their x_1-coordinates are equal. As mentioned in the proof of Theorem 5.1.1, we can find an invertible matrix T given by (5.5), make a change of variables $\boldsymbol{y} = T\boldsymbol{x}$ and assign $g(\boldsymbol{y}) := f(T^{-1}\boldsymbol{y})$. Here, we have

$$y_1 = x_1 + jx_2 + \cdots + j^{n-1}x_n$$

Algorithm 3 Computing SOS of polynomials modulo the gradient ideal

`sosgradientshape` := proc(f)

Input: $f \in \mathbb{Q}[\boldsymbol{x}]$ non-negative over $\mathbb{R}^n$ such that $\mathcal{I}_{\text{grad}}(f)$ is zero-dimensional and radical and all points in $V_{\text{grad}}(f)$ have distinct x_1-coordinates

Output: δ in $\mathbb{N}$, $[q_1,\ldots,q_s]$, $[w,v_2,\ldots,v_n] \subset \mathbb{Q}[x_1]$, $[\phi_2,\ldots,\phi_n] \subset \mathbb{Q}[\boldsymbol{x}]$, $[c_1,\ldots,c_s] \subset \mathbb{Q}_+$ satisfying

$$f = \sum_{j=1}^{s} c_j q_j^2 + \sum_{i=2}^{n} \phi_i(x_i - v_i). \tag{5.9}$$

1: Compute the reduced Gröbner basis $G = [w, x_2 - v_2, \ldots, x_n - v_n]$ of $\mathcal{I}_{\text{grad}}(f)$, with the lexicographical ordering $x_1 < x_2 < \cdots < x_n$, and $\delta = \deg w$

2: Compute the quotients $[\phi_2,\ldots,\phi_n]$ and remainder h of the division of f by G by performing `Eliminate`$(f, 1, v_2, \ldots, v_n)$

3: Compute a rational weighted SOS decomposition $h = c_1 q_1^2 + \cdots + c_s q_s^2$

4: **return** δ, $[q_1,\ldots,q_s]$, $[\phi_2,\ldots,\phi_n]$, $[w,v_2,\ldots,v_n]$, and $[c_1,\ldots,c_s]$

for some $j > 0$ and $y_i = x_i$ for $i = 2,...,n$. We get a new non-negative polynomial in n new variables with rational coefficients $g(\boldsymbol{y})$ whose gradient ideal satisfies the Shape Lemma assumption. Now, we can apply `sosgradientshape` for $g(\boldsymbol{y})$ and obtain as output the number $\bar{\delta}$, two lists $[\bar{q}_1,\ldots,\bar{q}_s]$, $[\bar{w},\bar{v}_2,\ldots,\bar{v}_n]$ of polynomials in $\mathbb{Q}[y_1]$, a list $[\bar{\phi}_1,\ldots,\bar{\phi}_n]$ of polynomials in $\mathbb{Q}[\boldsymbol{y}]$, and a list $[c_1,\ldots,c_s] \subset \mathbb{Q}_+$. Since $\#V_{\text{grad}}(f) = \#V_{\text{grad}}(g)$, one has $\bar{\delta} = \delta$. The new polynomial g can be decomposed as follows:

$$g(\boldsymbol{y}) = \sum_{j=1}^{s} c_j \bar{q}_j^2(y_1) + \bar{\phi}_1(\boldsymbol{y})\bar{w}(y_1) + \sum_{i=2}^{n} \bar{\phi}_i(\boldsymbol{y})(y_i - \bar{v}_i(y_1)).$$

Hence, f can be decomposed as:

$$f(\boldsymbol{x}) = \sum_{j=1}^{s} c_j \bar{q}_j^2(u(\boldsymbol{x})) + \bar{\phi}_1(T\boldsymbol{x})\bar{w}(u(\boldsymbol{x})) + \sum_{i=2}^{n} \bar{\phi}_i(T\boldsymbol{x})(x_i - \bar{v}_i(u(\boldsymbol{x}))), \tag{5.10}$$

where

$$u(\boldsymbol{x}) = x_1 + jx_2 + \cdots + j^{n-1}x_n.$$

Clearly, $[w(u), x_2 - \bar{v}_2(u), \ldots, x_n - \bar{v}_n(u)]$ is also a basis for $V_{\text{grad}}(f)$. Hence, (5.10) provides us an SOS decomposition of f modulo the gradient ideal of f.

To illustrate how the algorithm works, we consider the following simple example.

Example 5.1.4. Consider the polynomial $f_E = 2x_1^2 + 4x_1x_2 + x_2^4 + 3$. This polynomial is non-negative over $\mathbb{R}^n$. Firstly, the gradient ideal $\mathcal{I}_{\text{grad}}(f_E)$ is given by

$$\mathcal{I}_{\text{grad}}(f_E) = \langle 2x_1 + 2x_2, 4x_1 + 4x_2^3 \rangle$$

which is zero-dimensional and radical. We compute the reduced Gröbner basis of $\mathcal{I}_{\text{grad}}(f_E)$, namely $[x_1^3 - x_1, x_2 + x_1]$, here $v_2(x_1) = -x_1$,

$$\delta = \deg(x_1^3 - x_1) = 3 = \#V_{\text{grad}}(f_E).$$

Secondly, with the order $x_1 < x_2$, the quotients of the division of f by the Gröbner basis are $\phi_1 = 0$ and $\phi_2 = -x_1^3 + x_1^2x_2 - x_1x_2^2 + x_2^3 + 4x_1$, and the remainder is given by

$$h(x_1) = f_E(x_1, v_2) = x_1^4 - 2x_1^2 + 3.$$

Thirdly, one gets an SOS decomposition $h = (x_1^2 - 1)^2 + 2$. Finally, we obtain the following SOS decomposition of f_E modulo its gradient ideal:

$$f_E = (x_1^2 - 1)^2 + 2 + (-x_1^3 + x_1^2x_2 - x_1x_2^2 + x_2^3 + 4x_1) \times (x_1 + x_2).$$

Theorem 5.1.2. *Let f be a non-negative polynomial in $\mathbb{Q}[\boldsymbol{x}]$. Suppose that $\mathcal{I}_{\text{grad}}(f)$ is zero-dimensional and radical, and all points in $V_{\text{grad}}(f)$ have distinct x_1-coordinates. On input f,* `sosgradientshape` *terminates and computes an SOS decomposition of f modulo $\mathcal{I}_{\text{grad}}(f)$ with rational coefficients.*

Proof. Assume that $f \in \mathbb{Q}[\boldsymbol{x}]$ is non-negative over $\mathbb{R}^n$ and its gradient ideal is zero-dimensional and radical. Here, we use the lexicographic monomial order $x_1 < x_2 < \cdots < x_n$. Because the Shape Lemma assumption holds, the reduced Gröbner basis of $\mathcal{I}_{\text{grad}}(f)$ in Line 1 has the form $G = [w, x_2 - v_2, \ldots, x_n - v_n]$, and can be computed by using a zero-dimensional rational parametrization of $V_{\text{grad}}(f)$ as in Lemma 3.3.2. In Line 2, we compute the quotients $[\phi_2, \ldots, \phi_n]$ and the remainder r of the division of f by G by performing $\texttt{Eliminate}(f, 1, v_2, \ldots, v_n)$ (as in Algorithm 2). Here, we see that r coincides with h, where $h = f(x_1, v_2, \ldots, x_n)$ as in the proof of Theorem 5.1.1, because of

$$r = f - \sum_{i=2}^{n} \phi_i(x_i - v_i) = h.$$

In Line 3, the univariate polynomial h is non-negative with rational coefficients. Thus, by using `univsos1` or `univsos2` [53], we can compute an SOS decomposition of h, $h = c_1q_1^2 + \cdots + c_sq_s^2$. Hence, according to the proof of Theorem 5.1.1, we get (5.9) which is an SOS decomposition modulo the gradient ideal of f. □

5.1.3 Bit complexity analysis

This subsection investigates the bit complexity of Algorithm `sosgradientshape`. Assume that d and τ are the degree and an upper bound of the bitsize of the coefficients of $f \in \mathbb{Q}[\boldsymbol{x}]$ respectively. We provide estimates for the bitsizes of polynomials in

the output of `sosgradientshape`(f) as well as for the number of boolean operations required to execute it.

To analyze the bit complexity of Algorithm 3, we need to estimate bit complexities of all steps in Lines 1–3. We first analyze bit complexities for computing SOS decompositions of the non-negative univariate polynomial h in Line 3.

Proposition 5.1.3. *Let $v_2, \ldots, v_n$ be as in Lemma 3.3.2 and $h(x_1) = f(x_1, v_2, \ldots, v_n)$. To compute an SOS decomposition of h, Algorithm* `univsos1` *and Algorithm* `univsos2` *run in*

$$\widetilde{O}\Big((d^{n+1}/2)^{3d^{n+1}/2}(\tau + n + d)d^{3n+1}\Big) \tag{5.11}$$

and

$$\widetilde{O}\left((\tau + n + d)d^{6n+4}\right) \tag{5.12}$$

boolean operations, respectively.

Proof. Let $\tau_v = \max_i\{\mathrm{ht}(v_i)\}$. Lemma 3.3.2 tells us that the bitsize of τ_v is bounded from above by $\widetilde{O}\left((\tau + n + d)d^{3n}\right)$, and that the polynomials $w, v_2, \ldots, v_n$ have degree at most $(d-1)^n$. Since $\deg f = d$ and $h(x_1) = f(x_1, v_2, \ldots, v_n)$, the degree of h is at most $d(d-1)^n$.

Let β be the minimum common denominator of all non-zero coefficients of h. Computing an SOS decomposition of h boils down to computing an SOS decomposition of βh. In particular, the execution time of `univsos1` (resp., `univsos2`) on h is the same as for βh. Now we estimate the bitsize of the polynomial $\beta h \in \mathbb{Z}[x_1]$. By the definition of h, we observe that $\mathrm{ht}(h) \leq \tau + d\tau_v$. It follows that $\mathrm{ht}(\beta h) \leq \mathrm{ht}(\beta) + \tau + d\tau_v$. By definition we have $\mathrm{ht}(\beta) \leq \tau + d\tau_v$. This yields

$$\mathrm{ht}(\beta h) \leq 2\left(\tau + d\tau_v\right). \tag{5.13}$$

From (5.13) and above results, we obtain the following bitsize estimate for βh:

$$\widetilde{O}\left(2(\tau + d(\tau + n + d)d^{3n})\right) = \widetilde{O}\left((\tau + n + d)d^{3n+1}\right).$$

To compute an SOS decomposition of βh, we rely on `univsos1` or `univsos2`. From Theorem 4.2.2, the boolean running time of `univsos1` corresponds to the quantity given by (5.11). If we use `univsos2` then the number of boolean operations, by applying Theorem 4.2.3, will be bounded from above by

$$\widetilde{O}\left(d^4(d-1)^{4n} + d^4(\tau + n + d)(d-1)^{6n}\right),$$

which can be further reduced to (5.12). □

Proposition 5.1.4. *Let $v_2, \ldots, v_n$ be as in Proposition 5.1.3. To compute the list $\phi_2, \ldots, \phi_n$ in the output of* `sosgradientshape`, `Eliminate` *runs in $\widetilde{O}(n^2(\tau+n+d)d^{3n+1})$ boolean operations and the bitsizes of $\phi_2, \ldots, \phi_n$ are $\widetilde{O}\left(n(\tau+n+d)d^{3n+1}\right)$.*

Proof. From Lemma 4.4.2, the bitsize of polynomial v_i is at most $\widetilde{O}((\tau+n+d)d^{3n})$. We divide f by $[x_2 - v_2, \ldots, x_n - v_n]$ while performing `Eliminate`$(f, 1, v_2, \ldots, v_n)$ described as in Algorithm 2 to obtain quotients $[\phi_2, \ldots, \phi_n]$ and remainder $h = h(x_1, v_2, \ldots, v_n)$. Applying Lemma 4.3.3 for this division, we conclude that `Eliminate` runs in $\widetilde{O}(n^2(\tau + n + d)d^{3n+1})$ boolean operations, the estimate for the bitsize of ϕ_i is $\widetilde{O}\left(n(\tau+n+d)d^{3n+1}\right)$ as claimed. □

We are now ready to analyze the bit complexity of Algorithm 3.

Theorem 5.1.5. *Let $f \in \mathbb{Q}[\boldsymbol{x}]$ of degree d and let τ be the maximum bitsize of its coefficients. Assume that the two conditions in Theorem 5.1.1 hold. Then, on input f,* `sosgradientshape` *runs in*

$$\widetilde{O}\Big((\tau+n+d)^2 d^{6n} + (\tau+n+d)d^{3n+1}(d^{n+1}/2)^{3d^{n+1}/2}\Big) \tag{5.14}$$

or

$$\widetilde{O}\Big((\tau+n+d)^2 d^{6n} + (\tau+n+d)d^{6n+4}\Big) \tag{5.15}$$

boolean operations if in Line 3 we use Algorithm `univsos1` *or Algorithm* `univsos2`*, respectively.*

Proof. Assume that in Line 3 we use `univsos1` to compute an SOS decomposition of h. Then, the number of boolean operations that `sosgradientshape` uses to compute the SOS decomposition of f is the sum of the four following numbers:

1. The number of boolean operations required to compute the zero-dimensional rational parametrization $\mathcal{Q}$ of $V_{\text{grad}}(f)$ as in (4.4).
2. The number of boolean operations required to compute $w, v_2, \ldots, v_n \in \mathbb{Q}[x_1]$, defined in Lemma 4.4.2 as in (4.6).
3. The number of boolean operations required to compute an SOS decomposition of h by using Algorithm `univsos1` as in (5.11).
4. The number of boolean operations required to compute $\phi_2, \ldots, \phi_n$ in the output of `sosgradientshape` by using `Eliminate` (mentioned in Proposition 5.1.4).

This sum equals

$$\widetilde{O}\Big(n^2(d+\tau)d^{2n+1}\binom{n+d}{d} + (\tau+n+d)^2 d^{6n} + (\tau+n+d)d^{3n+1}\Big(\frac{d^{n+1}}{2}\Big)^{3d^{n+1}/2} + (\tau+n+d)n^2 d^{3n+2}\Big).$$

In this sum, the third term is larger than the first and last term for large enough d and n, yielding the estimate (5.14). If in Line 3 we use `univsos2`, the number of boolean operations of the algorithm is

$$\widetilde{O}\Big(n^2(d+\tau)d^{2n+1}\binom{n+d}{d} + (\tau+n+d)^2d^{6n} + (\tau+n+d)d^{6n+4} + n^2(\tau+n+d)d^{3n+2}\Big).$$

Noting that $\binom{n+d}{d} \leq (d+1)^n \leq d^{2n}$ for large enough d and n, we obtain (5.15). □

Theorem 5.1.6. *Assume that $f \in \mathbb{Q}[\boldsymbol{x}]$ satisfies the conditions of Theorem 5.1.5. Let w, $v_2, \ldots, v_n$, and h be as in Proposition 5.1.3. Then, the maximum bitsize of the coefficients involved in the SOS decomposition of h obtained by using Algorithm* `univsos1` *and Algorithm* `univsos2` *are bounded from above, respectively, by*

$$\widetilde{O}\Big((\tau+n+d)(d^{n+1}/2)^{3d^{n+1}/2}d^{3n+1}\Big), \tag{5.16}$$

and

$$\widetilde{O}\left((\tau+n+d)d^{5n+3}\right). \tag{5.17}$$

Proof. From the proof of Proposition 5.1.3, the estimates for degree and bitsize of βh are $d(d-1)^n$ and $\widetilde{O}\left((\tau+n+d)d^{3n+1}\right)$, respectively. According to Theorem 4.2.2 and Theorem 4.2.3, the maximum bitsize of the coefficients involved in the SOS decomposition of βh obtained by using `univsos1` and `univsos2` are bounded from above by (5.16) and (5.17), respectively. □

5.2 SOS of rational fractions modulo gradient ideals

In this section, we explain how to decompose $f \in \mathbb{Q}[\boldsymbol{x}]$ as an SOS of rational fractions modulo its gradient ideal. We recall that $f \in \mathbb{Q}[\boldsymbol{x}]$ is an *SOS of rational fractions* in $\mathbb{Q}(\boldsymbol{x})$, where $\mathbb{Q}(\boldsymbol{x})$ is the field of rational fractions in the variable $\boldsymbol{x}$ over $\mathbb{Q}$, if there exist rational fractions $f_1, \ldots, f_s$ in $\mathbb{Q}(\boldsymbol{x})$ and $[c_1, \ldots, c_s] \subset \mathbb{Q}_+$ such that $f = \sum_{j=1}^{s} c_j f_j^2$. Furthermore, f is an SOS of rational fractions over the quotient ring $\mathbb{Q}(\boldsymbol{x})/\mathcal{I}_{\text{grad}}(f)$ if there exists $g \in \mathcal{I}_{\text{grad}}(f)$ such that $f - g$ is an SOS of rational fractions in $\mathbb{Q}(\boldsymbol{x})$, i.e., f can be decomposed as follows:

$$f = \sum_{j=1}^{s} c_j f_j^2 + \sum_{i=1}^{n} \phi_i \frac{\partial f}{\partial x_i},$$

for some rational fractions $f_1, \ldots, f_s, \phi_1, \ldots, \phi_s$ in $\mathbb{Q}(\boldsymbol{x})$ and $[c_1, \ldots, c_s] \subset \mathbb{Q}_+$.

5.2.1 The existence of an SOS decomposition over the rationals

Denote by $\mathbb{Q}(x_1)[x_2,\dots,x_n]$ the space of polynomials in $n-1$ variables $(x_2,\dots,x_n)$ with coefficients in $\mathbb{Q}(x_1)$.

In the following theorem, we prove the existence of an SOS decomposition of rational fractions modulo the gradient ideal for f.

Theorem 5.2.1. *Assume that $f \in \mathbb{Q}[\boldsymbol{x}]$ is a non-negative polynomial of degree d and that $\mathcal{I}_{\text{grad}}(f)$ is zero-dimensional and radical. Let $\mathcal{Q} = ((w,\kappa_1,\dots,\kappa_n),x_1)$ be a zero-dimensional rational parametrization of $V_{\text{grad}}(f)$. Then, f can be decomposed as an SOS of rational fractions modulo the gradient ideal, in particular*

$$f = \frac{1}{(w')^d}\sum_{j=1}^{s} c_j q_j^2 + \sum_{i=1}^{n} \frac{\phi_i}{(w')^d}\left(w'x_i - \kappa_i\right), \tag{5.18}$$

for some $q_1,\dots,q_s \in \mathbb{Q}[x_1]$, $\phi_1,\dots,\phi_n \in \mathbb{Q}[\boldsymbol{x}]$, and $[c_1,\dots,c_s] \subset \mathbb{Q}_+$.

Proof. The gradient variety of f can be represented as follows:

$$V_{\text{grad}}(f) = \{\boldsymbol{x} \in \mathbb{C}^n : w'x_1 - \kappa_1 = \cdots = w'x_n - \kappa_n = 0\}. \tag{5.19}$$

Because $\mathcal{I}_{\text{grad}}(f)$ is radical, according to Theorem 3.1.1 (Hilbert's Strong Nullstellensatz), one has

$$\mathcal{I}_{\text{grad}}(f) = \left\langle w'x_1 - \kappa_1,\dots,w'x_n - \kappa_n\right\rangle.$$

We now apply the argument in the proof of Theorem 5.1.1. By substituting $x_i = \frac{\kappa_i}{w'}$ in f, for $i = 2,\dots,n$, we obtain a univariate polynomial $\bar{h}(x_1)$ such that

$$f\left(x_1, \frac{\kappa_2}{w'},\dots,\frac{\kappa_n}{w'}\right) = \frac{1}{(w')^d}\bar{h}. \tag{5.20}$$

Since f is non-negative with even degree d, $\bar{h}$ is also non-negative. In addition, the coefficients of $w',\kappa_1,\dots,\kappa_n$ and f are rational numbers. Therefore, the coefficients of $\bar{h}$ are also rational numbers. Applying Theorem 4.2.1 for $\bar{h}$, we conclude that there are $q_1,\dots,q_s$ in $\mathbb{Q}[x_1]$ and $[c_1,\dots,c_s] \subset \mathbb{Q}_+$ such that

$$\bar{h} = \sum_{j=1}^{s} c_j q_j^2. \tag{5.21}$$

Next, one considers the division of $(w')^d f - \bar{h}$ by $[w'x_1 - \kappa_1,\dots,w'x_n - \kappa_n]$ with the lexicographic order $x_1 < \cdots < x_n$. Based on Buchberger's Criterion (Theorem 3.2.1), we can show that $[w'x_1 - \kappa_1,\dots,w'x_n - \kappa_n]$ is a Gröbner basis of the ideal generated

by this system with respect to the order $<$ in $\mathbb{Q}[\boldsymbol{x}]$. Hence, there exist a (unique) list of quotients $[\phi_1, \ldots, \phi_n]$ in $\mathbb{Q}[\boldsymbol{x}]$, and r in $\mathbb{Q}[x_1]$ such that

$$(w')^d f - \bar{h} = \sum_{i=1}^{n} \phi_i (w' x_i - \kappa_i) + r, \tag{5.22}$$

with r of smaller degree than the cardinality δ of $V_{\text{grad}}(f)$. Note that the gradient variety of f can be represented as in (5.19). From (5.20), one sees that $(w')^d f - \bar{h}$ vanishes on $V_{\text{grad}}(f)$. With the same arguments as in the proof of Theorem 5.1.1, we conclude that $r \equiv 0$. Hence, from (5.20), (5.21), and (5.22), we obtain a representation of f as in (5.18). □

Remark 5.2.1. In Theorem 5.2.4, we assume that $\mathcal{Q} = ((w, \kappa_1, \ldots, \kappa_n), x_1)$ is a zero-dimensional rational parametrization of $V_{\text{grad}}(f)$ which is a generic assumption. In this assumption, the linear form λ is given by $\lambda(\boldsymbol{x}) = x_1$. If this assumption does not hold, we can change the coordinate system such that the obtained polynomial (with new variables) satisfies this assumption as in Case 2 of the proof of Theorem 5.1.1.

Remark 5.2.2. Denote by $\deg_{x_1} f$ the degree of f in the variable x_1. From (5.20), we see that $\deg \bar{h}$ does not exceed $\deg_{x_1} f + d \deg(w')$, where $\deg w' = \deg w - 1$. Thus, the degree of the univariate polynomial $\bar{h}$ is at most $d(d-1)^n$.

Remark 5.2.3. From (5.20), one can see that $(w')^d f - \bar{h}$ vanishes on the variety defined by $\langle w' x_2 - \kappa_2, \ldots, w' x_n - \kappa_n \rangle$. Hence, ϕ_1 in (5.18) is zero and (5.18) becomes

$$f = \frac{1}{(w')^d} \sum_{j=1}^{s} c_j q_j^2 + \sum_{i=2}^{n} \frac{\phi_i}{(w')^d} \left(w' x_i - \kappa_i \right). \tag{5.23}$$

To illustrate the formula (5.23), we consider the simplest case with $n = 2$ and $d = 2$ as follows.

Example 5.2.1. Consider the polynomial $f_E = 2x_1^2 + 4x_1 x_2 + x_2^4 + 3$, that is non-negative over $\mathbb{R}^n$. The gradient variety of f has a zero-dimensional rational parametrization $\mathcal{Q} = ((w, \kappa_1, \kappa_2), \lambda)$ given by

$$\lambda = x_1,\ w = x_1^3 - x_1,\ \kappa_1 = -2x_1,\ \kappa_2 = 2x_1.$$

One has $w' = 3x_1^2 - 1$. By replacing $x_2 = \kappa_2 / w'$ in f_E, we obtain

$$f_E\left(x_1, \frac{\kappa_2}{w'}\right) = 2x_1^2 + 4x_1\left(\frac{\kappa_2}{w'}\right) + \left(\frac{\kappa_2}{w'}\right)^4 + 3 = \frac{\bar{h}}{(3x_1^2 - 1)^4},$$

where

$$\bar{h} = 162x_1^{10} + 243x_1^8 - 432x_1^6 + 226x_1^4 - 42x_1^2 + 3.$$

By using `univsos2` to compute an SOS decomposition of h, one has

$$\bar{h} = \tfrac{82863}{512}\left(x_1^5 - \tfrac{6493}{4096}x_1^3 + \tfrac{20213}{65536}x_1\right)^2 + \tfrac{82863}{512}\left(\tfrac{70819}{32768}x_1^4 + \tfrac{70819}{32768}x_1^2 + \tfrac{2171}{16384}\right)^2 + \tfrac{81}{512}x_1^{10} + \tfrac{87698926233}{549755813888}x_1^8 + \tfrac{43587401805}{274877906944}x_1^6 + \tfrac{89433872333}{549755813888}x_1^4 + \tfrac{343326283161}{2199023255552}x_1^2 + \tfrac{21763571433}{137438953472}.$$

We only need to compute ϕ_2, which we find to be

$$\phi_2 = (81x_1^8 - 108x_1^6 + 54x_1^4 - 12x_1^2 + 1)x_2^3 + 324x_1^9 - 432x_1^7 + 216x_1^5 - 48x_1^3 + 4x_1 + \tfrac{1}{3x_1^2-1}(162x_1^9 - 216x^7 + 108x_1^5 - 24x_1^3 + 2x_1)x_2^2 + \tfrac{1}{(3x_1^2-1)^2}(324x_1^{10} - 432x_1^8 + 216x_1^6 - 48x_1^4 + 4x_1^2)x_2 + \tfrac{1}{(3x_1^2-1)^3}(648x_1^{11} - 864x_1^9 + 432x_1^7 - 96x_1^5 + 8x_1^3).$$

From the above results, we obtain an SOS decomposition of rational functions modulo the gradient ideal of f_E.

5.2.2 Algorithm to compute an SOS of rational fractions

From the proof of Theorem 5.2.1, we design the algorithm `sosgradient` to compute an SOS decomposition of rational fractions for f. This algorithm is obtained by a modification of Line 1 in `sosgradientshape` to get a zero-dimensional rational parametrization of the gradient variety of f.

Algorithm 4 Computing SOS of rational fractions modulo the gradient ideal

`sosgradient` := proc(f)

Input: $f \in \mathbb{Q}[\boldsymbol{x}]$ of degree d such that f is non-negative over $\mathbb{R}^n$ and $\mathcal{I}_{\text{grad}}(f)$ is zero-dimensional and radical

Output: $[w, \kappa_1, \ldots, \kappa_n]$, $[q_1, \ldots, q_s] \subset \mathbb{Q}[x_1]$, $[\psi_2, \ldots, \psi_n] \subset \mathbb{Q}(x_1)[x_2, \ldots, x_n]$, and $[c_1, \ldots, c_s] \subset \mathbb{Q}_+$ satisfying

$$f = \frac{1}{(w')^d}\sum_{j=1}^{s} c_j q_j^2 + \sum_{i=2}^{n} \frac{\psi_i}{(w')^d}\Big(x_i - \frac{\kappa_i}{w'}\Big). \tag{5.24}$$

1: Compute a zero-dimensional rational parametrization $[w, \kappa_1, \ldots, \kappa_n]$ of $V_{\text{grad}}(f)$

2: Compute the quotients $[\psi_2, \ldots, \psi_n]$ and the remainder $\bar{h}$ of the division of $(w')^d f$ by

$$[x_2 - \frac{\kappa_2}{w'}, \ldots, x_n - \frac{\kappa_n}{w'}]$$

by performing `Eliminate`$((w')^d f, w', \kappa_2, \ldots, \kappa_n)$

3: Compute a rational weighted SOS decomposition of $\bar{h} = c_1 q_1^2 + \cdots + c_s q_s^2$

4: **return** $[w, \kappa_1, \ldots, \kappa_n]$, $[q_1, \ldots, q_s]$, $[\psi_2, \ldots, \psi_n]$, and $[c_1, \ldots, c_s]$

The input of `sosgradient` is a non-negative polynomial f in $\mathbb{Q}[\boldsymbol{x}]$ whose gradient ideal $\mathcal{I}_{\text{grad}}(f)$ is zero-dimensional and radical. The outputs are a zero-dimensional rational parametrization of $V_{\text{grad}}(f)$, a list of univariate polynomials $[q_1, \ldots, q_s] \subset \mathbb{Q}[x_1]$, and a list $[\psi_2, \ldots, \psi_n]$ in $\mathbb{Q}(x_1)[x_2, \ldots, x_n]$ satisfying (5.24). Note that the ψ_i's in (5.18) and ϕ_i's

(5.24) are different up to a multiplier w', in particular $\psi_i = w'\phi_i$. Here, we prefer using ψ_i as computing ψ_i's through the division algorithm `Eliminate` is convenient.

In Line 1, we compute a zero-dimensional rational parametrization $[w, \kappa_1, \ldots, \kappa_n]$ of the variety $V_{\text{grad}}(f)$. In Line 2, by using Algorithm `Eliminate`, we compute the quotients $[\psi_2, \ldots, \psi_n]$ of the division of $(w')^d f$ by

$$[x_2 - \frac{\kappa_2}{w'}, \ldots, x_n - \frac{\kappa_n}{w'}].$$

Note that the remainder of this division coincides with $\bar{h}$ given in (5.20). In Line 3, we compute a rational weighted SOS decomposition of the univariate polynomial $\bar{h}$ by relying on Algorithms `univsos1` or `univsos2`.

The correctness of `sosgradient` is proved in a similar way as for `sosgradientshape` in Theorem 5.1.2.

Theorem 5.2.2. *Let $f \in \mathbb{Q}[\boldsymbol{x}]$ be non-negative over $\mathbb{R}^n$ and $\mathcal{I}_{\text{grad}}(f)$ be zero-dimensional and radical. On input f, Algorithm* `sosgradient` *terminates and the outputs provide us an SOS decomposition of f as in* (5.18).

We present an explicit SOS decomposition for the polynomial f_S which was obtained from Scheiderer's polynomial in Example 5.1.2. Here, we rely on `sosgradient` to get the SOS decomposition.

Example 5.2.2. We first compute a zero-dimensional rational parametrization $\mathcal{Q}$ of the gradient variety $V_{\text{grad}}(f_S)$:

$$\begin{aligned} w &= 4x_1^9 + x_1^6 - 16x_1^5 - 4x_1^3 - 4x_1^2 - 1, \\ \kappa_1 &= 15x_1^7 - 32x_1^6 - 9x_1^4 - 36x_1^3 - 6x_1 - 4, \\ \kappa_2 &= -3x_1^6 + 64x_1^5 + 24x_1^3 + 28x_1^2 + 9. \end{aligned}$$

In f_S, by substituting $x_2 = \kappa_2/w'$ as in (5.20), we get the non-negative univariate polynomial $\bar{h} = 1679616x_1^{36} + 3359232x_1^{34} - 559872x_1^{33} - 13670208x_1^{32} + 11197440x_1^{31} - 32799168x_1^{30} + 7301664x_1^{29} + 40124160x_1^{28} - 56581740x_1^{27} + 118393488x_1^{26} - 29030400x_1^{25} - 11429649x_1^{24} + 91968984x_1^{23} - 162286560x_1^{22} + 52664472x_1^{21} - 95470992x_1^{20} - 51948224x_1^{19} + 37314854x_1^{18} - 36173624x_1^{17} + 103156448x_1^{16} + 27660704x_1^{15} + 94133752x_1^{14} + 56849248x_1^{13} + 51186288x_1^{12} + 42348048x_1^{11} + 20765728x_1^{10} + 17391200x_1^9 + 7273168x_1^8 + 4607744x_1^7 + 1946186x_1^6 + 880960x_1^5 + 413632x_1^4 + 86580x_1^3 + 75816x_1^2 + 6561$.

Through Algorithm `Eliminate`, we obtain the quotients of the division in Line 2 of `sosgradient`: $\psi_1 = 0$ and ψ_2 given at polsys.lip6.fr/∼hieu/phisos.mm. By using `univsos2` to compute an SOS decomposition of $\bar{h}$, we obtain the list `sos` given in the above webpage such that $\bar{h} = \sum_{i=1}^{m} \texttt{sos}[2i-1]\texttt{sos}[2i]^2$, where $\texttt{sos}[i]$ stands for the i-th entry of `sos` and m is the half length of `sos`. Combining the above results, we obtain an SOS of rational fractions modulo the gradient of f_S as in (5.24).

5.2.3 Bit complexity analysis

To conclude this section, we estimate the bitsizes of the polynomials in the output as well as the number of boolean operations required to perform Algorithm `sosgradient`.

Proposition 5.2.3. *Assume that τ is the bitsize of f in the input of* `sosgradient`*. To compute $[\psi_2, \dots, \psi_n]$ in the output,* `Eliminate` *runs in $\widetilde{O}\left(n^2(\tau+n+d)d^{n+1}\right)$ boolean operations. Moreover, the bitsize of ψ_i is $\widetilde{O}\left(n(\tau+n+d)d^{n+1}\right)$, for $i = 2, \dots, n$.*

Proof. We compute the division of $(w')^d f$ by $[x_2 - \frac{\kappa_2}{w'}, \dots, x_n - \frac{\kappa_n}{w'}]$ by performing the procedure `Eliminate`$((w')^d f, w', \kappa_2, \dots, \kappa_n)$. We obtain the list of quotients $[\psi_2, \dots, \psi_n]$ and the remainder $\bar{h}$. The degree of $(w')^d f$ in variables $x_2, \dots, x_n$ is d, and the height is

$$\mathrm{ht}((w')^d f) = \widetilde{O}\left((\tau+n+d)d^{n+1}\right).$$

By applying Lemma 4.3.3 with $\mathrm{ht}(\kappa_i) = \widetilde{O}\left((\tau+n+d)(d-1)^n\right)$, we obtain the conclusions. □

Theorem 5.2.4. *Let $f \in \mathbb{Q}[\boldsymbol{x}]$ of degree d and let τ be the maximum bitsize of its coefficients. Assume that f is non-negative over $\mathbb{R}^n$ and that $\mathcal{I}_{\mathrm{grad}}(f)$ is zero-dimensional and radical. Then, on input f, Algorithm* `sosgradient` *uses*

$$\widetilde{O}\Big((d^{n+1}/2)^{3d^{n+1}/2}(\tau+n+d)d^{n+1}\Big), \tag{5.25}$$

or

$$\widetilde{O}((\tau+n+d)d^{4n+4}) \tag{5.26}$$

boolean operations if in Line 3 we use Algorithm `univsos1` *or Algorithm* `univsos2`*, respectively.*

Proof. From Corollary 4.4.1, $w, \kappa_1, \dots, \kappa_n$ in the zero-dimensional parametrization of $V_{\mathrm{grad}}(f)$ have degree at most $(d-1)^n$ and bitsize $\widetilde{O}\left((\tau+n+d)(d-1)^n\right)$. The degree of the remainder $\bar{h}$ (as defined in (5.20)) in Line 2 of `sosgradient` is at most $d(d-1)^n + d$ and its bitsize is $\widetilde{O}\left((\tau+n+d)d^{n+1}\right)$. To compute an SOS decomposition of $\bar{h}$, by applying Theorems 4.2.2 and 4.2.3, Algorithms `univsos1` and `univsos2` use

$$\widetilde{O}\Big((d^{n+1}/2)^{3d^{n+1}/2}(\tau+n+d)d^{n+1}\Big) \tag{5.27}$$

and

$$\widetilde{O}\left((\tau+n+d)d^{4n+4}\right) \tag{5.28}$$

boolean operations, respectively. The estimates (5.25) and (5.26) are obtained from Corollary 4.4.1, Proposition 5.2.3, and the estimates (5.27) and (5.28) with the same line of reasoning as in the proof of Theorem 5.1.5. □

Theorem 5.2.5. *Assume that $f \in \mathbb{Q}[\boldsymbol{x}]$ satisfies the conditions of Theorem 5.2.4. Then, the maximum bitsizes of the coefficients involved in the SOS decomposition of $\bar{h}$, obtained by using Algorithm* `univsos1` *and Algorithm* `univsos1`*, are bounded from above respectively by*

$$\widetilde{O}\Big((d^{n+1}/2)^{3d^{n+1}/2}(\tau+n+d)d^{n+1}\Big)$$

and

$$\widetilde{O}\left((\tau+n+d)d^{3n+3}\right).$$

Proof. From the proof of Theorem 5.2.4, the degree of $\bar{h}$ is at most $d(d-1)^n$ and the bitsize of $\bar{h}$ is $\widetilde{O}\left((\tau+n+d)d^{n+1}\right)$. The conclusions follow from Theorems 4.2.2 and 4.2.3 and the second assertion in Proposition 5.2.3. □

Remark 5.2.4. In general, `sosgradient` is faster than `sosgradientshape` at certifying non-negativity of polynomials with rational coefficients. When relying on `univsos2`, by comparing the estimates in (5.15) and (5.26), we conclude that the number of boolean operations to run `sosgradientshape` is about d^{2n} times larger than that of `sosgradient`. The underlying reason is that the maximum bitsizes of $w, v_2, \ldots, v_n$ are $(d-1)^{2n}$ times bigger than the ones of $\kappa_1, \ldots, \kappa_n$ that are obtained by a zero-dimensional rational parametrization of the gradient variety.

5.3 Practical experiments

This section is dedicated to showing experimental results obtained by using the algorithms `sosgradientshape` (Algorithm 3 from Section 5.1) and `sosgradient` (Algorithm 4 from Section 5.2). Both algorithms are implemented in MAPLE, and the results are obtained on an Intel Xeon E7-4820 CPU (2GHz) with 1.5 TB of RAM.

In practice, `univsos2` runs faster than `univsos1`, which is consistent with the theoretical results stated in Theorems 4.2.2 and 4.2.3. In addition, as mentioned in Remark 5.2.4, it is practically faster to compute SOS decompositions involving rational fractions than polynomials.

We compare timings of the slowest algorithm, `sosgradientshape` using `univsos1`, with the fastest algorithm, `sosgradient` using `univsos2`. For each algorithm, the first step consists of obtaining h by computing either the reduced Gröbner basis (using the procedure `Basis` in MAPLE) in `sosgradientshape` or the zero-dimensional rational parametrization (using the procedure `RationalUnivariateRepresentation` in MAPLE) in `sosgradient`. The runtime of this step is denoted by t_h. The degree and the bitsize of h are denoted by d_h and τ_h, respectively. The second step outputs an SOS decomposition of the non-negative univariate polynomial h by using either Algorithm `univsos1` in

`sosgradientshape` or Algorithm `univsos2` in `sosgradient`. Here, $t_{\texttt{sos}}$ is the runtime of the second step and $\tau_{\texttt{sos}}$ is the maximum bitsize of the output polynomials.

			`sosgradientshape`					`sosgradient`				
				bitsize 10^6-bits		time (s)			bitsize 10^4-bits		time (s)	
n	τ	δ	d_h	τ_h	$\tau_{\texttt{sos}}$	t_h	$t_{\texttt{sos}}$	d_h	τ_h	$\tau_{\texttt{sos}}$	t_h	$t_{\texttt{sos}}$
2	74	9	32	0.3	8.1	0.1	2.6	36	0.5	1.6	0.1	1.8
3	149	27	104	2.4	153	1.1	781	108	6.6	13.4	0.2	13.3
4	312	81	320	117	–	399	–	324	88	169	3.9	505
5	590	243	968	–		–		972	940	1306	169	4965

Table 5.1: Comparison of performance between `sosgradientshape` and `sosgradient`

In Table 5.1, we consider random polynomials of fixed degree $d = 4$ with number of variables n between 2 and 5 generated as follows: $a^4 + b_1^2 + \cdots + b_n^2 + c + 10^6$, where a (resp., b_i, c) is a dense linear (resp., quadratic, cubic) polynomial in n variables. The coefficients of a (resp., b_i, c) are chosen randomly in $\{-1, 1\}$ (resp., $\{-3, \ldots, 3\}$, $\{-1, 0, 1\}$) with respect to the uniform distribution. For $n \geq 4$, `sosgradientshape` failed to provide an SOS decomposition as the execution of `univsos1` did not finish after 12 hours of computation, as indicated by the symbol "$-$" in the corresponding lines. The underlying reason is that τ_h and d_h are both very large and that the complexity of `univsos1` is exponential in the degree of h (Theorem 4.2.2). Note that the intermediate polynomials correspond to worst cases, i.e., the maximum possible degree of w is attained, namely $\delta = \deg w = (d-1)^n$, so the degree of h is also maximum, i.e., $\deg h = d(d-1)^n - d$ (resp. $d(d-1)^n$) in `sosgradientshape` (resp. in `sosgradient`). For such cases, `sosgradient` cannot compute decompositions for $n \geq 4$ (corresponding to $\deg h \geq 324$) within 12 hours.

	`multivsos`			`sosgradient`	
d, n	success	τ	t	τ	t
4, 2	100%	1.3	0.16	2	2
4, 3	94%	3.7	0.26	18	22
4, 4	38%	8.9	0.18	78	153
4, 5	8%	12.5	0.32	234	630
6, 2	82%	3.5	0.24	45	142
6, 3	0%			160	500
6, 4	0%			744	4662

Table 5.2: Comparison of performance between `sosgradientshape` and `multivsos`

Next, we compare the performance of `sosgradient` (using `univsos2`) and Algorithm `multivsos` [51]. Recall that `multivsos` is designed to compute SOS decompositions of polynomials lying in the interior of the SOS cone. We report our experimental results in Table 5.2, obtained with seven classes of 50 randomly generated polynomials. The random polynomials corresponding to the four first rows, with $d = 4$ and $n = 2, \ldots, 5$, are obtained in a similar way: $a^4 + b_1^2 + b_2^2 + c + 10^6$, where a (resp., b_i, c) is a dense

linear (resp., quadratic, cubic) polynomial in n variables. The coefficients of a (resp., b_i, c) are chosen randomly in $\{\pm 1, \pm 2\}$ (resp., $\{-3, \ldots, 3\}$, $\{-1, \ldots, 1\}$) with respect to the uniform distribution. The polynomials from the three last rows, with $d = 6$ and $n = 2, 3, 4$, are constructed in a similar way: $a^6 + b^2 + c + 10^6$, where a (resp., b, c) is a dense linear (resp., cubic, cubic) polynomial in n variables. Coefficients of a (resp., b_i, c) are chosen randomly in $\{\pm 1, \pm 2\}$ (resp., $\{-3, \ldots, 3\}$, $\{-1, \ldots, 1\}$) with respect to the uniform distribution. Note that here the univariate polynomials generated when running the algorithm do not correspond to the worst case scenario in terms of degree and bitsize. For both algorithms, we denote by τ (10^4-bits) the average bitsize of the output and by t the average runtime in seconds.

From this table, we deduce that when the number of variables n increases, then the rate of success of `multivsos` decreases. This fact illustrates Blekherman's theorem [15] which says that if the degree $d \geq 4$ is fixed then, as the number of variables n grows, the cone of non-negative polynomials is significantly bigger than the cone of SOS polynomials. It also illustrates that `sosgradient` can tackle a large range of problems which are out of reach of state-of-the-art algorithms such as `multivsos`. When `multivsos` succeeds in computing SOS decompositions, then it provides more concise certificates than `sosgradient` while also being more efficient. However, when $d = 4$ and $n = 5$, `multivsos` can only decompose 4 polynomials out of 50 while `sosgradient` succeeds for all of them. This demonstrates the need of alternative procedures such as `sosgradient` for polynomials which presumably do not lie in the interior of the SOS cone.

CHAPTER 6

Exact SOHS decompositions of trigonometric univariate polynomials with Gaussian coefficients

Contents

In this chapter, we design, analyze and compare, theoretically and practically, three hybrid numeric-symbolic algorithms for computing weighted sums of Hermitian squares decompositions for trigonometric univariate polynomials positive on the unit circle with Gaussian coefficients. The numerical steps on which the first and second algorithm rely are complex root isolation and semi-definite programming, respectively. Exact sum of Hermitian squares decompositions are obtained thanks to compensation techniques. The third algorithm, also based on complex semi-definite programming, is an adaptation of the rounding and projection algorithm by Peyrl and Parrilo [67]. We compare their performances on randomly chosen benchmarks, and further design a certified finite impulse filter.

This chapter has three sections. Sections 6.1, 6.2, and 6.3 introduce Algorithms `csos1`, `csos2`, and `csos3`, respectively, and analyze their bit complexities. Practical experiments are given in Section 6.4.

Most of the content of this chapter is from the paper [54] entitled "*Exact SOHS decompositions of trigonometric univariate polynomials with Gaussian coefficients*" by Victor Magron, Mohab Safey El Din, Markus Schweighofer, and Trung Hieu Vu.

6.1 Algorithm based on root isolation

The set of trigonometric univariate polynomials with Gaussian integer coefficients, denoted by $\mathscr{H}(\mathbb{Z})[z]$, is a subset of Laurent polynomials with complex variable z as follows:

$$f(z) = f_0 + \left(\frac{f_1}{z} + \bar{f}_1 z\right) + \cdots + \left(\frac{f_d}{z^d} + \bar{f}_d z^d\right), \tag{6.1}$$

with $d \in \mathbb{N}$, $f_0 \in \mathbb{Z}$ and $f_j \in \mathbb{Z}[\boldsymbol{i}]$ for $j = 1, \ldots, d$.

Assume that $f \in \mathscr{H}(\mathbb{Z})[z]$ is positive on the unit circle. According to the proof of the Riesz-Fejér spectral factorization theorem [22, pp. 3–5], f can be written as an SOHS with a single term as follows:

$$f = a \times \prod_{k=1}^{d} (z - a_k) \times \left(\frac{1}{z} - \bar{a}_k\right), \tag{6.2}$$

where $(a_1, 1/\bar{a}_1) \ldots, (a_d, 1/\bar{a}_d)$ are d pairs of roots of f, and a is a positive scalar.

We propose the first algorithm, called `csos1`, to compute an exact SOHS decomposition of $f \in \mathscr{H}(\mathbb{Z})[z]$ being positive on the unit circle $\mathscr{C}$. It puts into practice a perturbation-compensation procedure based on complex roots isolation, and can be viewed as the extension of the procedure `univsos2` (stated in [18] and analyzed in [53, § 4]) to the complex setting.

6.1.1 Description and correctness

Description. Algorithm `csos1` takes as input a polynomial $f \in \mathscr{H}(\mathbb{Z})[z]$ of degree d which is positive on $\mathscr{C}$ and that f has no multiple roots. It outputs two positive rational numbers ε and α, a rational number u_0, and two lists of Gaussian numbers $[u_1, \ldots, u_d]$ and $[\alpha_1, \ldots, \alpha_d]$ such that

$$\begin{aligned} f(z) = \left(\varepsilon + u_0 - 2\sum_{k=1}^{d} |u_k|\right) + \sum_{k=1}^{d} |u_k| \left(z^k + \frac{u_k}{|u_k|}\right)\left(\frac{1}{z^k} + \frac{\bar{u}_k}{|u_k|}\right) \\ + \alpha \prod_{k=1}^{d} (z - \alpha_k)\left(\frac{1}{z} - \bar{\alpha}_k\right) \text{ with } \left(\varepsilon + u_0 - 2\sum_{k=1}^{d} |u_k|\right) > 0. \end{aligned} \tag{6.3}$$

In Line 1 we replace z by $x + \boldsymbol{i}y$ in f where x, y are (real) variables to obtain a real bivariate polynomial p of degree d. Since, by assumption, f is positive over the compact set $\mathscr{C}$, there exists $\varepsilon > 0$ small enough, such that $p - \varepsilon$ is positive on $\mathscr{C}$. The first while loop from Lines 2 and 3 computes such positive rational number ε. To do so, it uses an auxiliary procedure `hasrealrootoncircle`, which returns true if $p - \varepsilon$ cancels on the unit circle $\mathscr{C} = \{(x, y) \in \mathbb{R}^2 : x^2 + y^2 - 1 = 0\}$. Such a procedure is easily obtained

with any polynomial system solver for bivariate polynomial systems. In practice, we use the real root solver MSOLVE [12].

In the second while loop from Line 5 to 13, the algorithm computes at Line 6 Gaussian approximations $\alpha_1, \ldots, \alpha_d$, where $|\alpha_i| < 1$ for $i = 1, \ldots, d$ (and their conjugates), of the complex roots of $f - \varepsilon$ with accuracy δ. This is done using the procedure `complexroots` which on input a rational fraction and a required accuracy δ returns all the complex roots of the numerator of the fraction at accuracy δ (see, e.g., [14]).

Algorithm 5 Computing SOHS decomposition based on root isolation

`csos1` := proc(f)

Input: $f \in \mathscr{H}(\mathbb{Z})[z]$ is positive on $\mathscr{C}$ of degree d and has no multiple roots

Output: $\varepsilon, a \in \mathbb{Q}_+$, $u_0 \in \mathbb{Q}$, two lists $[u_1, \ldots, u_d]$ and $[\alpha_1, \ldots, \alpha_d]$ in $\mathbb{Q}[i]$ providing an SOHS decomposition of f on $\mathscr{C}$ as in (6.3)

1: Set $\delta := 1, \varepsilon := 1$ and compute $p := f(x + iy)$ ▷ $z = x + iy,\ z^{-1} = x - iy$
2: **while** `hasrealrootoncircle`$(p - \varepsilon)$ **do** $\varepsilon := \frac{\varepsilon}{2}$
3: **done**
4: Set boo := false
5: **while** not boo **do**
6: Compute $[\alpha_1, \ldots, \alpha_d] :=$ `complexroots`$(f - \varepsilon, \delta)$ ▷ $|\alpha_i| < 1,\ i = 1, \ldots, d$
7: Compute $F := \prod_{k=1}^{d} (z - \alpha_k)\left(z^{-1} - \bar{\alpha}_k\right)$
8: Compute $\alpha :=$ `coeffs`$(f - \varepsilon, 0)/$`coeffs`$(F, 0)$ and $u := (f - \varepsilon) - \alpha F$
9: Compute $[u_0, u_1, \ldots, u_d] :=$ `coeffs`(u)
10: **if** $\varepsilon > 2\sum_{k=1}^{d} |u_k| - u_0$ **then** boo := true
11: **else** $\delta := 2\delta$
12: **end**
13: **done**
14: **return** $\varepsilon, \alpha, u_0, [u_1, \ldots, u_d], [\alpha_1, \ldots, \alpha_d]$

The idea is to obtain, up to proper scaling α, an approximate SOHS decomposition F of $f - \varepsilon$. The auxiliary procedure `coeffs` provides the list of coefficients of a polynomial, e.g., `coeffs`$(f - \varepsilon, 0)$ returns the constant term of $f - \varepsilon$. We then consider the difference u at Line 8 which is the difference between $f - \varepsilon$ and its approximate SOHS decomposition which can be written as follows:

$$u = u_0 + \left(u_1 z^{-1} + \bar{u}_1 z\right) + \cdots + \left(u_d z^{-d} + \bar{u}_d z^d\right).$$

As proved in Section 6.1.2, if the precision of root isolation is large enough, then the stopping condition

$$\varepsilon > 2\sum_{k=1}^{d} |u_k| - u_0 \tag{6.4}$$

is fulfilled, otherwise the precision is increased.

To illustrate `csos1`, we use the following simple example.

Example 6.1.1. Let $f = 5 + (1+\boldsymbol{i})z^{-1} + (1-\boldsymbol{i})z$ which is positive on $\mathscr{C}$. We obtain $p = 5 + 2x + 2y$. With $\varepsilon = 1$, we check with `hasrealrootoncircle` that $p - \varepsilon$ is positive on $\mathscr{C}$. With precision $\delta = 16$, we compute complex approximation roots α_1 and $\bar{\alpha}_1$ of $f - \varepsilon$, here $\alpha_1 = -\frac{7}{4} - \frac{7}{4}\boldsymbol{i}$. Defining $F = (z - \alpha_1)(z^{-1} - \bar{\alpha}_1)$, we obtain $a = \frac{32}{57}$,

$$u = f - \varepsilon - aF = \Big(\frac{1}{57} + \frac{\boldsymbol{i}}{57}\Big)z^{-1} + \Big(\frac{1}{57} - \frac{\boldsymbol{i}}{57}\Big)z.$$

Clearly, $\varepsilon = 1 > \frac{2\sqrt{2}}{57}$, thus the condition (6.4) is satisfied. Then, f has an exact SOHS decomposition as follows:

$$f = \Big(1 - \frac{2\sqrt{2}}{57}\Big) + \frac{\sqrt{2}}{57}\Big(z + \frac{1+\boldsymbol{i}}{\sqrt{2}}\Big)\Big(z^{-1} + \frac{1-\boldsymbol{i}}{\sqrt{2}}\Big) + \frac{32}{57}\Big(z + \frac{7}{4} + \frac{7}{4}\boldsymbol{i}\Big)\Big(z^{-1} + \frac{7}{4} - \frac{7}{4}\boldsymbol{i}\Big).$$

To prove the correctness of the two algorithms `csos1` and `csos2` and estimate their bit complexities, we need the following lemma.

Lemma 6.1.1. *Let $f \in \mathscr{H}(\mathbb{Z})[z]$ be positive on $\mathscr{C}$, of degree d and τ be the maximum bitsize of its coefficients. Then, there exists a positive integer $N = \widetilde{O}(d^3(d+\tau))$ such that $f - \frac{1}{2^N}$ is positive on $\mathscr{C}$.*

Proof. We replace z by $x + \boldsymbol{i}y$ and z^{-1} by $x - \boldsymbol{i}y$ in f, then obtain $p(x,y)$. Because f belongs to $\mathscr{H}(\mathbb{Z})[z]$, one has $p \in \mathbb{Z}[x,y]$ with degree d and bitsize $O(\mathrm{ht}(d) + \tau)$. Clearly,

$$p_{\min} := \min\{p(x,y) : x^2 + y^2 = 1\} = \min\{f(z) : |z| = 1\}.$$

From Lemma 4.6.2, we can choose a positive integer $N = \widetilde{O}(d^3(d+\tau))$ large enough such that $p_{\min} > \frac{1}{2^N}$. This implies that $f - \frac{1}{2^N}$ is positive on $\mathscr{C}$. □

Theorem 6.1.2. *Assume that $f \in \mathscr{H}(\mathbb{Z})[z]$ is positive on $\mathscr{C}$ of degree d and that f has no multiple roots. On input f, Algorithm `csos1` terminates and outputs an SOSH decomposition of f as in (6.3).*

Proof. By Lemma 6.1.1, there exits a positive rational ε such that $f - \varepsilon$ is also positive on $\mathscr{C}$. Thus, the first while loop (from Line 2 to Line 3) of Algorithm `csos1` terminates. The magnitude of the coefficients of the difference polynomial u defined in Line 9 converges to 0 as the precision δ of the complex root finder goes to infinity (because of the continuity of roots with respect to coefficients). This implies that the condition of Line 10 is fulfilled after finitely many iterations, thus the second loop (from Line 5 to Line 13) always terminates. Eventually, we have

$$f = \varepsilon + u_0 + \Big(u_1 z^{-1} + \bar{u}_1 z\Big) + \cdots + \Big(u_d z^{-d} + \bar{u}_d z^d\Big) + aF.$$

In addition,

$$u_k z^{-k} + \bar{u}_k z^k = |u_k|\left(z^k + \frac{u_k}{|u_k|}\right)\left(z^{-k} + \frac{\bar{u}_k}{|u_k|}\right) - 2|u_k|, \tag{6.5}$$

yielding (6.3). The scaling α at Line 8 is actually an approximation of the scaling a from the decomposition (6.2) of f. Since both $\texttt{coeffs}(f-\varepsilon, 0)$ and $\texttt{coeffs}(F, 0)$ are rational, α is also rational.

Clearly, the polynomial F and the term on the right-hand side of (6.5) are SOHS. Hence, as the stopping condition (6.4), the right-hand side of (6.3) is a sum of $d+2$ Hermitian squares involving Gaussian (or Gaussian modulus) numbers. □

6.1.2 Bit complexity analysis

We now analyze the bit complexity of Algorithm `csos1`.

Theorem 6.1.3. *Assume that $f \in \mathscr{H}(\mathbb{Z})[z]$ is positive on $\mathscr{C}$ of degree d, bitsize τ and that f has no multiple roots. On input f, `csos1` computes an SOHS decomposition of f with Gaussian (or Gaussian modulus) coefficients whose the maximum bitsize is bounded from above by $\widetilde{O}(d^5(d+\tau))$.*

Proof. Firstly, let us show that the bitsizes of $u_0, \ldots, u_d$, $\alpha_1, \ldots, \alpha_d$ and α in (6.3) are bounded from above by $\widetilde{O}(d^5(d+\tau))$. The proof is almost the same as in the univariate real setting [53, Theorem 23], thus we only provide the main ingredients and skip some technical details.

From Lemma 6.1.1, there exists a positive integer

$$N = \widetilde{O}(d^3(d+\tau)) \tag{6.6}$$

such that $f-\varepsilon$ is positive on $\mathscr{C}$, with $\varepsilon = \frac{1}{2^N}$. We define $m := 2d$ and

$$g := z^d(f-\varepsilon) = \bar{f}_d z^{2d} + \cdots + \bar{f}_1 z^{d+1} + (f_0 - \varepsilon)z^d + f_1 z^{d-1} + \cdots + f_d. \tag{6.7}$$

Note that g and $f-\varepsilon$ have the same roots, and that $\|g\|_\infty \leq \|f\|_\infty + \varepsilon$. Denote by $\zeta_1, \ldots, \zeta_m$ the (exact) complex roots of g and by $\zeta'_1, \ldots, \zeta'_m$ their approximations with a precision δ, so that $\zeta'_j = \zeta_j(1+e_j)$, where $|e_j| \leq 2^{-\delta}$ for $j = 1, \ldots, m$. We consider a new polynomial g' which is defined as follows:

$$g' := \bar{f}_d(z-\zeta'_1)\ldots(z-\zeta'_m).$$

The polynomial u defined in Line 8 satisfies $z^d u = g - g'$.

We now prove that at a precision δ, where

$$\delta = N + \log_2((2d+1)^2\|g\|_\infty) = \widetilde{O}(d^3(d+\tau)),$$

we ensure that the coefficients of u satisfy the stopping condition (6.4) of the algorithm. Here, one can take $N = Cd^3(d+\tau)$, for a large enough constant $C > 1$. One has

$$e := 2^{-\delta} < \frac{1}{\delta} < \frac{1}{Cd^3(d+\tau)} < \frac{1}{m(m+1)} < \frac{1}{m}. \tag{6.8}$$

Let j be in $\{0, 1, \ldots, d\}$. Using Vieta's formulas (see, e.g., [84, Ch.3, p.89]), we have

$$\sum_{1\le i_1<\cdots<i_j\le m} \zeta_{i_1}\cdots\zeta_{i_j} = (-1)^j \frac{g_{m-j}}{g_m} = (-1)^j \frac{g_{m-j}}{\bar{f}_d}. \tag{6.9}$$

Similarly, we have

$$\sum_{1\le i_1<\cdots<i_j\le m} \zeta'_{i_1}\cdots\zeta'_{i_j} = (-1)^j \frac{g'_{m-j}}{\bar{f}_d}. \tag{6.10}$$

We estimate an upper bound for the coefficient $\bar{u}_{d-j}$ of the polynomial u. Clearly, one has $\bar{u}_{d-j} = g_{m-j} - g'_{m-j}$. From (6.9) and (6.10), we see that

$$|\bar{u}_{d-j}| = |\bar{f}_d| \left| \sum_{1\le i_1<\cdots<i_{d+j}\le m} \left(\zeta_{i_1}\cdots\zeta_{i_{d+j}} - \zeta'_{i_1}\cdots\zeta'_{i_{d+j}}\right)\right| \tag{6.11}$$

$$= |\bar{f}_d| \left| \sum_{1\le i_1<\cdots<i_{d+j}\le m} \zeta_{i_1}\cdots\zeta_{i_{d+j}} \left(1 - (1+e_{i_1})\cdots(1+e_{i_{d+j}})\right)\right|. \tag{6.12}$$

Apply [33, Lemma 3.3] for $e_{i_1}, \ldots, e_{i_{d+j}}$, we get $(1+e_{i_1})\cdots(1+e_{i_{d+j}}) \le 1+\theta_{d+j}$ with

$$|\theta_{d+j}| \le \frac{(d+j)e}{1-(d+j)e} \le \frac{me}{1-me}.$$

Since (6.8), we have

$$(m+1)e - \frac{me}{1-me} = \frac{e(1-m(m+1)e)}{1-me} \ge 0.$$

This yields $\frac{me}{1-me} \le (m+1)e$. So, we can conclude that

$$\left|1 - (1+e_{i_1})\cdots(1+e_{i_{d+j}})\right| \le |\theta_{d+j}| \le (m+1)e.$$

From the above presentation of $|\bar{u}_{d-j}|$ in (6.11) and (6.9), we obtain the following estimates:

$$|\bar{u}_{d-j}| \le |\bar{f}_d| \frac{|g_{m-j}|}{|\bar{f}_d|}(m+1)e \le \|g\|_\infty (m+1)e. \tag{6.13}$$

The conclusion holds for $j = 0, \ldots, d$ so one has

$$2\sum_{k=1}^{d} |u_k| - u_0 = 2\sum_{j=0}^{d-1} |\bar{u}_{d-j}| - u_0 \le e(m+1)^2\|g\|_\infty \le e(m+1)^2\|g\|_\infty.$$

It follows from $\delta = N + \log_2\left((2d+1)^2\|g\|_\infty\right)$ that $e(2d+1)^2\|g\|_\infty = \varepsilon$. Therefore, $\varepsilon > 2\sum_{k=1}^{d}|u_k| - u_0$ holds when $\delta = \widetilde{O}(d^3(d+\tau))$.

We choose $e_j = e = 2^{-\delta}$ and $z'_j = z_j(1+2^{-\delta})$. This implies that

$$|\bar{u}_{d-j}| = |\bar{f}_d||1-(1+2^{-\delta})^{d+j}|,$$

for all $j = 0,\ldots,d$. It follows from $\mathrm{ht}(\bar{f}_d) \leq \tau$, $\mathrm{ht}(\delta) = \widetilde{O}(d^3(d+\tau))$, and $\mathrm{ht}(\varepsilon) = \widetilde{O}(d^3(d+\tau))$, that

$$\mathrm{ht}(\bar{u}_{d-j}) = \widetilde{O}(d^3(d+\tau) + (d+j)d^3(d+\tau)) = \widetilde{O}(d^4(d+\tau)).$$

The maximal bitsize of the coefficients of u is bounded from above by $\widetilde{O}(d^4(d+\tau))$.

We now estimate the bit complexity of the coefficient α in (6.3). From Line 8 in `csos1`, one has $\alpha = \frac{f_0-\varepsilon}{F_0}$, where $F_0 = g'_d$ is the constant term of F. Clearly, $\mathrm{ht}(f_0-\varepsilon) = \widetilde{O}(d^3(d+\tau))$.

Let $\{\alpha_1,\ldots,\alpha_d,\frac{1}{\bar{\alpha}_1},\ldots,\frac{1}{\bar{\alpha}_d}\}$ be the approximate roots of the polynomial $z^d(f-\varepsilon)$. By applying Lemma 4.6.1 to the polynomial obtained by multiplying $z^d(f-\varepsilon)$ with the least common multiple of its coefficients, we require an accuracy of at least $\widetilde{O}(Nd)$ to compute distinct approximations of its roots in the worst case. Since (6.6), the last bound becomes $\widetilde{O}(d^4(d+\tau))$.

Because of (6.10), one has

$$g'_d = (-1)^d\bar{f}_d \sum_{1\leq i_1<\cdots<i_d\leq m} \zeta'_{i_1}\cdots\zeta'_{i_d}, \tag{6.14}$$

where $\zeta'_j \in \{\alpha_1,\ldots,\alpha_d,\frac{1}{\bar{\alpha}_1},\ldots,\frac{1}{\bar{\alpha}_d}\}$. Since $\mathrm{ht}(\alpha_i) = \widetilde{O}(d^4(d+\tau))$, from (6.14) we have the following estimates

$$\mathrm{ht}(g'_d) \leq d\,\mathrm{ht}(\alpha_i) + \log_2\binom{2d}{d} \leq d\,\mathrm{ht}(\alpha_i) + d\log_2(d+1) = \widetilde{O}(d^5(d+\tau)).$$

Finally, the maximal bitsize of the u_k's and α is bounded from above by $\widetilde{O}(d^5(d+\tau))$, as claimed. □

Theorem 6.1.4. *Assume that $f \in \mathscr{H}(\mathbb{Z})[z]$ be positive on $\mathscr{C}$ of degree d and bitsize τ and that f has no multiple roots. On input f, `csos1` computes an SOHS decomposition of f using at most $\widetilde{O}\left(d^6(d+\tau)\right)$ bit operations.*

Proof. The algorithm includes two steps. We consider the first step, checking that the polynomial g defined in (6.7) in the previous proof has no real root on the unit circle. Let ε be given as in Lemma 6.1.1 with

$$\mathrm{ht}(\varepsilon) = \widetilde{O}(d^3(d+\tau)).$$

Relying on Sylvester–Habicht sequences [47, Corollary 5.2], we can check it by using

$$O((2d)^2\,\mathrm{ht}(\varepsilon)) = \widetilde{O}(d^5(d+\tau))$$

boolean operations. In the second step, we compute approximate complex roots of g and check the condition at Line 10. It follows from [59, Theorem 4] that isolating disks of radius less than $2^{-\delta}$ for all complex roots of $g(z)$ can be computed in

$$\widetilde{O}(d^3 + d^2\,\mathrm{ht}(\varepsilon) + d\delta)) = \widetilde{O}(d^6(d+\tau))$$

boolean operations. The computation of all u_k has a negligible cost with respect to the computation of the complex roots. Therefore, we conclude that `csos1` runs in $\widetilde{O}(d^6(d+\tau))$ boolean operations. □

6.2 Algorithm based on complex SDP solving

This section states and analyzes another perturbation-compensation algorithm, named `csos2`, to compute an SOHS decomposition of a trigonometric polynomial being positive on $\mathscr{C}$. In the algorithm, the approximate SOHS decomposition for the perturbation is computed by using complex SDP solving. It can be viewed as the adaptation of the procedure `intsos` (stated and analyzed in [52, § 3]) to the complex univariate setting.

Let I_d stands for the identity matrix of size d. Given $f \in \mathscr{H}[z]$ of degree d, recall that a Hermitian matrix $Q \in \mathbb{C}^{(d+1)\times(d+1)}$ is a Gram matrix associated with f if $f = v_d^\star\, Q\, v_d$, where

$$v_d := (1, z, \ldots, z^d)^T$$

contains the canonical basis for polynomials of degree d in z. By [22, Theorem 2.5], f is positive on $\mathscr{C}$ if and only if there exists a positive definite Gram matrix associated to f.

6.2.1 Description and correctness

Description. The input of Algorithm `csos2` includes a polynomial $f \in \mathscr{H}(\mathbb{Z})[z]$ of degree d which is positive on $\mathscr{C}$. The outputs are $\varepsilon \in \mathbb{Q}_+$, a list of Gaussian numbers $[u_0, u_1, \ldots, u_d]$, and a list of polynomials $[s_1, \ldots, s_d]$ in $\mathbb{Q}[\boldsymbol{i}][z]$ providing an SOHS decomposition of f as follows

$$f = \Big(\varepsilon + u_0 - 2\sum_{k=1}^{d}|u_k|\Big) + \sum_{k=1}^{d}|u_k|\Big(z^k + \frac{u_k}{|u_k|}\Big)\Big(z^{-k} + \frac{\bar{u}_k}{|u_k|}\Big) + \sum_{k=0}^{d} s_k^\star s_k. \tag{6.15}$$

The first while loop of `csos2` (Lines 2–3) is exactly the same as in `csos1` to obtain $\varepsilon \in \mathbb{Q}_+$ such that $f - \varepsilon$ is positive on $\mathscr{C}$. Then, instead of using root isolation as in `csos1`, `csos2`

Algorithm 6 Computing SOHS decomposition based on complex SDP solving

`csos2` := proc(f)

Input: $f \in \mathscr{H}(\mathbb{Z})[z]$ positive on $\mathscr{C}$ of degree d

Output: $\varepsilon \in \mathbb{Q}_+$, $[u_0, u_1, \ldots, u_d]$ in $\mathbb{Q}[\boldsymbol{i}]$, $[s_0, \ldots, s_d]$ in $\mathbb{Q}[\boldsymbol{i}][z]$ providing an SOHS decomposition of f as in (6.15)

1: Set $\delta := 1$, $R := 1$, $\delta_c = 1$, $\varepsilon := 1$ and compute $p := f(x + \boldsymbol{i}y)$
2: **while** `hasrealrootoncircle`$(p - \varepsilon)$ **do** $\varepsilon := \frac{\varepsilon}{2}$
3: **done**
4: boo := false
5: **while** not boo **do**
6: Compute $(\tilde{Q}, \tilde{\lambda}) :=$ `sdp`$(f - \varepsilon, \delta, R)$
7: Compute $[s_0, \ldots, s_d] :=$ `cholesky`$(\tilde{Q}, \tilde{\lambda}, \delta_c)$ ▷ $f - \varepsilon \simeq \sum_{k=0}^{d} s_k^\star s_k$
8: Compute $u := (f - \varepsilon) - \sum_{k=0}^{d} s_k^\star s_k$, $[u_0, u_1, \ldots, u_d] :=$ `coeffs`(u)
9: **if** $\varepsilon > 2\sum_{k=1}^{d} |u_k| - u_0$ **then** boo := true
10: **else** $\delta := 2\delta$, $R := 2R$, $\delta_c := 2\delta_c$
11: **end**
12: **done**
13: **return** $\varepsilon, [u_0, u_1, \ldots, u_d], [s_0, \ldots, s_d]$

relies on complex SDP (Line 6) and Cholesky's decomposition (Line 7) to compute an approximate SOHS decomposition of the perturbed polynomial.

With $f - \varepsilon$, δ, and R, the `sdp` function calls an SDP solver to compute a rational approximation $\tilde{Q}$, which is positive definite, of a Gram matrix associated to $f - \varepsilon$ and a rational approximation $\tilde{\lambda}$ of its smallest eigenvalue. As in [52], we analyze the complexity of this procedure by assuming that `sdp` relies on the ellipsoid algorithm [31], running in polynomial-time within a given accuracy δ and a radius bound R on the Frobenius norm of $\tilde{Q}$. Its outputs are obtained by solving the following complex SDP:

$$\begin{aligned} \lambda_{\min} = \max_{Q,\lambda}\ & \lambda \\ \text{s.t.}\ & \operatorname{tr}(\Theta_k Q) = f_k - (\varepsilon \cdot 1_{k=0}),\ k = -d, \ldots, d\,, \\ & Q \succeq \lambda I_{d+1}\,, \lambda \geq 0\,,\ Q \in \mathbb{C}^{(d+1)\times(d+1)}\,, \end{aligned} \tag{6.16}$$

where Θ_k is the elementary Toeplitz matrix with ones on the $k-$th diagonal and zeros elsewhere, i.e. $1_{k=0} = 1$ if $k = 0$ and 0 otherwise. The equality constraints of the SDP (6.16) correspond to the relation $f - \varepsilon = v_d^\star Q\, v_d$. This SDP (corresponding to the SDP (2.14) in [22]) computes the Gram matrix associated to f with the largest minimal eigenvalue.

The `cholesky` function computes first an approximate Cholesky's decomposition $LL^\star$

of $\tilde{Q}$ with precision δ_c and provides as output a list of polynomials $[s_0, \ldots, s_d]$ in $\mathbb{Q}[i][z]$, where s_k is the inner product of the $(k+1)$-th row of L by v_d. One would expect to have $f - \varepsilon = \sum_{k=0}^{d} s_k^\star s_k$ after using exact SDP and Cholesky's decomposition. Since the SDP solver is not exact, we have to consider the difference $u = f - \varepsilon - \sum_{k=0}^{d} s_k^\star s_k$ and proceed exactly as in `csos1` to obtain an exact SOHS decomposition.

Remark 6.2.1. According to [22, Remark 2.8], we can convert the complex SDP (6.16) to a real one. Indeed, the variable matrix Q can be written as $Q = Q_{\mathrm{re}} + \boldsymbol{i} Q_{\mathrm{im}}$, where Q_{re} and Q_{im} are real matrices. Hence, the constraint $\mathrm{tr}(\Theta_k Q) = f_k - \varepsilon \cdot 1_{k=0}$ can be replaced by two constraints, $\mathrm{tr}(\Theta_k Q_{\mathrm{re}}) = \mathrm{re}(f_k - \varepsilon \cdot 1_{k=0})$ and $\mathrm{tr}(\Theta_k Q_{\mathrm{im}}) = \mathrm{im}(f_k - \varepsilon \cdot 1_{k=0})$. Moreover, the condition $Q \succeq 0$ is equivalent to

$$\begin{bmatrix} Q_{\mathrm{re}} & -Q_{\mathrm{im}} \\ Q_{\mathrm{im}} & Q_{\mathrm{re}} \end{bmatrix} \succeq 0.$$

Example 6.2.1. Consider the polynomial $f = 5 + (1+\boldsymbol{i})z^{-1} + (1-\boldsymbol{i})z$ that is positive on $\mathscr{C}$. We provide an SOHS decomposition of f by using `csos2`. Similarly, as in Example 6.1.1, with $\varepsilon = 1$, we can check that $p - \varepsilon$ is positive on $\mathscr{C}$. With precision $\delta = 2^{64}$, we compute the complex approximation matrix $\tilde{Q}$. Here, we use the UD decomposition of $\tilde{Q}$. We have $\tilde{Q} = UDU^\star$, where

$$\tilde{Q} = \begin{bmatrix} \frac{76207117}{82595451} - \frac{84775740}{90917777}\boldsymbol{i} & 1-\boldsymbol{i} \\ 1-\boldsymbol{i} & \frac{90917777}{42387870} \end{bmatrix},$$

and

$$U = \begin{bmatrix} 1 & \frac{42387870}{90917777} - \frac{42387870}{90917777}\boldsymbol{i} \\ 0 & 1 \end{bmatrix}, \quad D = \begin{bmatrix} \frac{76207117}{82595451} & 0 \\ 0 & \frac{90917777}{42387870} \end{bmatrix}.$$

We have $u_0 = 0, u_1 = \frac{-1781161}{35367472262109859495610}$, $s_0 = 1$, and $s_1 = \left(\frac{42387870}{90917777} - \frac{42387870}{90917777}\boldsymbol{i}\right) + \frac{1}{z}$. Clearly, $\varepsilon = 1 > 2 \times \frac{1781161}{35367472262109859495610}$, so the condition in Line 9 is satisfied. Then, f has an exact SOHS decomposition as follows:

$$\begin{aligned} f = \left(1 - 2 \times \tfrac{1781161}{35367472262109859495610}\right) &+ \tfrac{76207117}{82595451} \times 1 \times 1 \\ &+ \tfrac{90917777}{42387870} \times \left(\tfrac{42387870}{90917777} - \tfrac{42387870}{90917777}\boldsymbol{i} + \tfrac{1}{z}\right) \times \left(\tfrac{42387870}{90917777} + \tfrac{42387870}{90917777}\boldsymbol{i} + z\right). \end{aligned}$$

The lemma below prepares the correctness and bit complexity analysis of `csos2`.

Lemma 6.2.1. *Let $f \in \mathscr{H}(\mathbb{Z})[z]$ be positive on $\mathscr{C}$ of degree d and bitsize τ. Assume that Q is a positive definite Gram matrix associated to f. Then, there exist $\varepsilon \in \mathbb{Q}_+$ of bitsize $\widetilde{O}(d^3(d+\tau))$ such that $f - \varepsilon$ is positive on $\mathscr{C}$, δ of bitsize $\widetilde{O}(d^3(d+\tau))$ and R of bitsize $O(\mathrm{ht}(d) + \tau)$ such that $Q - \frac{\varepsilon}{d+1} I_{d+1}$ is a Gram matrix associated to $f - \varepsilon$ with*

$$Q - \frac{\varepsilon}{d+1} I_{d+1} \succ 2^{-\delta} I_{d+1} \quad \text{and} \quad \sqrt{\mathrm{tr}((Q - \varepsilon I_{d+1})^2)} \leq R. \tag{6.17}$$

Proof. By Lemma 6.1.1, there is a positive integer N and $\varepsilon = 2^{-N}$ with $N = \widetilde{O}(d^3(d+\tau))$ such that $f - 3\varepsilon/2 > 0$ on $\mathscr{C}$. Let

$$\delta := \lceil N + 1 + \log_2(d+1) \rceil = \widetilde{O}(d^3(d+\tau))$$

so that $2^{-\delta} \leq \frac{\varepsilon}{2(d+1)}$. Note that $v_d^\star v_d = d+1$. Thus, we have the following representation:

$$f - \varepsilon = v_d^\star \left(Q - \frac{\varepsilon}{d+1} I_{d+1} \right) v_d.$$

Since $f - \varepsilon - 2^{-\delta}(d+1) \geq f - 3\varepsilon/2$ is positive over the unit circle, the Gram matrix associated to $f - \varepsilon - 2^{-\delta}(d+1)$ is positive definite. Specifically, we have that

$$\left(Q - \frac{\varepsilon}{d+1} I_{d+1} \right) - 2^{-\delta} I_{d+1} \succ 0.$$

Let $R := f_0\sqrt{(d+1)}$. It follows that R is of bitsize $O(\mathrm{ht}(d) + \tau)$. Note that the equality constraint of SDP (6.16) with $k = 0$ reads $\mathrm{tr}(Q) = f_0 - \varepsilon \leq f_0$. The maximal eigenvalue of Q is less than f_0 and

$$\mathrm{tr}((Q - \varepsilon I_{d+1})^2) \leq \mathrm{tr}(Q^2) \leq (d+1) f_0^2 = R^2.$$

This is the last desired inequality. □

Theorem 6.2.2. *Let $f \in \mathscr{H}(\mathbb{Z})[z]$ be positive on $\mathscr{C}$ of degree d. On input f, Algorithm* `csos2` *terminates and outputs an SOHS decomposition of f as in* (6.15).

Proof. Since $f - \varepsilon$ is positive on $\mathscr{C}$, according to [22, Theorem 2.5], SDP (6.16) always has a strictly feasible solution for precision parameters (δ, R) with bitsizes as in Lemma 6.2.1 and the `sdp` function returns an approximate Gram matrix $\tilde{Q}$ associated to $f - \varepsilon$ such that $\tilde{Q} \succeq 2^{-\delta} I_{d+1}$ and $\mathrm{tr}(Q^2) \leq R^2$ as in (6.17). In particular, we obtain a rational approximation $\tilde{\lambda} \geq 2^{-\delta}$ of the smallest eigenvalue of $\tilde{Q}$. Let δ_c be the smallest integer such that

$$2^{-\delta_c} < \frac{2^{-\delta}}{(d+1)^2 + d + 1 + d2^{-\delta}}.$$

Since the following function in the variable t,

$$t \mapsto \frac{t}{(d+1)^2 + d + 1 + dt},$$

increases on $[0, +\infty)$ and $\tilde{\lambda} \geq 2^{-\delta}$, the inequality (4.12) holds. According to Lemma 4.5.1, at Line 6 we compute an approximate Cholesky decomposition of $\tilde{Q}$ by using the `cholesky` procedure. We obtain a nonsingular factor $\tilde{Q} = LL^\star$, where L has Gaussian entries.

We denote $s_k := L^\star v_d$ and consider the difference polynomial

$$u = (f - \varepsilon) - \textstyle\sum_{k=0}^{d} s_k^\star s_k.$$

The second while loop (Lines 5–12) terminates when the stopping condition (6.4) is fulfilled. This condition holds if $|u_k| \leq \frac{\varepsilon}{2d+1}$, for all $k = 0, \ldots, d$. We prove that these last conditions hold when δ and δ_c are both large enough. Indeed, we have

$$u_k = f_k - \varepsilon_k - \Big(\sum_{j=0}^{d} s_j^\star s_j\Big)_k, \; k = -d, \ldots, d, \tag{6.18}$$

where $\varepsilon_0 = \varepsilon$, $\varepsilon_k = 0$ for $k \neq 0$, and $(\sum_{j=0}^{d} s_j^\star s_j)_k$ is the coefficient of monomial z^k in the involved polynomial. Recall that the positive definite matrix $\tilde{Q}$ computed by the SDP solver is an approximation of the Gram matrix associated to $f - \varepsilon$. With the precision δ, from (6.18) and $\tilde{Q} \succeq 2^{-\delta} I$, we see that

$$|f_k - \varepsilon_k - \operatorname{tr}(\Theta_k \tilde{Q})| = |f_k - \varepsilon_k - \sum_{i+j=k} \tilde{Q}_{ij}| \leq 2^{-\delta}.$$

Furthermore, from (4.11), the approximate Cholesky decomposition $LL^\star$ of $\tilde{Q}$ performed at precision δ satisfies $LL^\star = \tilde{Q} + H$ and

$$|H_{ij}| \leq \frac{(d+2)2^{-\delta_c}\sqrt{|\tilde{Q}_{ii}\tilde{Q}_{jj}|}}{1 - (d+2)2^{\delta_c}},$$

for all i, j in $\{-d, \ldots, d\}$. Applying the Cauchy-Schwarz inequality for the trace function, we see that

$$\sum_{k=-d}^{d} |\tilde{Q}_{kk}| = \operatorname{tr}(\tilde{Q}) \leq \sqrt{\operatorname{tr}(\tilde{Q}^2)}\sqrt{\operatorname{tr}(I)} \leq R\sqrt{d+1}.$$

So, for each k in $\{-d, \ldots, d\}$, we have

$$\Big|\sum_{i+j=k} \sqrt{\tilde{Q}_{ii}\tilde{Q}_{jj}}\Big| \leq \sum_{i+j=k} \frac{\tilde{Q}_{ii} + \tilde{Q}_{jj}}{2} \leq \operatorname{tr}(\tilde{Q}) \leq R\sqrt{d+1}. \tag{6.19}$$

Therefore, we have

$$\Big|\sum_{i+j=k} \tilde{Q}_{ij} - \Big(\sum_{j=0}^{d} s_j^\star s_j\Big)_k\Big| = \Big|\sum_{i+j=k} \tilde{Q}_{ij} - \sum_{i+j=k} (LL^\star)_k\Big| = \Big|\sum_{i+j=k} H_{ij}\Big|.$$

It follows from (4.11) and (6.19) that the last number is bounded by

$$\Big|\sum_{i+j=k} \tilde{Q}_{ij} - \Big(\sum_{j=0}^{d} s_j^\star s_j\Big)_k\Big| \leq \frac{(d+2)2^{-\delta_c}}{1 - (d+2)2^{-\delta_c}} \sum_{i+j=k} \sqrt{|\tilde{Q}_{ii}\tilde{G}_{jj}|} \leq \frac{R\sqrt{d+1}(d+2)2^{-\delta_c}}{1 - (d+2)2^{-\delta_c}}.$$

Take the smallest δ such that $2^{-\delta} \leq \frac{\varepsilon}{2(2d+1)} = \frac{1}{(2d+1)2^{N+1}}$ as well as the smallest δ_c such that

$$\frac{R\sqrt{d+1}(d+2)2^{-\delta_c}}{1 - (d+2)2^{-\delta_c}} \leq \frac{\varepsilon}{2(2d+1)},$$

i.e., $\delta = \lceil N + 1 + \log_2(2d+1) \rceil$ and

$$\delta_c = \lceil \log_2 R + \log_2(d+2) + \log_2(2^{N+1}(2d+1)^{3/2} + 1) \rceil.$$

From (6.18) and above inequalities, we obtain the following estimates:

$$\begin{aligned} |u_k| &\leq \Big| f_k - \varepsilon_k - \sum_{i+j=k} \tilde{Q}_{ij} \Big| + \Big| \sum_{i+j=k} \tilde{Q}_{ij} - \Big(\sum_{j=0}^{d} s_j^\star s_j \Big)_k \Big| \\ &\leq \frac{\varepsilon}{2(2d+1)} + \frac{\varepsilon}{2(2d+1)} = \frac{\varepsilon}{2d+1}. \end{aligned}$$

This guarantees that the second while loop terminates for large enough δ and δ_c, with bitsizes $\widetilde{O}(d^3(d+\tau))$. □

6.2.2 Bit complexity analysis

Theorem 6.2.3. *Let $f \in \mathscr{H}(\mathbb{Z})[z]$ be positive on $\mathscr{C}$ of degree d and coefficients of maximum bitsize τ. On input f, `csos2` computes an SOHS decomposition of f with (modulus of) Gaussian coefficients using at most $\widetilde{O}(d^{13}(d+\tau)^2)$ bit operations. In addition, the maximal bitsize of the output coefficients is bounded from above by $\widetilde{O}(d^6(\tau+d))$.*

Proof. Firstly, we prove that Algorithm `csos2` runs in $\widetilde{O}(d^{13}(d+\tau)^2)$ boolean operations. Assume that ε, δ, R and δ_c are given as above so that, before terminating, `csos2` performs a single iteration in each while loop. From Lemma 6.2.1, the bitsize of R is $O(\mathrm{ht}(d) + \tau)$ and the bitsize of each $\varepsilon, \delta, \delta_c$ is upper bounded by $\widetilde{O}(d^3(d+\tau))$.

To investigate the computational cost of the call to `sdp` at Line 6, we rely on the bit complexity analysis of the ellipsoid method [69]. Denote $n_{\mathrm{sdp}} = d+1$ by the size of $\tilde{Q}$ and $m_{\mathrm{sdp}} = 2d+1$ by the number of affine constraints of the SDP (6.16). According to Theorem 4.5.2, SDP (6.16) is solved in

$$O(n_{\mathrm{sdp}}^4 \log_2(2^\tau n_{\mathrm{sdp}} R\, 2^\delta))$$

iterations of the ellipsoid method, where each iteration requires

$$O(n_{\mathrm{sdp}}^2(m_{\mathrm{sdp}} + n_{\mathrm{sdp}}))$$

arithmetic operations over $\log_2(2^\tau n_{\mathrm{sdp}} R\, 2^\delta)$-bit numbers. We obtain the following estimates:

$$O(n_{\mathrm{sdp}}^4 \log_2(2^\tau n_{\mathrm{sdp}} R\, 2^\delta)) = \widetilde{O}(d^7(d+\tau)), \;\; O(n_{\mathrm{sdp}}^2(m_{\mathrm{sdp}} + n_{\mathrm{sdp}})) = O(d^3),$$

and

$$O(\log_2(2^\tau n_{\mathrm{sdp}} R\, 2^\delta)) = \widetilde{O}(d^3(d+\tau)).$$

Therefore, to compute the approximate Gram matrix $\tilde{Q}$, the ellipsoid algorithm runs in boolean time $\widetilde{O}(d^{13}(d+\tau)^2)$.

Next, we compute the cost of calling `cholesky` in Line 7. Note that Cholesky's decomposition is performed in $O(n_{\text{sdp}}^3)$ arithmetic operations over δ_c-bit numbers. Because $\delta_c = \widetilde{O}(d^3(d+\tau))$ and $n_{\text{sdp}} = d+1$, `cholesky` runs in boolean time $\widetilde{O}(d^6(d+\tau))$. The other elementary arithmetic operations of Algorithm `csos2` have a negligible cost with respect to the `sdp` procedure. Hence, the algorithm runs in boolean time $\widetilde{O}(d^{13}(d+\tau)^2)$.

The bitsize of the output coefficients is upper bounded by the output bitsize of the Cholesky's decomposition of the matrix $\widetilde{Q}$, that is $O(\delta_c(d+1)^3) = \widetilde{O}(d^6(d+\tau))$. □

6.3 Algorithm based on rounding-projection technique

In this chapter, we introduce Algorithm `csos3` which is an adaptation of the rounding-projection method by Peyrl and Parrilo, stated in [67] and analyzed in [52, § 3.4], and investigate its bit complexity.

6.3.1 Description

The input of `csos3` is a polynomial $f \in \mathscr{H}(\mathbb{Z})[z]$ of degree d which is positive over $\mathscr{C}$. The outputs consist of a list $[c_0, \ldots, c_d] \subset \mathbb{Q}_+$ and a list of polynomials $[s_0, \ldots, s_d]$ in $\mathbb{Q}[i][z]$ that provide an SOHS decomposition of f, namely

$$f = \sum_{k=0}^{d} c_k s_k^\star s_k. \tag{6.20}$$

As in `csos2`, the first while loop from Lines 3–8 provides an approximate Gram matrix $\tilde{Q}$ associated to f and an approximation $\tilde{\lambda}$ of its smallest eigenvalue. In Line 11, we round the matrix $\tilde{Q}$ up to precision $\hat{\delta}$ to obtain a matrix $\hat{Q}$, with Gaussian coefficient entries. The for loop from Line 12 to Line 15 is the projection step to ensure that the equality constraints of SDP (6.16) hold exactly. Then, we compute the $LDL^\star$ decomposition of Q. The list $[c_0, \ldots, c_d]$ is the list of coefficients of the diagonal matrix D and each s_k is the inner product of the $(k+1)$-th row of L with the vector v_d of all monomials up to degree d. If all c_k's are positive rationals and all polynomials s_k have Gaussian coefficients, then the second while loop ends. Otherwise, we increase the precision $\hat{\delta}$.

6.3.2 Correctness and bit complexity analysis

Theorem 6.3.1. *For $f \in \mathscr{H}(\mathbb{Z})[z]$ positive on $\mathscr{C}$ of degree d and bitsize τ, there exist precisions δ, $\hat{\delta}$ upper bounded by $\widetilde{O}(d^3(d+\tau))$, and a radius bound R upper bounded by $O(\mathrm{ht}(d)+\tau)$*

Algorithm 7 Computing SOHS decomposition based on rounding-projection technique

`csos3` := proc(f)

Input: $f \in \mathscr{H}(\mathbb{Z})[z]$ positive on $\mathscr{C}$ of degree d

Output: $[c_0, \ldots, c_d] \subset \mathbb{Q}_+$ and $[s_0, \ldots, s_d] \subset \mathbb{Q}[\boldsymbol{i}][z]$ providing an SOHS decomposition of f as in (6.20)

Set $\delta := 1$, $R := 1$, $\delta_c = 1$, $\hat{\delta} := 1$
Set boo := false
while not boo **do**
  Compute $(\tilde{Q}, \tilde{\lambda}) := \mathtt{sdp}(f, \delta, R)$
  if $\tilde{\lambda} > 0$ **then** boo := true
  else $\delta := 2\delta$, $R := 2R$
  end
done
Set boo := false
while not boo **do**
  Compute $\hat{Q} := \mathtt{round}(\tilde{Q}, \hat{\delta})$
  for $j \in \{0, \ldots, d\}, k \in \{0, \ldots, j\}$ **do**
    Compute $Q_{j,j-k} := \hat{Q}_{j,j-k} - \frac{1}{d-k+1}(\sum_{i=k}^{d} \hat{Q}_{i,i-k} - f_k)$
    Compute $Q_{j-k,j} := Q^{\star}_{j,j-k}$ ▷ $Q = Q^{\star}$
  done
  Compute $[c_0, \ldots, c_d; s_0, \ldots, s_d] := \mathtt{ldl}(Q)$ ▷ $f = \sum_{k=0}^{d} c_k s_k^{\star} s_k$
  if $c_0, \ldots, c_d \in \mathbb{Q}_+, s_0, \ldots, s_d \in \mathbb{Q}[\boldsymbol{i}][z]$ **then** boo := true
  else $\hat{\delta} := 2\hat{\delta}$
  end
done
return $[c_0, \ldots, c_d]$, $[s_0, \ldots, s_d]$

such that Algorithm `csos3` *terminates and outputs an SOHS decomposition of* f.

Proof. After the second while loop (Lines 10–20), we obtain the positive definite matrix Q associated to f with the smallest eigenvalue λ. Let $N \in \mathbb{N}$ be the smallest integer satisfying $2^{-N} \leq \lambda$. From Lemma 6.2.1, the bitsize of N is $\widetilde{O}(d^3(d+\tau))$. The matrix $\hat{Q}$ is obtained after the rounding step at Line 11. The number δ_E stands for the distance between $\hat{Q}$ and Q which is defined as follows (all norms in a Euclidean space are equivalent):

$$2^{-\delta_E} := \sqrt{\sum_{i,j}(\hat{Q}_{ij} - Q_{ij})^2}.$$

Since $\hat{\delta}$ is the precision of rounding, one has $|\hat{Q}_{ij} - \tilde{Q}_{ij}| \leq 2^{-\hat{\delta}}$ for all i, j in $\{0, \ldots, d\}$. As in the proof of Theorem 6.2.2, at the SDP precision δ, one has $\tilde{Q} \succeq 2^{-\delta} I$. By [67, Proposition 8], `csos3` terminates and outputs such matrix Q together with an SOHS decomposition of f if

$$2^{-\hat{\delta}} + 2^{-\delta_E} \leq 2^{-N}. \tag{6.21}$$

By repeating the argument in the proof of [52, Theorem 13], we can conclude that $\mathrm{ht}(N) = \widetilde{O}(d^3(d+\tau))$ ensures the condition (6.21).

Similar to the argument in the proof of Lemma 6.2.1, the `sdp` function is successful if R is $O(\mathrm{ht}(d) + \tau)$. □

As emphasized in [52, § 3.4], it turns out that the two algorithms `csos2` and `csos3` have the same bit complexity. We omit any technicalities as the proof is almost the same as [52, Theorem 12].

Theorem 6.3.2. *Let* $f \in \mathscr{H}(\mathbb{Z})[z]$ *be positive on* $\mathscr{C}$ *of degree* d *and coefficients of maximum bitsize* τ. *On input* f, `csos3` *outputs an SOHS decomposition of* f *with Gaussian coefficients using at most* $\widetilde{O}(d^{13}(d+\tau)^2)$ *bit operations. Additionally, the maximal bitsize of the output coefficients is bounded from above by* $\widetilde{O}(d^6(d+\tau))$.

Proof. For $f \in \mathscr{H}(\mathbb{Z})[z]$ positive over $\mathscr{C}$ with degree d and maximal bitsize τ, there exist δ, $\hat{\delta}$ with bitsizes upper bounded by $\widetilde{O}(d^3(d+\tau))$, and R with bitsize upper bounded by $O(\mathrm{ht}(d) + \tau)$ such that Algorithm `csos3` outputs an SOHS decomposition of f. The bitsize of the output coefficients is upper bounded by the output bitsize of the LDL^T decomposition of the matrix Q, that is

$$O(\hat{\delta}(d+1)^3) = \widetilde{O}(d^6(d+\tau)).$$

The running time $\widetilde{O}(d^{13}(d+\tau)^2)$ is estimated as the running time of `csos2`. □

Example 6.3.1. Consider the polynomial $f = 5 + (1+\boldsymbol{i})z^{-1} + (1-\boldsymbol{i})z$ that is positive on $\mathscr{C}$. We provide an SOHS decomposition of f using `csos3`. With precision $\delta = 2^{64}$, we obtain the approximate matrix $\hat{Q}$,

$$\hat{Q} = \begin{bmatrix} 1.8551 & 1-\boldsymbol{i} \\ 1+\boldsymbol{i} & 2.1449 \end{bmatrix}.$$

After the rounding and projection steps (Lines 12–15), we have

$$Q = \begin{bmatrix} \frac{4177311994322459}{2251799813685248} & 1-\boldsymbol{i} \\ 1+\boldsymbol{i} & \frac{4829887260418533}{2251799813685248} \end{bmatrix}.$$

Computing the LDL factorization of Q, we obtain

$$L = \begin{bmatrix} 1 & 0 \\ \frac{2251799813685248}{4177311994322459}(1+\boldsymbol{i}) & 1 \end{bmatrix}, \quad D = \begin{bmatrix} \frac{4177311994322459}{2251799813685248} & 0 \\ & \frac{10034741182345744764918476089639}{9406470370520464927385643384832} \end{bmatrix}.$$

Hence, we have an SOHS decomposition of f as follows:

$$f = \tfrac{4177311994322459}{2251799813685248} \times \left(1 + \tfrac{2251799813685248}{4177311994322459}(1+\boldsymbol{i}) \times \tfrac{1}{z}\right) \times \left(1 + \tfrac{2251799813685248}{4177311994322459}(1-\boldsymbol{i}) \times z\right) + \tfrac{10034741182345744764918476089639}{9406470370520464927385643384832} \times 1 \times 1.$$

6.4 Practical experiments

This section is dedicated to experimental results for our three certification algorithms, `csos1`, `csos2` and `csos3`, stated in Section 6.1, 6.2 and 6.3, respectively. Firstly, we compare their performance towards certifying positivity on the unit circle for trigonometric polynomials with Gaussian coefficients. Next, we describe how to extend our third algorithm, `csos3`, to design a finite impulse response (FIR) filter in a certified fashion.

Our code is implemented in JULIA, available online at polsys.lip6.fr/∼hieu/csos.zip, and the results are obtained on an Intel Xeon 6244 CPU (3.6GHz) with 1.5 TB of RAM.

In `csos1` and `csos2`, we compute ε such that $f - \varepsilon$ is positive on $\mathscr{C}$ in Lines 2–3 by using MSOLVE [12] within the Julia library `GroebnerBasis.jl`. The corresponding running time is denoted by t_ε. We denote by t_u the running time spent to compute the difference polynomial u and to perform the comparison involving its coefficients and ε. In the algorithm `csos1`, we compute approximate roots of $f - \varepsilon$ with the arbitrary-precision library `PolynomialRoots.jl` [82]. In `csos2` and `csos3`, we model SDP (6.16) though JUMP [23] and solve it with Mosek [1]. Exact arithmetic is performed with the CALCIUM library available in `Nemo.jl`.

6.4.1 Positivity verification

We consider a family of trigonometric polynomials with Gaussian integer coefficients

$$f_d = 10d + \sum_{k=1}^{d}((1-\boldsymbol{i})z^{-k} + (1+\boldsymbol{i})z^k),$$

for $d \in \{50, 100, 150, 200, 250\}$. On the unit circle $\mathscr{C}$, each such f_d is positive since $z^{-k} + z^k \geq -2$. We provide certificates of positivity for these polynomials by computing exact SOHS decompositions though `csos1`, `csos2` and `csos3`.

For `csos1`, we use a precision $\delta = 64$ (bits) to isolate complex roots. As a side note, we were not able to use arbitrary-precision SDP solvers (e.g., SDPA-GMP) within `csos2` and `csos3`, because JuMP only allows us to rely on double floating-point arithmetic at the moment. The running times (in seconds) of the three algorithms are reported in Table 6.1. As expected from the theoretical bit complexity results from Theorem 6.1.3 and Theorem 6.2.3, Algorithm `csos1` performs better than `csos2` and `csos3`. Moreover, `csos2` is faster than `csos3` because of the fact that the latter algorithm requires the computation of an exact Cholesky's factorization. Even though `csos1` happens to be the best choice to verify the positivity of polynomials with known coefficients, the use of an SDP solver is mandatory to optimize over positive polynomials with unknown coefficients, as demonstrated in the next subsection.

	`csos1`			`csos2`			`csos3`
d	t_ε	t_u	total	t_ε	t_u	total	total
50	0.2	0.3	0.6	0.2	6.6	6.8	7.7
100	1.6	2.9	4.5	1.6	128	130	184
150	5.2	13	19	5.2	830	838	1460
200	24	26	51	24	3460	3485	7214
250	64	55	120	64	10553	10622	24852

Table 6.1: Performance of Algorithms `csos1`, `csos2`, and `csos3`

6.4.2 Design of a certified linear-phase FIR filter

This section is devoted to the design of a linear-phase Finite Impulse Response (FIR) filter. This boils down to solving an energy minimization problem. To obtain a certified filter, we first solve a semi-definite optimization problem (corresponding to SDP (5.12) from [22]) and transform the numerical output into an exact certificate via a projection procedure similar to the one used in `csos3`.

An FIR filter of order d is an univariate trigonometric polynomial with real coefficients

$$H(z) = \sum_{k=-d}^{d} h_k z^{-k}$$

where $h := [h_0, \ldots, h_d]$ be the coefficient vector of H. Since we work on the unit circle, we have $z = \exp(i\omega)$, for $\omega \in \mathbb{R}$, and we abuse notation by writing $H(\omega)$ instead of $H(z)$. The passband and stopband are $[0, \omega_p]$ and $[\omega_s, \pi]$ respectively, where ω_p, ω_s are given. The stopband energy of the FIR filter is

$$E_s = \frac{1}{\pi} \int_{\omega_s}^{\pi} |h(\omega)|^2 d\omega.$$

To design such a linear-phase filter, we minimize the stopband energy under modulus constraints involving two parameters γ_p, γ_s:

$$\begin{array}{ll} \min\limits_{H \in \mathscr{H}[z]} & E_s \\ \text{s.t.} & |H(\omega) - 1| \leq \gamma_p, \ \forall \omega \in [0, \omega_p], \\ & |H(\omega)| \leq \gamma_s, \qquad \forall \omega \in [\omega_s, \pi]\,. \end{array} \tag{6.22}$$

We will reformulate the above optimization problem to an SDP. To do so, we need to recall some notations. We denote $\Theta_k \in \mathbb{R}^{(d+1)\times(d+1)}$ by the elementary Toeplitz matrix with ones on the k-th diagonal and zeros elsewhere, for $k = 0, \ldots, d$. Denote by

$$C = \text{Toep}(c_0, \ldots, c_d)$$

the Toeplitz matrix with the first row $(c_0, \ldots, c_d)$, where

$$c_k = \begin{cases} 1 - \frac{\omega_s}{\pi}, & \text{if } k = 0, \\ -\frac{\sin k\omega_s}{k\pi}, & \text{if } k > 0. \end{cases}$$

We define

$$\tilde{C} = P^T C P \succeq 0,$$

where

$$P = \begin{bmatrix} 0 & J_d \\ 1 & 0 \\ 0 & I_d \end{bmatrix},$$

and J_d being the counter identity matrix of size d. The matrix $\Phi_k \in \mathbb{R}^{(d-1)\times(d-1)}$ is defined as in [22, Formula 2.95, p.50], they are zero matrices whenever k is out of range.

As shown in [22, § 5.1.1], the optimization problem (6.22) can be reformulated as the SDP

$$\begin{array}{ll} \min\limits_{h,\, Q_1,\ldots,\, Q_7} & h^T \tilde{C} h \\ & (1+\gamma_p)1_{k=0} - h_k = L_k(Q1)\,, \\ & h_k - (1-\gamma_p)1_{k=0} = L_{k,0,\omega_p}(Q_2, Q_3)\,, \\ \text{s.t.} & \gamma_s 1_{k=0} - h_k = L_{k,\omega_s,\pi}(Q_4, Q_5)\,, \\ & \gamma_s 1_{k=0} + h_k = L_{k,0,\omega_p}(Q_6, Q_7)\,, \quad k = 0,\ldots,d\,, \\ & Q_1 \succeq 0, \ldots, Q_7 \succeq 0, \end{array} \tag{6.23}$$

where Q_1, Q_2, Q_4, and Q_6 are real $(d+1)\times(d+1)$-matrices, Q_3, Q_5 and Q_7 are real $(d-1)\times(d-1)$-matrices, $L_k(A) = \operatorname{tr}(\Theta_k A)$, and

$$L_{k,\alpha,\beta}(A,B) = \operatorname{tr}(\Theta_k A) + \operatorname{tr}\Big(\big(\frac{a+b}{2}(\Phi_{k-1}+\Phi_{k+1}) - (ab+\frac{1}{2})\Phi_k - \frac{1}{4}(\Phi_{k-2}+\Phi_{k+2})\big)B\Big),$$

with $a = \cos\alpha$, $b = \cos\beta$. By contrast with the unconstrained case (Algorithm `csos3`), this program involves 7 real Gram matrix variables and $d+1$ real variables $h_0, \ldots, h_d$, which are the coefficients of the polynomial corresponding to the filter.

After solving (6.23), we obtain numerical values for the coefficients of h and the entries of $Q_1, \ldots, Q_7$, which are further rounded to $\hat{h}$ and $\hat{Q}_1, \ldots, \hat{Q}_7$. To project $\hat{Q}_1$ to a matrix Q_1 satisfying the first set of equality constraints in SDP (6.23), we apply the formula in Line 13 of Algorithm `csos3` after replacing f_k by $p_k := (1+\gamma_p)1_{k=0} - \hat{h}_k$. Similarly, we obtain the two matrices Q_2 and $Q_3 := \hat{Q}_3$ satisfying the second set of equality constraints in SDP (6.23), after substitution by

$$\hat{h}_k - (1-\gamma_p)1_{k=0} - \operatorname{tr}\Big(\big(\frac{a+b}{2}(\Phi_{k-1}+\Phi_{k+1}) - (ab+\frac{1}{2})\Phi_k - \frac{1}{4}(\Phi_{k-2}+\Phi_{k+2})\big)Q_3\Big),$$

Eventually, similar projection steps provide the remaining matrices $Q_4, \ldots, Q_7$ so that all equality constraints in (6.23) hold exactly.

Example 6.4.1. As in [22, Example 5.1], we design a filter with parameters $d = 25$, $\omega_p = \pi/5$, $\omega_s = \pi/4$, $\gamma_p = 1/10$ (corresponding to a passband ripple of 1.74 dB) and $\gamma_s = 0.0158$ (a stopband attenuation of 36 dB). We first obtain a numerical lower bound of the stopband energy $E_s' = 4.461501 \times 10^{-5}$. However, this bound happens to be inexact as the Gram matrices obtained after the projection step are not positive semi-definite anymore. To overcome this certification issue, we replace the last constraint in (6.23) by $Q_7 - 10^{-9} I_{24} \succeq 0$. Doing so, we can successfully project the approximate Gram matrices into exact ones with positive eigenvalues, and obtain a certified exact lower bound of $E_s = 4.461503 \times 10^{-5}$ in 0.74 seconds.

PART III

CONCLUSIONS AND PERSPECTIVES

CHAPTER 7

Conclusions and Perspectives

We summarize the main obtained results of the thesis and propose several directions to develop or improve them.

Contents

7.1 Exact certificates for real polynomials

Conclusions. In Chapter 5, we designed and analyzed two algorithms to decompose a non-negative polynomial as an SOS of polynomials/rational fractions modulo the gradient ideal with rational coefficients. The correctness of our framework relies on a genericity condition, namely that the gradient ideal of the input polynomial is zero-dimensional and radical. Practical experiments demonstrated that our algorithms can tackle a large range of problems that are out of reach for state-of-the-art algorithms.

Perspectives. We plan to further develop and improve our algorithms in the following directions:

Extension to the constrained case. We aim to provide a necessary and sufficient condition for the non-negativity of $f \in \mathbb{Q}[x]$ over a real algebraic variety by relying on polar varieties, as in [28].

Exploiting specific structures. We shall improve the scalability of our algorithms by exploiting the specific structure of the input polynomial, such as correlative [45] or term sparsity [88], symmetries [73] or by using recent improvements on the computation of critical sets when the related system is invariant under group actions [25].

Improving the bit complexity. We shall improve the bit complexities of the two algorithms `sosgradientshape` and `sosgradient` by reducing the degree of the univariate polynomial h in (5.2). Indeed, h is non-negative over the real roots of w in (5.1). Hence, the results of [41] can be applied.

Looking for new certificates. We also plan to seek new certificates for non-negativity of polynomials without imposing the zero-dimensional and radical condition on the

gradient ideal. In Example 5.1.3 we proposed a certificate of non-negativity for the Motzkin polynomial whose gradient ideal does not satisfy the condition. Note that the condition is generic. Hence, if we perturb the original polynomial with a tiny change then the condition holds. Along this line, we shall study coercive polynomials which are dense in $\mathbb{Q}[\boldsymbol{x}]$ w.r.t the ℓ_1-norm [37].

7.2 Exact certificates for complex polynomials

Conclusions. We have designed three algorithms, of polynomial bit complexity, to compute weighted sums of Hermitian squares decompositions for trigonometric univariate polynomials positive on the unit circle with Gaussian coefficients. Note that positivity of such a trigonometric polynomial f is equivalent to that of a polynomial

$$a_0 + \sum_{k=1}^{d} a_k \cos(kt) + b_k \sin(kt)$$

for all $t \in [0, 2\pi]$, where a_k, b_k are rational coefficients obtained from the coefficients f_i. In turn, if we do the change of variables $t = 2\arctan(x)$ then the trigonometric polynomial becomes a rational function whose denominator is a power of $(1 + x^2)$. Thus, this boils down to proving the positivity of a real univariate polynomial and so one can apply the methods from [53].

Perspectives. In the future, we plan to develop and improve our algorithms as follows:

Improving the bit complexity. The bit complexities obtained in Sections 6.2 and 6.3 are somehow artificial as they are based on the complexity of the ellipsoid method. In the practical experiments shown in Section 6.4, we relied on interior-point methods to solve the SDPs. The corresponding complexity has been recently analyzed in [21]. Even though in the latter article, the exponents of the bounds are not explicitly given, it would certainly be possible. Hence, the bit complexities in Sections 6.2 and 6.3 can be reduced. Moreover, it would also be possible to improve the resulting estimates by exploiting the specific structure of the Toeplitz/Gram matrices involved in our SDP program. Gluing together such results would certainly help to explain the discrepancy between our theoretical (high) complexity bounds and our practical (good) algorithmic performance for our application of filter design, as in Example 6.4.1.

Extension to the multivariate case. It is possible to extend our algorithm to the multivariate case, considering non-negativity on the unit n-circle. In this setting, the degree of the squares can be higher than the degree of the input polynomial. We plan to estimate the degree bound for the squares in our SOHS decompositions by relying on the kernel polynomial method used in [24, 83].

Extension to optimal power flow problems. We also intend to extend our certification techniques to the sparse setting [89] to obtain guaranteed bounds for optimal power flow instances [3] that are complex multivariable polynomial optimization problems.

Investigating non-negativity. We leave the situation where the input polynomial vanishes on the unit circle for future investigation. The extension beyond positivity is significantly more difficult and, in particular, none of our three algorithms can be applied to this case. This is because when f cancels on the unit circle, the perturbation steps in Algorithms `csos1` and `csos2` do not work and a Gram matrix associated to f will not be in the interior of the positive semi-definite matrix cone and so the rounding-projection method [67] cannot be applied.

Bibliography

[1] E. D. Andersen and K. D. Andersen. The MOSEK interior point optimizer for linear programming: an implementation of the homogeneous algorithm. In *High performance optimization*, pages 197–232. Springer, 2000.

[2] E. Artin. Über die zerlegung definiter funktionen in quadrate. In *Abhandlungen aus dem mathematischen Seminar der Universität Hamburg*, volume 5, pages 100–115. Springer, 1927.

[3] S. Babaeinejadsarookolaee, A. Birchfield, R. D. Christie, C. Coffrin, C. DeMarco, R. Diao, M. Ferris, S. Fliscounakis, S. Greene, R. Huang, et al. The power grid library for benchmarking AC optimal power flow algorithms. *arXiv preprint arXiv:1908.02788*, 2019.

[4] Z. Bai, J. Demmel, and A. McKenney. *On floating point errors in Cholesky*. University of Tennessee, 1989.

[5] B. Bank, M. Giusti, J. Heintz, M. S. El Din, and E. Schost. On the geometry of polar varieties. *Applicable Algebra in Engineering, Communication and Computing*, 21(1):33–83, 2010.

[6] B. Bank, M. Giusti, J. Heintz, and G. M. Mbakop. Polar varieties and efficient real elimination. *Mathematische Zeitschrift*, 238(1):115–144, 2001.

[7] B. Bank, M. Giusti, J. Heintz, and L. M. Pardo. Generalized polar varieties: Geometry and algorithms. *Journal of complexity*, 21(4):377–412, 2005.

[8] B. Bank, M. Giusti, J. Heintz, and M. Safey El Din. Intrinsic complexity estimates in polynomial optimization. *Journal of Complexity*, 30(4):430–443, 2014.

[9] S. Basu, R. Pollack, and M.-F. Roy. A new algorithm to find a point in every cell defined by a family of polynomials. In *Quantifier elimination and cylindrical algebraic decomposition*, pages 341–350. Springer, 1998.

[10] W. Baur and V. Strassen. The complexity of partial derivatives. *Theoretical computer science*, 22(3):317–330, 1983.

[11] A. Beck. *First-order methods in optimization*. SIAM, 2017.

[12] J. Berthomieu, C. Eder, and M. Safey El Din. Msolve: A library for solving polynomial systems. In *Proceedings of ISSAC*, page 51–58, 2021.

[13] J. Bezanson, A. Edelman, S. Karpinski, and V. B. Shah. Julia: A fresh approach to numerical computing. *SIAM review*, 59(1):65–98, 2017.

[14] D. A. Bini and L. Robol. Solving secular and polynomial equations: A multiprecision algorithm. *Journal of Computational and Applied Mathematics*, 272:276–292, 2014.

[15] G. Blekherman. There are significantly more nonegative polynomials than sums of squares. *Israel Journal of Mathematics*, 153(1):355–380, 2006.

[16] S. Boyd, S. P. Boyd, and L. Vandenberghe. *Convex optimization*. Cambridge University Press, 2004.

[17] A. L. B. Cauchy. Calcul des indices des fonctions. *Journal de l'Ecole Polytechnique*, 15(25):176 – 229, 1832.

[18] S. Chevillard, J. Harrison, M. Joldeş, and C. Lauter. Efficient and accurate computation of upper bounds of approximation errors. *Theoretical Computer Science*, 412(16):1523–1543, 2011.

[19] M.-D. Choi, T. Y. Lam, and B. Reznick. Sums of squares of real polynomials. In *Proceedings of Symposia in Pure mathematics*, volume 58, pages 103–126. American Mathematical Society, 1995.

[20] D. Cox, J. Little, and D. O'Shea. *Ideals, varieties, and algorithms: an introduction to computational algebraic geometry and commutative algebra*. Springer Science & Business Media, 2013.

[21] E. de Klerk and F. Vallentin. On the Turing model complexity of interior point methods for semidefinite programming. *SIAM Journal on Optimization*, 26(3):1944–1961, 2016.

[22] B. Dumitrescu. *Positive trigonometric polynomials and signal processing applications*. Springer, 2017.

[23] I. Dunning, J. Huchette, and M. Lubin. JuMP: A modeling language for mathematical optimization. *SIAM review*, 59(2):295–320, 2017.

[24] K. Fang and H. Fawzi. The sum-of-squares hierarchy on the sphere and applications in quantum information theory. *Mathematical Programming*, 190(1):331–360, 2021.

[25] J.-C. Faugère, G. Labahn, M. S. El Din, É. Schost, and T. X. Vu. Computing critical points for invariant algebraic systems. *Journal of Symbolic Computation*, 2022.

[26] P. Gianni and T. Mora. Algebraic solution of systems of polynomial equations using Gröbner bases. In *Applied Algebra, Algebraic Algorithms and Error Correcting Codes, Proceedings of AAECC-5, volume 356 of LNCS*, pages 247–257. Springer, 1989.

[27] M. Giusti, G. Lecerf, and B. Salvy. A gröbner free alternative for polynomial system solving. *Journal of complexity*, 17(1):154–211, 2001.

[28] A. Greuet, F. Guo, M. Safey El Din, and L. Zhi. Global optimization of polynomials restricted to a smooth variety using sums of squares. *Journal of Symbolic Computation*, 47(5):503–518, 2012.

[29] A. Greuet and M. Safey El Din. Probabilistic algorithm for polynomial optimization over a real algebraic set. *SIAM Journal on Optimization*, 24(3):1313–1343, 2014.

[30] D. Y. Grigor'ev and N. N. Vorobjov Jr. Solving systems of polynomial inequalities in subexponential time. *Journal of symbolic computation*, 5(1-2):37–64, 1988.

[31] M. Grötschel, L. Lovász, and A. Schrijver. *Geometric Algorithms and Combinatorial Optimization*, volume 2 of *Algorithms and Combinatorics*. Springer, second corrected edition edition, 1993.

[32] D. Henrion, S. Naldi, and M. Safey El Din. Exact algorithms for linear matrix inequalities. *SIAM Journal on Optimization*, 26(4):2512–2539, 2016.

[33] N. J. Higham. *Accuracy and stability of numerical algorithms*. SIAM, 2002.

[34] Hilbert. Ueber die Darstellung definiter Formen als Summen von Formenquadraten. *Mathematische Annalen*, 32:342–350, 1888.

[35] D. Hilbert. Mathematical problems. In *Mathematics: People· Problems· Results*, pages 273–278. Chapman and Hall/CRC, 2019.

[36] U. Jannsen. Hasse principles for higher-dimensional fields. *Annals of Mathematics*, pages 1–71, 2016.

[37] V. Jeyakumar, J. B. Lasserre, and G. Li. On polynomial optimization over non-compact semi-algebraic sets. *Journal of Optimization Theory and Applications*, 163(3):707–718, 2014.

[38] E. Kaltofen, B. Li, Z. Yang, and L. Zhi. Exact certification of global optimality of approximate factorizations via rationalizing sums-of-squares with floating point scalars. In *Proceedings of the twenty-first international symposium on Symbolic and algebraic computation*, pages 155–164, 2008.

[39] E. Kaltofen, Z. Yang, and L. Zhi. A proof of the monotone column permanent (MCP) conjecture for dimension 4 via sums-of-squares of rational functions. In *Proceedings of the 2009 conference on Symbolic numeric computation*, pages 65–70, 2009.

[40] E. L. Kaltofen, B. Li, Z. Yang, and L. Zhi. Exact certification in global polynomial optimization via sums-of-squares of rational functions with rational coefficients. *Journal of Symbolic Computation*, 47(1):1–15, 2012.

[41] T. Krick, B. Mourrain, and A. Szanto. Univariate rational sums of squares. *Revista de la Union Matematica Argentina*, 2022.

[42] W. Krull. Idealtheorie in Ringen ohne Endlichkeitsbedingung. *Mathematische Annalen*, 101(1):729–744, 1929.

[43] E. Landau. Über die darstellung definiter funktionen durch quadrate. *Mathematische Annalen*, 62(2):272–285, 1906.

[44] J. B. Lasserre. Global optimization with polynomials and the problem of moments. *SIAM Journal on optimization*, 11(3):796–817, 2001.

[45] J. B. Lasserre. Convergent SDP-relaxations in polynomial optimization with sparsity. *SIAM Journal on Optimization*, 17(3):822–843, 2006.

[46] M. Laurent. Sums of squares, moment matrices and optimization over polynomials. In *Emerging applications of algebraic geometry*, pages 157–270. Springer, 2009.

[47] T. Lickteig and M.-F. Roy. Sylvester–Habicht sequences and fast Cauchy index computation. *Journal of Symbolic Computation*, 31(3):315–341, 2001.

[48] H. Lombardi, D. Perrucci, and M.-F. Roy. *An elementary recursive bound for effective Positivstellensatz and Hilbert's 17th problem*, volume 263. American mathematical society, 2020.

[49] V. Magron. *The quest of modeling, certification and efficiency in polynomial optimization*. Habilitation thesis, Université Toulouse 3 Paul Sabatier, 2021.

[50] V. Magron and M. Safey El Din. On exact Polya and Putinar's representations. In *Proceedings of ISSAC*, pages 279–286, 2018.

[51] V. Magron and M. Safey El Din. RealCertify: a Maple package for certifying non-negativity. *ACM Communications in Computer Algebra*, 52(2):34–37, 2018.

[52] V. Magron and M. Safey El Din. On exact Reznick, Hilbert-Artin and Putinar's representations. *Journal of Symbolic Computation*, 107:221–250, 2021.

[53] V. Magron, M. Safey El Din, and M. Schweighofer. Algorithms for weighted sum of squares decomposition of non-negative univariate polynomials. *Journal of Symbolic Computation*, 93:200–220, 2019.

[54] V. Magron, M. Safey El Din, M. Schweighofer, and T. H. Vu. Exact SOHS decompositions of trigonometric univariate polynomials with Gaussian coefficients. In *Proceedings of the 2022 International Symposium on Symbolic and Algebraic Computation*, pages 325–332, 2022.

[55] V. Magron, M. Safey El Din, and T.-H. Vu. Sum of squares decompositions of polynomials over their gradient ideals with rational coefficients. *SIAM Journal of Oprimization*, accepted, 2022.

[56] V. Magron, H. Seidler, and T. de Wolff. Exact optimization via sums of nonnegative circuits and arithmetic-geometric-mean-exponentials. In *Proceedings of ISSAC*, pages 291–298, 2019.

[57] K. Mahler. An inequality for the discriminant of a polynomial. *Michigan Mathematical Journal*, 11(3):257–262, 1964.

[58] M. Marshall. *Positive polynomials and sums of squares*. Number 146. American Mathematical Society, 2008.

[59] K. Mehlhorn, M. Sagraloff, and P. Wang. From approximate factorization to root isolation with application to cylindrical algebraic decomposition. *Journal of Symbolic Computation*, 66:34–69, 2015.

[60] T. S. Motzkin. The arithmetic-geometric inequality. *Inequalities (Proc. Sympos. Wright-Patterson Air Force Base, Ohio, 1965)*, pages 205–224, 1967.

[61] K. G. Murty and S. N. Kabadi. Some NP-complete problems in quadratic and nonlinear programming. *Mathematical Programming*, 39:117–129, 1987.

[62] J. Nie, J. Demmel, and B. Sturmfels. Minimizing polynomials via sum of squares over the gradient ideal. *Mathematical Programming*, 106(3):587–606, 2006.

[63] P. A. Parrilo. *Structured semidefinite programs and semialgebraic geometry methods in robustness and optimization*. PhD thesis, California Institute of Technology, 2000.

[64] P. A. Parrilo. Semidefinite programming relaxations for semialgebraic problems. *Mathematical programming*, 96(2):293–320, 2003.

[65] P. A. Parrilo. Chapter 3: Polynomial optimization, sums of squares, and applications. In *Semidefinite Optimization and Convex Algebraic Geometry*, pages 47–157. SIAM, 2012.

[66] P. A. Parrilo. An explicit construction of distinguished representations of polynomials nonnegative over finite sets. *IfA AUT02-02*, ETH Zürich, 2002.

[67] H. Peyrl and P. A. Parrilo. Computing sum of squares decompositions with rational coefficients. *Theoretical Computer Science*, 409(2):269–281, 2008.

[68] A. Pfister. Zur darstellung definiter funktionen als summe von quadraten. *Inventiones mathematicae*, 4(4):229–237, 1967.

[69] L. Porkolab and L. Khachiyan. On the complexity of semidefinite programs. *Journal of Global Optimization*, 10(4):351–365, 1997.

[70] Y. Pourchet. Sur la représentation en somme de carrés des polynômes à une indéterminée sur un corps de nombres algébriques. *Acta Arithmetica*, 19(1):89–104, 1971.

[71] V. Powers. *Certificates of Positivity for Real Polynomials: Theory, Practice, and Applications*, volume 69. Springer Nature, 2021.

[72] B. A. Reznick. *Sum of even powers of real linear forms*, volume 463. American Mathematical Society, 1992.

[73] C. Riener and M. Safey El Din. Real root finding for equivariant semi-algebraic systems. In *Proceedings of the 2018 ACM International Symposium on Symbolic and Algebraic Computation*, pages 335–342, 2018.

[74] R. M. Robinson. Some definite polynomials which are not sums of squares of real polynomials. *Selected questions of algebra and logic (collection dedicated to the memory of A. I. Malcev) (Russian)*, pages 264–282, 1973.

[75] F. Rouillier. Solving zero-dimensional systems through the rational univariate representation. *Applicable Algebra in Engineering, Communication and Computing*, 9(5):433–461, 1999.

[76] M. Safey El Din and E. Schost. Polar varieties and computation of one point in each connected component of a smooth real algebraic set. In *Proceedings of the 2003 international symposium on Symbolic and algebraic computation*, pages 224–231, 2003.

[77] M. Safey El Din and É. Schost. Bit complexity for multi-homogeneous polynomial system solving–application to polynomial minimization. *Journal of Symbolic Computation*, 87:176–206, 2018.

[78] M. Safey El Din and L. Zhi. Computing rational points in convex semi-algebraic sets and sums-of-squares decompositions. *SIAM Journal on Optimization*, 20(6):2876–2889, 2010.

[79] C. Scheiderer. Sums of squares of polynomials with rational coefficients. *Journal of the European Mathematical Society*, 18(7):1495–1513, 2016.

[80] M. Schweighofer. *Algorithmische beweise für nichtnegativ-und positivstellensätze*. Master's thesis, Universität Passau, 1999.

[81] N. Z. Shor. Class of global minimum bounds of polynomial functions. *Cybernetics*, 23(6):731–734, 1987.

[82] J. Skowron and A. Gould. General complex polynomial root solver and its further optimization for binary microlenses. *arXiv:1203.1034*, 2012.

[83] L. Slot. Sum-of-Squares hierarchies for polynomial optimizatpion and the Christoffel–Darboux kernel. *SIAM Journal on Optimization*, 32(4):2612–2635, 2022.

[84] Ė. B. Vinberg. *A course in algebra*. Number 56. American Mathematical Soc., 2003.

[85] J. Von Zur Gathen and J. Gerhard. *Modern Computer Algebra*. Cambridge University Press, 2013.

[86] H. Waki, M. Nakata, and M. Muramatsu. Strange behaviors of interior-point methods for solving semidefinite programming problems in polynomial optimization. *Computational Optimization and Applications*, 53(3):823–844, 2012.

[87] J. Wang and V. Magron. A second order cone characterization for sums of nonnegative circuits. In *Proceedings of the 45th International Symposium on Symbolic and Algebraic Computation*, pages 450–457, 2020.

[88] J. Wang, V. Magron, and J.-B. Lasserre. Chordal-TSSOS: a moment-SOS hierarchy that exploits term sparsity with chordal extension. *SIAM Journal on Optimization*, 31(1):114–141, 2021.

[89] J. Wang, V. Magron, J. B. Lasserre, and N. H. A. Mai. CS-TSSOS: Correlative and term sparsity for large-scale polynomial optimization. *ACM Transactions on Mathematical Software*, Accepted for publication, 2022.

[90] H. Wolkowicz, R. Saigal, and L. Vandenberghe. *Handbook of semidefinite programming: theory, algorithms, and applications*, volume 27. Springer Science & Business Media, 2012.