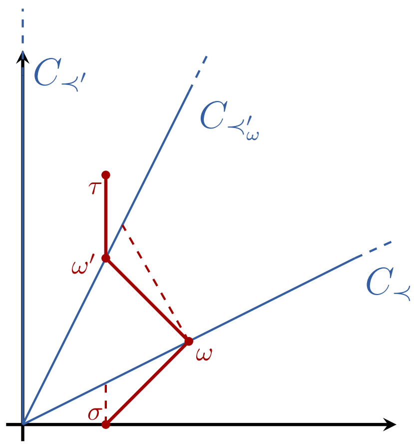
Technical University of Berlin

Fakultät 2
Institut für Mathematik

The Gröbner walk revisited

Thesis MSc Mathematics



Supervisor and First Reviewer:
Prof. Dr. Carlos Enrique Améndola Cerón

Second Reviewer:
Prof. Dr. Michael Joswig

Author:
Francesco Nowell

July 2024

ABSTRACT

The *Gröbner walk* is an algorithm for Gröbner basis conversion of ideals in the polynomial ring $I \lhd k[x_1, ..., x_n]$. It exploits the geometric properties of the Gröbner fan, which is a polyhedral fan in $\mathbb{R}^n$, the maximal cones of which are in one-to-one correspondence with the marked Gröbner bases of I. In this text, we describe the functionality of the algorithm step by step and provide a formal proof of correctness. We then survey a series of modifications which have been proposed over the years, most notably *path perturbation* and the *generic Gröbner walk*. Finally we discuss the performance of the Gröbner walk in a variety of settings and compare it with other Gröbner basis conversion algorithms in both `Macaulay2` and `OSCAR`.

GERMAN ABSTRACT

Der *Gröbner walk* ist ein Algorithmus für den Wechsel von Gröbnerbasen eines polynomiellen Ideals $I \lhd k[x_1, ..., x_n]$. Das Verfahren nutzt die geometrischen Eigenschaften des *Gröbnerfächers* aus; letzterer ist ein polyedrischer Fächer in $\mathbb{R}^n$, dessen Kegel maximaler Dimension in Eins-zu-eins Korrispondenz zu den markierten Gröbnerbasen von I sind. In diesem Text werden die Schritte des Algorithmus ausführlich beschrieben und es wird einen formalen Beweis seiner Korrektheit geliefert. Weiterhin beschreiben wir eine Reihe von Veränderungen wie der *perturbierte* und der *generische walk*, die im Laufe der Jahre enstanden sind. Zum Schluss untersuchen wir die Leistung der unterschiedlichen *Gröbner walks* in einer Vielzahl von Kontexten und vergleichen sie mit weiteren Algorithmen zur Gröbnerbasis-Berechnung in `Macaulay2` und `OSCAR`.

SELBSTÄNDIGKEITSERKLÄRUNG

Hiermit erkläre ich, dass ich die vorliegende Arbeit selbständig und eigenhändig sowie ohne unerlaubte fremde Hilfe und ausschließlich unter Verwendung der aufgeführten Quellen und Hilfsmittel angefertigt habe.

Berlin, den ....

ACKNOWLEDGEMENTS

There are so many wonderful people to thank. First and foremost, I would like to thank my supervisor Carlos Améndola for being a constant source of insights, enthusiasm, encouragement, and (perhaps most importantly) cheerfulness over this past year. I greatly look forward to the work that lies ahead. The `OSCAR` implementation would simply not have been possible without Kamillo Ferry: I am forever indebted to you for your friendly explanations, your patience with my slow laptop and even slower hands, and your overall computer wizardry. Special thanks also go to Jordi Welp and Anne Frübis-Krüger for laying the groundwork of the implementation and offering their support to us along the way. More generally, I am extremely grateful to all of the TU faculty members who made my master's studies such a fascinating and enjoyable experience: Marta Panizzut, Michael Joswig, Martin Henk, Peter Bürgisser, John Sullivan, and many more. Quite frankly, I was not expecting the maths bug to bite me as severely as it has, and I have you to thank for this.

In Italian, we use the adjective *poliedrico* (literally "polyhedral") to refer to someone with a wide variety of skills and/or interests. I would like to thank my many friends who assist me every day in being the fullest, most multi-faceted person possible: Ben, Daniel, Isabelle, Hojoong, Zoey, Robin1, Robin2, Charlotte, Alfie, Otis, Amedeo, Kroner, Victoria, Georg, and Haley. Finally, I would like to thank my parents and my brother Joe, who know every side of me and have supported me throughout.

0 Introduction

Gröbner bases play a central role in computational algebra. They solve the ideal membership and equality problems and are a vital tool for the study and solution of systems of polynomial equations in several variables. For example, the variety of a zero-dimensional ideal may be computed via a lexicographic Gröbner basis. Here, the variables in the individual polynomials are isolated in such a way that the system may be solved via backwards substitution. Because of this, Gröbner basis computation is sometimes considered a polynomial analogue of Gaussian elimination.

Bruno Buchberger's original algorithm for Gröbner basis computation remains widely used in practical applications to this day; however, it is well known that its performance is extremely variable, depending on both the generating set and the monomial order given as inputs. Unfortunately, the determination of lexicographic Gröbner bases (and more widely, elimination term orderings) tends to be considerably more expensive than, for example, the computation of a degree reverse lexicographic Gröbner basis. Because of this, newer approaches often adopt a two-stage strategy in which a Gröbner basis w.r.t a "cheap" order is computed first, and subsequently converted to a Gröbner basis of the desired order.

The Gröbner walk is an example of such a conversion algorithm; given a reduced Gröbner basis of an ideal $I \lhd \mathbb{Q}[x_1, ..., x_n]$ w.r.t some *starting monomial order* $\prec$, it computes the reduced Gröbner basis of I w.r.t desired *target order* $\prec'$ via a series of intermediate steps. This is done by exploiting the geometry of the *Gröbner fan*, which is a polyhedral fan in $\mathbb{R}^n$, the maximal cones of which are in one-to-one correspondence with the marked Gröbner bases of I. We call these maximal cones *Gröbner cones*.

The intermediate computations are determined by tracing a line segment between the Gröbner cones corresponding to $\prec$ and $\prec'$; each time the line segment enters a new cone at a point ω, a Gröbner basis the corresponding monomial order is computed by first computing a Gröbner basis of the ideal of initial forms $in_\omega(I)$ with Buchberger's algorithm and subsequently "lifting" this basis to a Gröbner basis of I. A feature of these intermediate computations is that they occur on truncated polynomials which generally have fewer terms than the Gröbner basis elements of I. Hence, the Gröbner walk's strategy may be described as replacing one "heavy" computation (the direct computation of a Gröbner basis of I with respect to $\prec'$) by many "light" instances (the computation of the starting Gröbner basis and the intermediate conversions).

Over the years, different strategies for the actual execution of the walk have been proposed. For example, [CKM97] originally suggested to perturb the starting and target vectors by some "small" quantity (in such a way as to remain in their respective cones) in order to guarantee that the line segment intersects boundaries of cones only at facets. This is a desirable outcome since the number of terms of the initial forms $in_\omega(g)$ is minimal when ω lies on a *facet* of a Gröbner cone.

Unfortunately this method of *deterministic path perturbation* is ill-advised in practice, as it leads to intermediate vectors with long coefficients, resulting in heavy computations and issues with accuracy. However, path perturbation formed the theoretical basis for the *generic Gröbner walk* [Fuk+07]. Here, a symbolic perturbation on the start and target vectors is carried out instead, and all intermediate sets of the form $in_\omega(G)$ are computed via the outer normal vector of the facet on which ω lies. The vector ω itself remains unknown throughout, thus avoiding the aforementioned numerical issues.

This text has two main aims; the first is to constitute a comprehensive resource for the theory behind the original algorithm and its subsequently introduced variants. This includes formal proofs of correctness which have been as of yet absent in the literature. The second aim is to implement the algorithm in `OSCAR` and to perform tests and comparisons: both of the Gröbner walk and other Gröbner basis computation algorithms, and of the `OSCAR` implementation compared to the already present `Macaulay2` implementation.

In Section 1 we recall some definitions and prove preliminary results concerning monomial orders, weight vectors and Gröbner cones. In Section 2 we formalize the algorithm and prove its correctness. Section 3 contains several step-by-step executions of the algorithm on a small example which emphasizes the importance of the choice of path. Section 4 and Section 5 are discussions of the *perturbed* and *generic* variants of the algorithm respectively. Section 6 investigates the performance of the Gröbner walk in a variety of settings in `Macaulay2`. Section 7 presents our implementation of the Gröbner walk in `OSCAR` and discusses some preliminary benchmark results.

Contents

List of Figures

List of Tables

1 Preliminaries

Throughout this text we will adopt the following notation. Definitions and elementary results concerning ideals, monomial orders and Gröbner bases can be found in Appendix A.

Notation.

- For an arbitrary field K, we denote the polynomial ring in n variables over K by $K[x_1, ..., x_n]$.
- We denote ideals in $K[x_1, ..., x_n]$ by I and write $I \lhd K[x_1, .., x_n]$. For a subset $M \subseteq K[x_1, ..., x_n]$, we denote the ideal generated by M by $\langle M \rangle$.
- We denote the set of all monomials in $x_1, ..., x_n$ by $\mathrm{Mon}_n(x)$. We adopt the convention of writing monomials as $x^\alpha = x_1^{\alpha_1} \cdot x_2^{\alpha_2} \cdot ... \cdot x_n^{\alpha_n}$, where $\alpha = (\alpha_1, ..., \alpha_n) \in \mathbb{N}^n$.
- We we call admissible total orders on the set $\mathrm{Mon}_n(x)$ which are also well-orderings *monomial orders* (Definition A.5). We denote them by the symbol "$\prec$" and variants thereof. The symbol "$<$" is reserved for ordering relations on vector spaces.
- Given a monomial order $\prec$ and a non-zero polynomial $f \in K[x_1, ..., x_n]$, we denote the leading term of f with respect to $\prec$ by $\mathrm{LT}_\prec(f)$ and its leading monomial by $in_\prec(f)$ (cf. Definition A.9).
- The initial ideal of an ideal I with respect to $\prec$ is denoted by $in_\prec(I)$ (cf. Definition A.11).
- We denote the *marked* Gröbner basis of I w.r.t $\prec$ by $G_\prec$. This is the monic reduced Gröbner basis of I with the leading terms identified (cf. Definition A.16).

Unless otherwise specified, we always assume that the underlying field is $\mathbb{Q}$.

1.1 Weight vectors and initial forms

Let $\mathbb{Q}^n_{\geq 0} := \{\omega = (\omega_1, ..., \omega_n) \in \mathbb{Q}^n \ , \ \omega_i \geq 0 \ \text{ for all } i \in \{1, ..., n\}\}$ denote the non-negative orthant of $\mathbb{Q}^n$. We refer to vectors $\omega \in \mathbb{Q}^n_{\geq 0}$ as *weight vectors.*

Definition 1.1. Let $\omega \in \mathbb{Q}^n_{\geq 0}$ be a weight vector.

- Given a polynomial $f = \sum c_\alpha x^\alpha \in \mathbb{Q}[x_1, ..., x_n]$, the **support** of f is the set of exponent vectors of its terms:
$$\mathrm{supp}(f) := \{\alpha \in \mathbb{N}^n : c_\alpha \neq 0\} \subset \mathbb{N}^n$$
- the **ω-degree** of a monomial x^α is
$$\deg_w(x^\alpha) := \sum_{i=1}^n w_i \alpha_i = \langle \omega, \alpha \rangle \in \mathbb{Q}_{\geq 0}.$$
- the ω-degree of a non-zero *poly*nomial $f \in \mathbb{Q}[x_1, ..., x_n]$ is the maximal $\omega-$degree of its terms:
$$\deg_\omega(f) := \max_{\alpha \in \mathrm{supp}(f)} \{\langle \omega, \alpha \rangle\}$$
- For a non-zero polynomial $f \in \mathbb{Q}[x_1, ...x_n]$ the **initial form** of f w.r.t ω is the sum of all terms of f of maximal ω-degree:
$$in_\omega(f) := \sum_{\alpha \in S'_f} c_\alpha x^\alpha \qquad \text{where} \qquad S'_f := \{\alpha \in \mathrm{supp}(f) : \langle \omega, \alpha \rangle = \deg_\omega(f)\}.$$

- the ω**-tail** of a non-zero polynomial f is the polynomial $f - in_\omega(f)$. We denote it by $\text{tail}_\omega(f)$.
- Given an ideal $I \lhd \mathbb{Q}[x_1, ..., x_n]$ the **ideal of initial forms** of I with respect to ω is

$$in_\omega(I) = \langle \{ in_\omega(f) \ , \ f \in I \} \rangle$$

- For a set of polynomials $F \subset \mathbb{Q}[x_1, ..., x_n]$ we define the set of its initial forms

$$in_\omega(F) := \{ in_\omega(f) \mid \ f \in F \}.$$

While $in_\prec(I)$ is a monomial ideal for any monomial order $\prec$, the ideal of initial forms $in_\omega(I)$ generally is not:

Example 1.2. Let $I = \langle x + y^3, \ x^2 + xy \rangle \lhd \mathbb{Q}[x, y]$, $\prec$ be the lexicographic order with $x \succ y$ and $\omega = (3, 1)$. Then

$$in_\prec(I) = \langle x, \ x^2 \rangle = \langle x \rangle \qquad \text{and} \qquad in_\omega(I) = \langle x + y^3, \ x^2 \rangle \ .$$

The latter is not to be confused with the set of initial forms

$$in_\omega(\{x + y^3, \ x^2 + xy\}) = \{x + y^3, x^2\}$$

which is not an ideal.

Remark 1.3. Let $f, g \in \mathbb{Q}[x_1, ..., x_n]$ be non-zero polynomials $\omega \in \mathbb{Q}^n_{n \geq 0}$. The following observations follow from the notions in Definition 1.1:

(*i*) $\deg_\omega(f) = \deg_\omega(in_\omega(f))$

(*ii*) $\deg_\omega(fg) = \deg_\omega(f) + \deg_\omega(g)$

(*iii*) $\deg_\omega(f + g) \leq \max\{\deg_\omega(f), \deg_\omega(g)\}$

(*iv*) For $\omega = (1, ..., 1) = \mathbf{1}$

$$\deg_\omega(f) = \deg_{\mathbf{1}}(f) = \max_{\alpha \in \text{supp}(f)} \left\{ \sum_{i=1}^n \alpha_i \right\}$$

is what is commonly known as the *total degree* of f.

The definitions and results which follow describe the interplay between weight vectors and monomial orders. They are taken from [Stu95, Chapter 1] and [CKM97]. In particular, our Proposition 1.9 is a slightly stronger version of [Stu95, Corollary 1.9].

Definition 1.4. We say that monomial order $\prec$ **refines** a weight vector $\omega \in \mathbb{Q}^n_{\geq 0}$ if the following holds:

$$\langle \omega, \beta \rangle < \langle \omega, \alpha \rangle \implies x^\beta \prec x^\alpha \qquad \text{for all} \quad \alpha, \beta \in \mathbb{N}^n. \tag{1}$$

Given a monomial order $\prec$ and a weight vector ω, a new monomial order may be defined as follows:

Definition 1.5. Given a monomial order $\prec$ and weight vector $\omega \in \mathbb{Q}^n_{\geq 0}$, the **refinement** of ω w.r.t $\prec$ is the relation on $\text{Mon}_n(x)$ defined by

$$x^\beta \prec_\omega x^\alpha \quad \text{ if and only if } \ \langle \omega, \beta \rangle < \langle \omega, \alpha \rangle \quad \text{or} \quad \big(\langle \omega, \beta \rangle = \langle \omega, \alpha \rangle \quad \text{and} \quad x^\beta \prec x^\alpha \big). \tag{2}$$

Expressed in words, $\prec_\omega$ compares the ω-degrees of two monomials and breaks ties with $\prec$. The proof that $\prec_\omega$ is indeed a monomial order can be found in the appendix (cf. proposition A.8).

Remark 1.6. For any monomial order $\prec$ and weight vector $\omega \in \mathbb{Q}^n_{\geq 0}$, the monomial order $\prec_\omega$ refines ω.

Lemma 1.7. *Let $\omega \in \mathbb{Q}^n_{\geq 0}$ be a weight vector and $\prec$ be a monomial order. Then:*

$$\prec \ \textit{ refines } \ \omega \iff in_\prec(in_\omega(f)) = in_\prec(f) \quad \textit{for all} \ \ f \in \mathbb{Q}[x_1, ..., x_n], \quad f \neq 0. \tag{3}$$

Proof. We prove " $\implies$ " by contraposition; assuming that there exists a non-zero polynomial f such that $in_\prec(f) = x^\beta$ and $in_\prec(in_\omega(f)) = x^\alpha$ for some $\beta, \alpha \in \text{supp}(f)$ with $\beta \neq \alpha$ then in particular $\beta \notin \text{supp}(in_\omega(f))$ otherwise the monomials would coincide. This implies $\langle\omega, \beta\rangle < \langle\omega, \alpha\rangle$. However, $in_\prec(f) = x^\beta$ implies $x^\alpha \prec x^\beta$. Thus, α and β do not fulfill the condition (1), so $\prec$ does not refine ω.

" $\impliedby$ ": If $in_\prec(in_\omega(f)) = in_\prec(f)$ holds for all $f \neq 0$, then it holds in particular for binomials of the form $x^\alpha + x^\beta$. Assuming $\langle\omega, \alpha\rangle < \langle\omega, \beta\rangle$, then $in_\prec(x^\alpha + x^\beta) = in_\prec(in_\omega(x^\alpha + x^\beta)) = in_\prec(x^\beta) = x^\beta$. This implies $x^\alpha \prec x^\beta$, so (by Definition 1.4) ω refines $\prec$. □

Proposition 1.8. *Let I be an ideal, $\prec$ a monomial order, and $\omega \in \mathbb{Q}^n_{\geq 0}$ a weight vector. Then*

$$in_{\prec_\omega}(I) = in_\prec(in_\omega(I)). \tag{4}$$

Proof. "$\subseteq$": The key observation here is that for any non-zero $f \in \mathbb{Q}[x_1, .., x_n]$:

$$in_{\prec_\omega}(f) = in_\prec(in_\omega(f)). \tag{5}$$

This is due to the definition of $\prec_\omega$: the leading monomial of a polynomial w.r.t $\prec_\omega$ is the unique monomial of $in_\omega(f)$ which is maximal with respect to $\prec$. The inclusion follows immediately.
"$\supseteq$": As both sets in (4) are monomial ideals, it suffices to show that an arbitrary monomial in $in_\prec(in_\omega(I))$ lies in $in_{\prec_\omega}(I)$. (This is a consequence of Lemma A.3.) Let $x^\alpha \in in_\prec(in_\omega(I))$. Then there exists a $g \in in_\omega(I)$ such that $in_\prec(g) = x^\alpha$. It follows from the multivariate division algorithm (more specifically, applying Proposition A.12 to g, I and $\prec_\omega$) that the following statement holds:

There exists an $f \in I$ such that $f = g + h$ and $\deg_\omega(h) < \deg_\omega(g)$.

Consequently, $in_{\prec_\omega}(f) = in_{\prec_\omega}(g + h) = in_{\prec_\omega}(g) = in_\prec(in_\omega(g)) = x^\alpha$, proving $x^\alpha \in in_{\prec_\omega}(I)$. □

This result has two important consequences.

Proposition 1.9. *Let I be an ideal, $\prec$ be a monomial order, and ω be a weight vector. Let $G_{\prec_\omega}$ be the marked Gröbner basis of I w.r.t $\prec_\omega$. Then the set*

$$in_\omega(G_{\prec_\omega}) = \{in_\omega(g) \mid g \in G_{\prec_\omega}\}$$

is the marked Gröbner basis of $in_\omega(I)$ w.r.t $\prec$, where the markings of $in_\omega(G_\prec)$ are taken to be those of $G_{\prec_\omega}$.

Proof. For readability we set $G' := in_\omega(G_{\prec_\omega})$. We first show that G' is a Gröbner basis of $in_\omega(I)$ by showing $in_\prec(G') = in_\prec(in_\omega(I))$.

"$\subseteq$" follows directly from $G' \subseteq in_\omega(I)$.

"$\supseteq$": let $x^\alpha \in in_\prec(in_\omega(I))$. Then $x^\alpha \in in_{\prec_\omega}(I)$ by (5). As $G_{\prec_\omega}$ is a Gröbner basis of I with respect to $\prec_\omega$ we have that

$$x^\alpha \in in_{\prec_\omega}(G_{\prec_\omega}) = in_\prec(in_\omega(G_{\prec_\omega})) = in_\prec(G') \tag{6}$$

as desired.

To show that G' is reduced, we must prove that no term of $in_\omega(g)$ is divisible by any of the leading terms of the initial monomials $in_\omega(\tilde{g}) \in G' \setminus \{in_\omega(g)\}$. Assuming this is not the case, we have that for some $in_\omega(g) \in G'$ and $\alpha \in \text{supp}(in_\omega(g))$, the monomial x^α lies in $in_\prec(G' \setminus \{in_\omega(g)\})$. Then Lemma A.3 implies that there is some $in_\omega(\tilde{g}) \in G' \setminus \{in_\omega(g)\}$ such that $in_\prec(in_\omega(\tilde{g}))$ divides x^α. But now since $\alpha \in \text{supp}(g)$ and $in_\prec(in_\omega(\tilde{g})))$ is a monomial appearing in $\tilde{g} \in G_{\prec_\omega} \setminus \{g\}$, this would imply that $G_{\prec_\omega}$ is not reduced.
Furthermore, the polynomials in G' are monic. This is a consequence of the observation (5) from the previous proof; we have that

$$\text{LT}_\prec(in_\omega(g_i)) = \text{LT}_{\prec\omega}(g_i) = in_{\prec_\omega}(g_i) \text{ for all } in_\omega(g_i) \in G', \tag{7}$$

where the second equality holds because $G_{\prec_\omega}$ is reduced (and particularly monic). This proves that the set (6) is the reduced Gröbner basis of $in_\omega(I)$ with respect to $\prec$. Finally, (7) implies that the markings of this basis are precisely those of the original basis $G_{\prec_\omega}$. This concludes the proof. $\square$

Corollary 1.10. *If $\omega \in \mathbb{Q}^n_{\geq 0}$ is such that $in_\omega(I)$ is a monomial ideal, then*

$$in_\omega(I) = in_{\prec_\omega}(I)$$

for any monomial order $\prec$.

Proof. We first show "$\subseteq$" : Let $in_\omega(I) = \langle \{x^{\alpha_1}, ...x^{\alpha_t}\} \rangle$ for some $\alpha_1, ..., \alpha_t \in \mathbb{N}^n$. Then for all i, $x^{\alpha_i} = in_\prec(x^{\alpha_i}) \in in_\prec(in_\omega(I)) = in_{\prec_\omega}(I)$. This proves "$\subseteq$".

"$\supseteq$" follows from Proposition 1.8: if $x^\alpha \in in_{\prec_\omega}(I) = in_\prec(in_\omega(I))$, then there exists an $f \in in_\omega(I)$ such that $in_\prec(f) = x^\alpha$. Due to the assumption that $in_\omega(I)$ is a monomial ideal, each term of f lies in $in_\omega(I)$. Thus, $x^\alpha \in in_\omega(I)$. This proves "$\supseteq$" (Once again, it suffices to show the inclusions for monomials, as both sets are monomial ideals). $\square$

Upon fixing an ideal $I \lhd \mathbb{Q}[x_1, .., x_n]$ we can introduce the notion of a weight vector ω *representing* a monomial order $\prec$.

Definition 1.11. Let $I \lhd \mathbb{Q}[x_1, ..., x_n]$ and $\prec$ be a monomial order. We say that a weight vector $\omega \in \mathbb{Q}^n_{\geq 0}$ **represents** $\prec$ for I if

$$in_\prec(I) = in_\omega(I).$$

If the ideal I being referred to is clear from context, we may simply say that ω represents $\prec$.

While the notion of refinement is not dependent on any particular ideal I, the notion of representation clearly is.

Example 1.12. Consider $\mathbb{Q}[x, y]$ with the monomial order $\prec$ defined to be the graded lexicographic ordering $glex_{x \succ y}$. Then the vector $\omega = (1, 1)$ refines $\prec$, seeing as $\prec$ first compares total degree and subsequently breaks ties with lex. It also represents the ideal $I = \langle x^2 + y \rangle$, as $in_\omega(I) = \langle x^2 \rangle = in_\prec(I)$. However, it does not represent the ideal $J = \langle x + y \rangle$ as $in_\omega(J) = \langle x + y \rangle \neq \langle x \rangle = in_\prec(J)$.

1.2 Gröbner cones

The statement of Corollary 1.10 may be rephrased as follows: given an ideal I and a sufficiently generic weight vector ω, then there exists a monomial order $\prec$ such that ω represents $\prec$. The following result from [Stu95] states that conversely, given an ideal I and a monomial order $\prec$, we can find a weight vector $\omega \in \mathbb{Q}^n_{\geq 0}$ such that ω represents $\prec$ with respect to I.

Theorem 1.13. *Let $G_\prec = \{g_1, ..., g_s\}$ be the reduced Gröbner basis of an ideal I with respect to a monomial order $\prec$. Then any vector ω in the set*

$$\{\omega \in \mathbb{R}^n_{\geq 0} \ : \ in_\omega(g_i) = in_\prec(g_i) \textit{ for } i = 1, ...s\} \tag{8}$$

represents $\prec$.

Proof. Consult [Stu95, Theorem 1.11 (pg. 4)]. □

The proof of this theorem states even more; the vectors which represent I w.r.t $\prec$ lie in the interior of a full-dimensional polyhedral cone (cf. Definition C.2). We call such full-dimensional cones *Gröbner cones*.

Definition 1.14.

- The topological closure of the set described in (8) is called the **Gröbner cone** of I w.r.t $\prec$. It is an n-dimensional polyhedral cone with non-empty interior. When the ideal I is clear from context, we denote it by $C_\prec$.

$$C_\prec = \overline{\{\omega \in \mathbb{R}^n : in_\omega(g_i) = in_\prec(g_i) \text{ for } i = 1, ...s\}} \tag{9}$$

- The polyhedral complex (cf. definition C.11) which has Gröbner cones as full-dimensional faces and lower-dimensional faces defined by their incidences is called the **Gröbner fan** of I. We denote it by $\mathrm{GFan}(I)$.
- The **Gröbner region** of I is

$$\mathrm{GR}(I) = \{\omega' \in \mathbb{R}^n\colon\ in_\omega(I) = in_{\omega'}(I) \text{ for some } \omega \in \mathbb{R}^n_{\geq 0}\}$$

  It is the region covered by the Gröbner fan, and contains the entire non-negative orthant $\mathbb{Q}^n_{\geq 0}$.

For more details (such as proof of the fact that the Gröbner fan is indeed a fan) we refer to [MR88] and [Stu95]. There is one subtlety to be noted: in the aforementioned papers , Gröbner cones/fans are defined as sets in $\mathbb{R}^n$, and this is necessary for them to be considered as polyhedral objects in the proper sense. The notions of initial forms/weight vectors from the previous section may be defined on $\mathbb{R}^n$ without restriction; however, as we are ultimately interested in matters of implementation, we chose to define these notions directly over the rational numbers. We shall continue with this convention, keeping in mind that we may at times naturally identify a Gröbner cone $C_\prec$ with its set of rational points $C_\prec \cap \mathbb{Q}^n$.

Example 1.15. Let $I = \langle x + y^3, x^2 + xy\rangle$ be the ideal from Example 1.2. The marked Gröbner basis of I w.r.t $\prec$, where $\prec$ is the lexicographic ordering $lex_{x \succ y}$ is

$$G_\prec = \{\underline{y^6} - y^4, \underline{x} + y^3\}.$$

The corresponding Gröbner cone is topological closure of the set of all vectors $\omega \in \mathbb{R}^2$ such that $\langle\omega, (0, 6)\rangle > \langle\omega, (0, 4)\rangle$ and $\langle\omega, (1, 0)\rangle > \langle\omega, (0, 3)\rangle$. This is the 2-dimensional cone with rays $(1, 0)$ and $(3, 1)$.

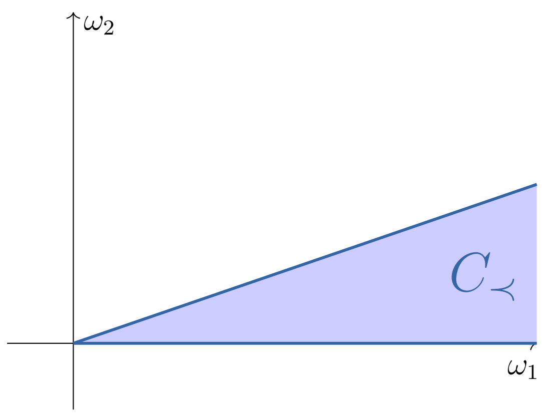


Figure 1: The Gröbner cone of I with respect to the lexicographic ordering. By Theorem 1.13, any vector in the interior of $C_\prec$ represents $\prec$ for I.

In the remainder of this chapter we present results from the original paper on the Gröbner walk [CKM97] with more detailed proofs and notation consistent with our own.

Corollary 1.16. *([CKM97, Lemma 2.1])*
Let I be an ideal, $\omega \in \mathbb{Q}^n_{\geq 0}$ a weight vector, $\prec$ a monomial order and $G_\prec = \{g_1, ..., g_s\}$ be the reduced Gröbner basis of I w.r.t $\prec$. Then the following holds:

$$\omega \text{ represents } \prec \iff in_\omega(g) = in_\prec(g) \quad \text{for all } g \in G_\prec. \tag{10}$$

Proof. " $\Longleftarrow$ " is a reformulation of the statement of Theorem 1.13.
" $\Longrightarrow$ ": If $in_\omega(I) = in_\prec(I)$ holds, then in particular $in_\omega(I)$ is a monomial ideal. By Proposition 1.9

$$in_{\prec_\omega}(I) = in_\omega(I) = in_\prec(I)$$

Thus, the 1-1 correspondence between initial ideals and marked Gröbner bases (cf. Proposition A.17) implies $G_{\prec_\omega} = G_\prec$. In particular we have equality of the leading terms, therefore

$$in_{\prec_\omega}(g) = in_\prec(in_\omega(g)) = in_\prec(g) \quad \text{for all} \quad g \in G_\prec. \tag{11}$$

By Proposition 1.8 we know that $\{in_\omega(g_i), i \in \{1, .., s\}\}$ is the reduced Gröbner basis of $in_\omega(I)$ w.r.t $\prec$. Under our assumption that $in_\omega(I)$ is a monomial ideal, each $in_\omega(g_i)$ must be a monomial (Clearly, reduced Gröbner bases of monomial ideals consist solely of monomials). Thus, (11) can only hold if $in_\omega(g_i) = in_\prec(g_i)$ holds for all i. This concludes the proof. □

Corollary 1.17. *([CKM97, Lemma 2.2])*
For two monomial orders $\prec_1$, $\prec_2$ we have:

$$C_{\prec_1} = C_{\prec_2} \iff in_{\prec_1}(g) = in_{\prec_2}(g) \quad \text{for all } g \in G_{\prec_1} \tag{12}$$

or equivalently:

$$C_{\prec_1} = C_{\prec_2} \iff G_{\prec_1} = G_{\prec_2}\ , \tag{13}$$

where the right hand side of (13) *is an equality of* <u>*marked*</u> *Gröbner bases. That is, we require both the set equality $G_{\prec_1} = \{g_1, ..., g_s\} = G_{\prec_2}$ and the equality of markings, i.e. that $in_{\prec_1}(g_i) = in_{\prec_2}(g_i)$ holds for all $i \in \{1, ..., s\}$.*

Proof. " $\Longrightarrow$ " follows immediately from the results above by taking a weight vector ω that represents both monomial orders.
" $\Longleftarrow$ " Let $\omega \in \mathbb{Q}^n_{\geq 0}$ be a weight vector representing $\prec_1$. By assumption and Corollary 1.16, ω also represents $\prec_2$. A consequence of the proof of Theorem 1.13 is that $\omega \in \text{int}(C_{\prec_1}) \cap \text{int}(C_{\prec_2})$. As $C_{\prec_1}$ and $C_{\prec_2}$ are cones in a polyhedral fan (and any two distinct cones in a polyhedral fan intersect at the boundary), $C_{\prec_1} = C_{\prec_2}$ must hold. □

It may not be immediately clear that the set described in (9) corresponds to a polyhedral cone. In order to clarify this, as well as to simplify the description of computations in later chapters, we introduce the following notion

Definition 1.18. Let $G_\prec = \{g_1, ..., g_s\}$ be the marked Gröbner basis of I w.r.t $\prec$.
For each $i \in \{1, .., s\}$ we write

$$\text{LT}_\prec(g_i) = in_\prec(g_i) = x^{\alpha_i} \quad \text{and} \quad S'_{g_i} = \text{supp}(g_i) \setminus \{\alpha_i\}.$$

We call set of all vectors

$$\text{BV}(G_\prec) := \left\{ \alpha_i - \beta \quad \text{for } i \in \{1, ...s\}\ ,\ \beta \in S'_{g_i} \right\} \subset \mathbb{Z}^n$$

the **bounding vectors** of $G_\prec$ (or equivalently, $C_\prec$).

Example 1.19. For the marked Gröbner basis $G_\prec = \{\underline{y^6} - y^4, \underline{x^2} + y^3\}$ Example 1.15, the bounding vectors of $G_\prec$ are

$$\mathrm{BV}(G_\prec) = \left\{ \begin{pmatrix} 0 \\ 2 \end{pmatrix}, \begin{pmatrix} 1 \\ -3 \end{pmatrix} \right\}.$$

It is no coincidence that these vectors form an H-description of $C_\prec$.

Using this definition, the condition for inclusion in a Gröbner cone obtained from the proof of Theorem 1.13 may be reformulated as follows:

Proposition 1.20. *Let $x \in \mathbb{Q}^n_{\geq 0}$, $C_\prec$ be a Gröbner cone of an ideal I w.r.t the monomial order $\prec$ and* $\mathrm{BV}(G_\prec) := \{v_1, ..., v_m\}$ *be the corresponding bounding vectors. Then*

$$x \in C_\prec \qquad \Longleftrightarrow \qquad \langle x, v_j \rangle \geq 0 \qquad \textit{for all } j \in \{1, ..., m\}. \tag{14}$$

Furthermore:

$$x \in \partial C_\prec \iff \textit{equality holds for at least one } j.$$

Proof. This follows from the proof of Theorem 1.13 (Cf. [Stu95, pg. 4]). □

Consequently, the vectors in $\mathrm{BV}(G_\prec)$ give an H-description (cf. Theorem C.4) of the cone $C_\prec$. Moreover, there is following relationship between refinement and inclusion in a Gröbner cone:

Lemma 1.21. *Let $\omega \in \mathbb{Q}^n_{\geq 0}$ and $\prec$ be a monomial order. Then:*

$$\prec \ \textit{refines } \omega \implies \omega \in C_\prec.$$

Proof. Let $\prec$ be a refinement of ω , $g \in G_\prec$ and $x^\alpha = in_\prec(g)$. From Lemma 1.7 we know that $in_\prec(in_\omega(g)) = in_\prec(g)$, therefore $\alpha \in \mathrm{supp}(in_\omega(g))$. In particular $\langle \omega, \alpha \rangle \geq \langle \omega, \beta \rangle$ for all $\beta \in \mathrm{supp}(g)$ which proves $\omega \in C_\prec$ by Proposition 1.20. □

The converse does not hold in general. Consider the following example:

Example 1.22. Let $I = \langle x + y \rangle \lhd \mathbb{Q}[x, y]$. The Gröbner cone of I w.r.t the lexicographic ordering $lex_{x \succ y}$ is

$$C_{lex} = \{\omega \in \mathbb{Q}^2_{\geq 0} : \omega_1 \geq \omega_2\}.$$

In particular, we have that $(1, 1) \in C_{lex}$. However lex is not a refinement of the vector $(1, 1)$. For example, for $f = x^2 + xy^2$ we have that

$$xy^2 = in_\prec(in_\omega(f)) \neq in_\prec(f) = x^2.$$

A modified version of the converse does hold:

Lemma 1.23. *Let $\omega \in \mathbb{Q}^n_{\geq 0}$, $\prec$ be a monomial order, and $C_\prec$ be a Gröbner cone of some ideal I w.r.t $\prec$. Then:*

$$\omega \in C_\prec \iff C_\prec = C_{\prec_\omega}.$$

Proof. " $\Longleftarrow$ " If $C_\prec = C_{\prec_\omega}$, then $\omega \in C_\prec$ follows directly from Lemma 1.21 and the observation that $\prec_\omega$ refines ω.
For " $\Longrightarrow$ " assume $\omega \in C_\prec$. We show $C_\prec = C_{\prec_\omega}$ by showing

$$in_{\prec_\omega}(g) = in_\prec(g) \ \text{ for all } g \in G_\prec \ ,$$

which implies this direction by Corollary 1.17.
Let $in_{\prec_\omega}(g) = x^\alpha$. Then $\langle \omega, \alpha \rangle \geq \langle \omega, \beta \rangle$ for all other $\beta \in \mathrm{supp}(g)$. If we assume $in_\prec(g) \neq x^\alpha$ then $in_\prec(g) = x^\beta$ for some β with $\langle \omega, \beta \rangle < \langle \omega, \alpha \rangle$. However, this would imply $\langle \omega, \beta - \alpha \rangle < 0$ and therefore $\omega \notin C_\prec$ (by Proposition 1.20 as $(\beta - \alpha) \in \mathrm{BV}(G_\prec)$). This is a contradiction. Hence, $C_\prec = C_{\prec_\omega}$ holds. □

Combining this result with Lemma 1.21 yields the following equivalent condition:

Corollary 1.24. *([CKM97, Remark 2.3])*
Let I be an ideal, $G_{\prec}$ and $C_{\prec}$ be the Gröbner basis/cone of I w.r.t some monomial order $\prec$. For $\omega \in \mathbb{Q}^n_{\geq 0}$ the following holds:

$$\omega \in C_{\prec} \iff in_{\prec}(g) = in_{\prec_\omega}(g) = in_{\prec}(in_\omega(g)) \quad \textit{for all} \quad g \in G_{\prec}. \tag{15}$$

The Gröbner walk follows line segments which transverse cones in the Gröbner fan. For this reason, it is useful to describe what happens at points at which a line segment crosses into a new cone. This is done by the following result.

Proposition 1.25. *([CKM97, Proposition 2.4])*
Let I be an ideal, $\sigma, \tau \in \mathbb{Q}^n_{\geq 0}$ be weight vectors, and $\prec$ a monomial order such that $\prec$ refines τ. Then there exists a $\omega \in \overline{\sigma\tau}$, $\omega \neq \sigma$ such that

$$\overline{\sigma\omega} \subset C_{\prec_\sigma}.$$

Proof. The fact that $\sigma \in C_{\prec_\sigma}$ holds is a consequence of Corollary 1.24.
Let $G_{\prec_\sigma} = \{g_1, ..., g_r\}$ be the (marked) reduced Gröbner basis of I w.r.t $\prec_\sigma$. We fix an $i \in \{1, ..., r\}$ and write $in_{\prec_\sigma}(g_i) = x^{\alpha_i}$. Recall that the $\sigma-$tail of a polynomial f is $\text{tail}_\sigma(f) := f - in_\sigma(f)$ (cf. Definition 1.1).
As the monomial order $\prec_\sigma$ starts by comparing $\sigma-$degrees, we have that

$$\langle \sigma, \alpha_i - \beta \rangle > 0 \quad \text{for all} \quad \beta \in \text{supp}(\text{tail}_\sigma(g_i)).$$

Furthermore by continuity, there exists a point $\omega_i \in \overline{\sigma\tau}$, $\omega_i \neq \sigma$ such that

$$\langle \psi, \alpha_i - \beta \rangle > 0 \quad \text{for all} \quad \beta \in \text{supp}(\text{tail}_\sigma(g_i)) \ , \ \psi \in \overline{\sigma\omega_i}. \tag{16}$$

The proposition follows from the following claim.

<u>*Claim:*</u> $\quad \langle \psi, \alpha_i - \beta \rangle \geq 0 \quad$ for all $\psi \in \overline{\sigma\omega_i}$ for all $\beta \in \text{supp}(g_i)$.

Proof of claim: Fix an arbitrary $\psi \in \overline{\sigma\omega_i}$ and $\beta \in \text{supp}(g_i)$. For β, one of the following two cases holds:
If $\beta \in \text{supp}(\text{tail}_\sigma(g_i))$ then the claim is precisely (16).
If $\beta \in \text{supp}(in_\sigma(g_i))$, then $\langle \sigma, \alpha_i - \beta \rangle = 0$. Futhermore, due to the assumption that$\prec$ refines τ, we have that $\langle \tau, \alpha_i - \beta \rangle \geq 0$. Evidently $\psi \in \overline{\sigma\omega_i} \subset \overline{\sigma\tau}$, therefore (upon rewriting $\psi = (1-t)\sigma + t\tau$ for some $t \in (0, 1]$) we see that

$$\langle \psi, \alpha_i - \beta \rangle = \langle (1-t)\sigma + t\tau, \alpha_i - \beta \rangle = (1-t) \underbrace{\langle \sigma, \alpha_i - \beta \rangle}_{=0} + t \underbrace{\langle \tau, \alpha_i - \beta \rangle}_{\geq 0} \geq 0.$$

This proves the claim.
A consequence of the claim is that $\alpha_i \in \text{supp}(in_\psi(g_i))$, therefore $in_{\prec_\sigma}(in_\psi(g_i)) = x^{\alpha_i} = in_{\prec_\sigma}(g_i)$.
Finally, for each $i \in \{1, ..., r\}$ we write ω_i as $\omega_i = (1-t_i)\sigma + t_i\tau$, where ω_i is a point on $\overline{\sigma\tau}$ fulfilling (16). We subsequently set $\omega := (1-\hat{t})\sigma + \hat{t}\tau$, where $\hat{t} := \min\{t_1, ..., t_r\}$ Then for all $\psi \in \overline{\sigma\omega}$ and all $i \in \{1, ..., r\}$ we have that $in_{\prec_\sigma}(in_\psi(g_i)) = x^{\alpha_i} = in_{\prec_\sigma}(g_i)$. This implies $\omega \in C_{\prec_\sigma}$ by Corollary 1.24, which completes the proof. $\square$

Example 1.26. To illustrate the statement of Proposition 1.25, we consider once again the ideal $I = \langle x + y^3, x^2 + xy \rangle$, for which we have already computed the marked Gröbner basis w.r.t $lex_{x \succ y}$ in Example 1.15. This Gröbner basis corresponds to the cone with rays $(1, 0)$ and $(3, 1)$. We call this cone C_1. After computing the reduced Gröbner bases w.r.t *grevlex* and $lex_{y \succ x}$ (by applying Buchberger's algorithm and subsequently reducing), we end up with the marked Gröbner bases

$$G_{grevlex} = \{\underline{y^3} + x, \underline{x^2} + xy\} \qquad G_{lex_{y \succ x}} = \{\underline{x^4} - x^2, \underline{xy} + x^2, \underline{y^3} + x\}$$

with corresponding Gröbner cones

$$C_2 := C_{grevlex} := cone(\{(3,1),(1,1)\}) \quad C_3 := cone(\{(1,1),(1,0)\}),$$

computed via the H-description given by bounding vectors.
Taken together, these three cones cover $\mathbb{Q}^2_{\geq 0}$. Thus, they are the maximal cones of the Gröbner fan of I. In particular, the Gröbner cone of I w.r.t any other monomial order $\prec$ is one of these three cones. Now consider the two weight vectors $\sigma = (2,1)$ and $\tau = (0,1)$. $lex_{y \succ x}$ refines τ and indeed, it can be verified (via a Gröbner basis computation) that the Gröbner cone of I w.r.t $(lex_{y \succ x})_\sigma$ is C_2. Proposition 1.25 states that there exists a $\omega \in C_2$ such that $\overline{\sigma\omega} \subset C_2$. The ω closest to τ such that this is the case is $\omega = (1,1)$.

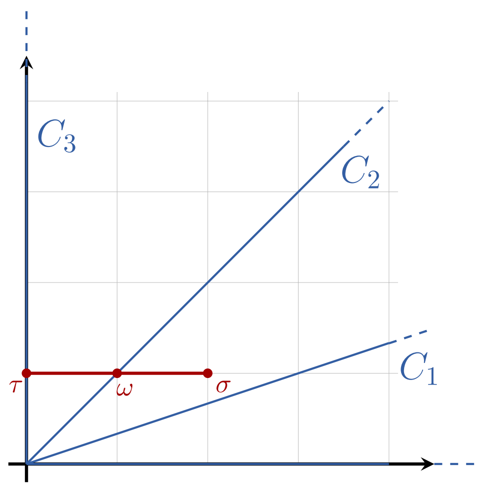


Figure 2: The three cones C_1, C_2, C_3 are the maximal cones of the Gröbner fan of I. By Proposition 1.25, there exists a ω such that $\overline{\sigma\omega} \subset C_2$.

1.3 Monomial orders and matrices

In computer algebra systems, monomial orders are encoded as matrices. In the last section of this chapter we state some facts about this correspondence which will be needed to describe the Gröbner walk. More information may be found in Appendix B.
Let $k, n \in \mathbb{N}$, $k \geq n$, and $A \in \mathbb{Q}^{k,n}$ be a matrix . We consider the relation $<_A$ on $\mathbb{Q}^n$ defined as follows:

$$\text{For } \; u, v, \in \mathbb{Q}^n : \qquad u <_A v \iff Au <_{lex} Av,$$

where lex denotes the lexicographic order on $\mathbb{Q}^k$ (cf. Definition B.1). In words, $<_A$ compares the entries of the vectors $Au, Av \in \mathbb{Q}^k$ until a tie is broken. If $rk(A) = n$, then it can easily be checked that $<_A$ is a strict total order on $\mathbb{Q}^n$ (cf. Lemma B.4). The relation between matrix orders and monomial orders is described by the following two propositions.

Proposition 1.27. *Let $k \geq n$ and $A \in \mathbb{Q}^{k,n}$ be a matrix of rank n such that its first row $a_1 \in \mathbb{Q}^n$ is not the zero vector and has non-negative entries. The relation $\prec_A$ on the set of all monomials of $\mathbb{Q}[x_1, ..., x_n]$ defined by*

$$x^\beta \prec_A x^\alpha \quad :\iff \quad \beta <_A \alpha \qquad \textit{for all} \;\; \alpha, \beta \in \mathbb{N}^n \tag{17}$$

is a monomial order.

Proof. Consult Proposition B.5. The restrictions on a_1 are required so that $\prec_A$ is a well-ordering. $\square$

Remark 1.28. In some literature, monomial orders are defined directly as total admissible well-orderings on the set of exponent vectors $\mathbb{N}^n$. The corresponding relation on the set of monomials $\mathrm{Mon}_n(x)$ is induced via the one-to-one correspondence between $\mathrm{Mon}_n(x)$ and $\mathbb{N}^n$. If one adopts this convention monomial order $\prec_A$ is precisely the restriction of $<_A$ onto $\mathbb{N}^n$.

Every matrix fulfilling the conditions of proposition 1.27 induces a monomial order. Conversely, any monomial order $\prec$ may be represented (in the sense of (17)) by such a matrix.

Proposition 1.29. *For any monomial order $\prec$, there exists a matrix $A \in \mathbb{Q}^{k,n}$ of rank n, $k \geq n$ and first row $a_1 \in \mathbb{Q}^n_{\geq 0}$ such that for all $\alpha, \beta \in \mathbb{N}^n$:*

$$x^\beta \prec x^\alpha \iff x^\beta \prec_A x^\alpha \iff \beta <_A \alpha. \tag{18}$$

Proof. Consult [Ovc02, pg.239] or [Rob85, pg.4]. $\square$

Definition 1.30. For a monomial order $\prec$, we call a matrix $A \in \mathbb{Q}^{k,n}$ fulfilling the conditions of Proposition 1.29 a **monomial order matrix** of $\prec$. We conventionally denote it by $A_\prec$.

Remark 1.31. The version of proposition 1.29 stated in [Ovc02] is slightly stronger. It states that, for any monomial order $\prec$, there exists an *invertible* matrix $A_\prec$ with *non-negative entries* such that (18) holds. Because of this, we will assume at times that $A_\prec$ has these additional properties. In matters of implementation one must proceed with caution, as monomial orders are not represented by such matrices by default.

Remark 1.32. The correspondence between monomial orders and matrices is not one-to-one. As an example, consider the lexicographic ordering on 3 variables. An obvious choice for a monomial order matrix would be the identity matrix I_3. However, the matrices

$$\begin{pmatrix} 2 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix}, \quad \begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \\ 1 & 1 & 1 \end{pmatrix}, \quad \begin{pmatrix} 2 & 0 & 0 \\ 0 & 0 & -1 \\ 0 & -1 & 0 \end{pmatrix},$$

are all alternative choices for monomial order matrices for this ordering.

We conclude with two more results about monomial order matrices which will be required in Section 2.

Proposition 1.33. *Let $\prec$ be a monomial order on $\mathbb{Q}[x_1, ..., x_n]$ and $A \in \mathbb{Q}^{k,n}$ be a monomial order matrix of $\prec$. Let $a_1 \in \mathbb{Q}^n_{\geq 0}$ denote the first row of A. Then $\prec$ refines a_1 in the sense of definition 1.4. That is:*

$$\langle a_1, \beta \rangle < \langle a_1, \alpha \rangle \implies x^\beta \prec x^\alpha \quad \text{for all } \alpha, \beta \in \mathbb{N}^n.$$

In particular, $a_1 \in C_\prec$ holds for any ideal $I \lhd \mathbb{Q}[x_1, ..., x_n]$.

Proof. $a_1 \in \mathbb{Q}^n_{\geq 0}$ holds by construction. The implication above follows immediately from the definition of monomial order matrix. $a_1 \in C_\prec$ follows from Lemma 1.21. $\square$

Corollary 1.34. *Let $\prec$ be a monomial order with monomial order matrix $A_\prec$, and $\omega \in \mathbb{Q}^n_{\geq 0}$ be a weight vector. A monomial order matrix for the refinement $\prec_\omega$ is given by the matrix*

$$A_{\prec_\omega} = \begin{pmatrix} \omega \\ A \end{pmatrix} \in \mathbb{Q}^{k+1,n}.$$

Proof. This also follows directly from the definitions of $A_\prec$ and $\prec_\omega$. The monomial order $\prec_\omega$ first compares $\omega-$degrees and then breaks ties with $\prec$. $\square$

Example 1.35. A commonly encountered monomial order is the graded lexicographic order *glex*. It may be seen as the lexicographic ordering refined by the vector $\omega = \mathbf{1} = (1, ..., 1)$. A matrix for the lexicographic order is given by the identity matrix. Thus, a monomial order matrix for *glex* is given by

$$\begin{pmatrix} \mathbf{1} \\ I_n \end{pmatrix} = \begin{pmatrix} 1 & 1 & \cdots & 1 \\ 1 & 0 & \cdots & 0 \\ 0 & 1 & \ddots & \vdots \\ \vdots & \ddots & \ddots & 0 \\ 0 & \cdots & 0 & 1 \end{pmatrix} \in \mathbb{Q}^{n+1,n}.$$

.

We now have all the tools necessary to describe the Gröbner walk as it was described in [CKM97].

2 The algorithm

The Gröbner walk is a Gröbner basis conversion algorithm. Its goal is to complete the following task:

> *Given an ideal I and the reduced Gröbner basis of I w.r.t $\prec$,*
> *compute the reduced Gröbner basis of I w.r.t some other monomial order $\prec'$*

Throughout this text we will refer to $\prec$ and $\prec'$ as the *starting* and *target monomial* orders respectively. In accordance with Proposition 1.29, they may be represented by matrices $A_{\prec}$ and $A_{\prec'} \in \mathbb{Q}^{k,n}$ with $k \geq n$ and full row rank. The algorithm requires a starting vector $\sigma \in C_{\prec}$ as well as a target vector $\tau \in C_{\prec'}$. By Proposition 1.33, σ and τ can be taken to be the first rows $A_{\prec}$ and $A_{\prec'}$ respectively. Abstractly, the algorithm can be described through the following steps:

- Starting at σ, proceed on the line segment $\overline{\sigma\tau}$ until reaching a point $\omega \in \partial C_{\prec}$ on the boundary of the starting cone.
- Convert the set of initial forms $in_\omega(G_{\prec})$ to a Gröbner basis M of $in_\omega(I)$ with respect to $\prec'_\omega$.
- "Lift" M to a Gröbner basis G of I w.r.t $\prec'_\omega$.
- Reduce G to the marked Gröbner basis $G_{\prec'_\omega}$.
- If $\tau \in C_{\prec'_\omega}$, return $G_{\prec'_\omega}$. Otherwise set $\sigma := \omega$, $A_{\prec} := A_{\prec'_\omega}$ and repeat the steps above.

The goal for this chapter is to formalize these steps and prove the correctness of the algorithm.

2.1 The standard Gröbner walk

A formal algorithm for the procedure described above is as follows:

Algorithm 1 StandardGroebnerWalk($G_{\prec}$, $A_{\prec}$, $A_{\prec'}$)

```
Input: G≺                                    ▷ the marked Gröbner basis of I w.r.t ≺
       A≺ and A≺'                            ▷ monomial order matrices for ≺ and≺'

Output: G≺'                                  ▷ the marked Gröbner basis of I w.r.t ≺'

  σ ← (A≺)_{1,·}                             ▷ A weight vector refined by ≺
  τ ← (A≺')_{1,·}                            ▷ A weight vector refined by ≺'
  done ← "False"
  while done = "False" do

     ω ← GetNextW(G≺, σ, τ)                  ▷ the weight vector at which στ exits C≺
     G' ← Lift(G≺, ω, τ)                     ▷ outputs a Gröbner basis of I w.r.t ≺'_ω
     G' ← Reduce(G')                         ▷ interreduction

     if ω = τ then
        done ← "True"                        ▷ if we have reached τ, terminate outputting G'

     else                                    ▷ otherwise, update the starting Gröbner basis/order to ≺'_ω
        σ ← ω
        G≺ ← G'
        A≺ ← A≺'_ω                           ▷ the new starting monomial order is ≺'_ω
     end if
  end while
  return G'
```

The correctness/ termination of StandardGroebnerWalk depends on the correctness/termination of the GetNextW and Lift subroutines. We therefore start by describing these.

2.2 Computing the next weight vector

We fix an ideal I and a monomial order $\prec$. Let $\mathrm{BV}(G_\prec) = \{v_1, ..., v_m\}$ denote the bounding vectors of $G_\prec$.

Lemma 2.1. *Let I be an ideal, $\prec$ be a monomial order, $C_\prec$ be the Gröbner cone of I w.r.t $\prec$, and $\sigma, \tau \in \mathbb{Q}^n_{\geq 0}$ be weight vectors. We assume $\sigma \in C_\prec$ and $\tau \notin C_\prec$. The point $\omega \in \partial C_\prec \cap \overline{\sigma\tau}$ at which the line segment $\overline{\sigma\tau}$ exits $C_\prec$ is a weight vector, and may be written as $\omega = (1-\hat{u})\sigma + \hat{u}\tau$, where*

$$\hat{u} := \max\Big\{ u \in [0,1] : (1-u)\langle\sigma, v_i\rangle + u\langle\tau, v_i\rangle \geq 0 \ \text{ for all } i \in \{1, ..., m\} \Big\}. \tag{19}$$

Proof. The first observation follows from Proposition 1.20; upon writing a point $x \in \overline{\sigma\tau}$ as $x = (1-u)\sigma + u\tau$ for some $u \in [0,1)$, the condition (14) for inclusion in a Gröbner cone states that

$$x \in C_\prec \iff (1-u)\langle\sigma, v_i\rangle + u\langle\tau, v_i\rangle \geq 0 \ \text{ for all } i \in \{1, ..., m\}.$$

Clearly, the point furthest along $\overline{\sigma\tau}$ such that the condition on the right hand side holds is $(1-\hat{u})\sigma + \hat{u}\tau$. Finally, the point $\omega := (1-\hat{u})\sigma + \hat{u}\tau$ is a weight vector as it is a linear combination of weight vectors with non-negative rational coefficients. □

The following result characterizes $\hat{u}$. It was first mentioned without proof in [CLO05, pg.437].

Proposition 2.2. *Let I, $\prec$, $G_\prec$, τ and σ be such as in the setting of Lemma 2.1. We define the set*

$$M := \{v_i \in \mathrm{BV}(G_\prec) \mid \langle\tau, v_i\rangle < 0\} \ . \tag{20}$$

Then the quantity $\hat{u}$ defined in (19) *is*

$$\hat{u} = \min_{v_i \in M} \left\{ \frac{\langle\sigma, v_i\rangle}{\langle\sigma, v_i\rangle - \langle\tau, v_i\rangle} \right\}. \tag{21}$$

Proof. Let $M := \{v_1, ..., v_k\} \subseteq \mathrm{BV}(G_\prec)$ be defined as in (20). The assumption $\tau \notin C_\prec$ implies $M \neq \emptyset$ by Proposition 1.20, therefore $k \geq 1$ holds. For each $i \in \{1, .., k\}$ we define the quantity $u_i := \frac{\langle\sigma, v_i\rangle}{\langle\sigma, v_i\rangle - \langle\tau, v_i\rangle}$. We observe that

$$(1-u_i)\langle\sigma, v_i\rangle + u_i\langle\tau, v_i\rangle = \langle\sigma, v_i\rangle - \frac{\langle\sigma, v_i\rangle^2}{\langle\sigma, v_i\rangle - \langle\tau, v_i\rangle} + \frac{\langle\sigma, v_i\rangle\langle\tau, v_i\rangle}{\langle\sigma, v_i\rangle - \langle\tau, v_i\rangle} = 0 \ \text{ for all } i.$$

Now assume w.l.o.g that the minimum over all u_i is attained at u_1. In particular $\hat{u} = u_1$, and $(1-u_1)\langle\sigma, v_1\rangle + u_1\langle\tau, v_1\rangle = 0$. Moreover, for all $v_i \in \mathrm{BV}(G_\prec)$ the following holds:

$$\begin{aligned}(1-\hat{u})\langle\sigma, v_i\rangle + \hat{u}\langle\tau, v_i\rangle &\geq (1-u_i)\langle\sigma, v_i\rangle + \hat{u}\langle\tau, v_i\rangle & (22)\\ &\geq (1-u_i)\langle\sigma, v_i\rangle + u_i\langle\tau, v_i\rangle = 0. & (23)\end{aligned}$$

The inequalities at (22) and (23) may be justified as follows:
If $\langle\tau, v_i\rangle \geq 0$ then (22) and (23) are both trivial. (Recall that $\langle\sigma, v_i\rangle \geq 0$ holds due to $\sigma \in C_\prec$.)
If $\langle\tau, v_i\rangle < 0$ then $\hat{u} \leq u_i$ implies both $\hat{u}\langle\tau, v_i\rangle \geq u_i\langle\tau, v_i\rangle$ and $(1-\hat{u})\langle\sigma, v_i\rangle \geq (1-u_i)\langle\sigma, v_i\rangle$.

Thus, Proposition 1.20 implies $(1-\hat{u})\sigma + \hat{u}\tau \in \partial C_\prec$. Finally, note that for any $u' \in [0,1]$ with $u' > \hat{u}$ we have

$$(1-u')\langle\sigma, v_1\rangle + u'\langle\tau, v_1\rangle < (1-\hat{u})\langle\sigma, v_1\rangle + \hat{u}\langle\tau, v_1\rangle = 0 \ , \tag{24}$$

where the inequality follows analogously to (22) and (23) above. Again by Proposition 1.20, this would imply $(1-u')\sigma + u'\tau \notin C_\prec$. Hence, the quantity defined in (21) is precisely the one defined in (19). This concludes the proof. □

Proposition 2.2 provides us with a method for computing the next weight vector ω.

Algorithm 2 GETNEXTW($G_\prec$, σ, τ)

Input: $G_\prec$ ▷ a marked Gröbner basis
σ ▷ a starting weight vector $\sigma \in C_\prec$
τ ▷ a target weight vector $\tau \in C_{\prec'}$

Output: ω ▷ The next weight vector in the Gröbner walk

```
BV ← BV(G_≺)                        ▷ The bounding vectors of G_≺
û ← 1
for v_j in BV do
    if ⟨τ, v_j⟩ < 0 then              ▷ Check if v_j ∈ M
        u_j ← ⟨σ,v_j⟩ / (⟨σ,v_j⟩ − ⟨τ,v_j⟩)
        if u_j < û then
            û ← u_j                   ▷ return the smallest u_j
        end if
    end if
end for
return (1 − û)σ + ûτ
```

If the conditions for Proposition 2.2 are fufilled, then the output of "GETNEXTW" is precisely the $\hat{u}$ defined in (21). In order for Algorithm 1 "STANDARDGROEBNERWALK" to avoid being stationary we must ensure that $\hat{u} > 0$ holds. The following two results describe when this is the case.

Lemma 2.3. *Let σ,τ and $\prec$, $\prec'$ weight vectors and monomial orders respectively, such that σ is the first row of $A_\prec$ and τ is the first row of $A_{\prec'}$. We assume $\tau \notin C_\prec$. Let $\hat{u} \in [0, 1]$ be the quantity defined in* (19)*. Then:*

$$C_\prec = C_{\prec'_\sigma} \implies \hat{u} > 0.$$

Proof. We prove this by contraposition. If $\hat{u} = 0$, then there exists a $v \in \mathrm{BV}(G_\prec)$ such that

$$\langle \sigma, v\rangle = 0 \qquad \text{and} \qquad \langle \tau, v\rangle < 0. \tag{25}$$

Now write v once again as $v = \alpha - \beta$, where $x^\alpha = in_\prec(g)$ and $\beta \in \mathrm{supp}(g - in_\prec(g))$ for some $g \in G_\prec$. Then we may rewrite (25) as

$$\langle \sigma, \alpha\rangle = \langle \sigma, \beta\rangle \qquad \text{and} \qquad \langle \tau, \alpha\rangle < \langle \tau, \beta\rangle\ . \tag{26}$$

But now because τ is taken to be the first row of the matrix of the target monomial order $A_{\prec'}$ (26) implies that $in_\prec(g) = x^\alpha \neq x^\beta = in_{\prec'_\sigma}(g)$. (The monomial order $\prec'_\sigma$ compares σ-degrees first, and then τ-degrees) By Corollary 1.17 $C_\prec \neq C_{\prec'_\sigma}$ holds, proving " $\implies$ " by contraposition. □

In Algorithm 2 "GETNEXTW", the restriction from Lemma 2.3 that $\tau \notin C_\prec$ holds is omitted. This is done on purpose to ensure a termination condition, the correctness of which is given by the following lemma.

Lemma 2.4. *Let $\prec$ be a monomial order, and σ and τ be weight vectors with $\sigma \in C_\prec$. Then the following holds:*

$$\tau \in C_\prec \implies \text{GETNEXTW}(G_\prec,\ \sigma,\ \tau) = \tau\ .$$

Proof. If $\tau \in C_\prec$ then $\langle \tau, v\rangle \geq 0$ for all $v \in \mathrm{BV}(G_\prec)$. Therefore the first "if" condition of Algorithm 2 is never fulfilled. It follows that $\hat{u} = 1$ holds, implying GETNEXTW($G_\prec$, σ, τ) $= \tau$. □

Remark 2.5. In the setting of Algorithm 1, if $\sigma \in \partial C_{\prec}$ then $C_{\prec} = C_{\prec'_\sigma}$ may not hold the very first time "GETNEXTW" is called in the while loop. In this case it may be that the output of "GETNEXTW" is $\hat{u} = 0$, in which case GETNEXTW($G_{\prec}$, σ, τ)$= \sigma$ would hold. (An example of this is seen in Example 2.6.) However, the STANDARDGROEBNERWALK does not remain stationary: the end of the first while loop, the starting monomial order/reduced Gröbner basis of the second iteration are updated to $G_{\prec'_\sigma}$ and $\prec'_\sigma$. Clearly $\sigma \in C_{\prec'_\sigma}$ and it is the first row of $A_{\prec'_\sigma}$. Thus, at every call of "GETNEXTW" in STANDARDGROEBNERWALK, the conditions for Lemma 2.3 are fulfilled at every at every iteration of the while loop of Algorithm 1 with possible exception of the very first one.

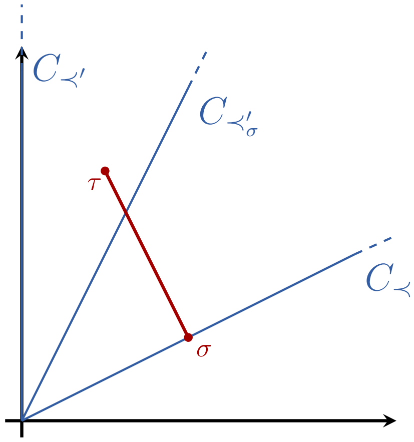


Figure 3: If σ lies on the boundary of $C_{\prec}$, then $C_{\prec} \neq C_{\prec'_\sigma}$ may hold. It is a consequence of Proposition 1.25 that upon leaving $C_{\prec}$, the line segment $\overline{\sigma\tau}$ enters $C_{\prec'_\sigma}$.

Example 2.6. The marked Gröbner basis of the ideal $I = \langle x^2 + yz, xy + z^2\rangle$ with respect to the graded lexicographic ordering is

$$G_{grevlex} = \{\underline{xy} + z^2, \underline{x^2} + yz, \underline{y^2z} - xz^2\}.$$

$\sigma = (1, 1, 1)$ is a natural choice of starting vector for any Gröbner walk with this starting monomial order. In the step-by-step example in Section 3.1 we will see that

$$G_{\prec'_\sigma} = \{\underline{xy} + z^2, \underline{x^2} + yz, \underline{xz^2} - y^2z, \underline{y^3z} + z^4\} \neq G_{grevlex},$$

implying $C_{\prec} \neq C_{\prec'_\sigma}$ by Corollary 1.17. Furthermore, the output of GETNEXTW($G_{grevlex}$, σ, τ) is once again σ.

We now turn our attention to the lifting step.

2.3 The lifting step

In this section we assume that we have computed the weight vector ω at which $\overline{\sigma\tau}$ leaves the starting cone and discuss how to use this to obtain a Gröbner basis of the Gröbner cone we have just entered. A consequence of Proposition 1.25 is that once the line segment $\overline{\sigma\tau}$ exits $C_{\prec}$ at $\omega \in \overline{\sigma\tau}$ it enters $C_{\prec'_\omega}$. This is because $\prec'_\omega$ is a refinement of $\prec'$. Now we would like to compute a Gröbner basis of I w.r.t $\prec'_\omega$. Proposition 1.9 implies that after computing ω we get a reduced Gröbner basis of the ideal of initial forms $in_\omega(I)$ "for free":

Lemma 2.7. *Let I be an ideal and $\prec$ a monomial order. For $G_{\prec} = \{g_1, ..., g_r\}$ and $\omega \in C_{\prec} \cap \mathbb{Q}^n_{\geq 0}$, the set*

$$in_\omega(G_{\prec}) = \{in_\omega(g_1), ..., in_\omega(g_r)\}$$

is the marked Gröbner basis of $in_\omega(I)$ w.r.t $\prec$.

Proof. $\omega \in C_\prec$, so by Lemma 1.23 $C_{\prec_\omega} = C_\prec$. This in turn implies $G_\prec = G_{\prec_\omega}$ by Corollary 1.17. Thus, $in_\omega(G_\prec) = in_\omega(G_{\prec_\omega})$ is the marked Gröbner basis of $in_\omega(I)$ w.r.t $\prec$ by Proposition 1.9. $\square$

The idea now is to convert $in_\omega(G_\prec)$ to a reduced Gröbner basis of the ideal of initial forms $in_\omega(I)$ with respect to the new monomial order $\prec'_\omega$ and subsequently "lift" it to a Gröbner basis of I. The correctness of this procedure is proven via the following two results:

Lemma 2.8. *Let $\omega \in C_\prec$ and $M = \{m_1, ..., m_s\}$ be the reduced Gröbner basis of $in_\omega(I)$ with respect to $\prec'_\omega$. For all $i \in \{1, ..., s\}$ there exists an $r \in \mathbb{N}$ and polynomials $h_{i1}, ..., h_{ir} \in \mathbb{Q}[x_1, ..., x_n]$ such that*

$$m_i = \sum_{j=1}^{r} h_{ij} in_\omega(g_j), \tag{27}$$

and the following 3 statements hold for all $j \in \{1, ..., r\}$ with $h_{i,j} \neq 0$:

(i) $in_\prec(h_{ij} in_\omega(g_j)) \prec in_\prec(m_i)$ *or* $in_\prec(h_{ij} in_\omega(g_j)) = in_\prec(m_i)$.

(ii) $\deg_\omega(m_i) = \deg_\omega(h_{ij} in_\omega(g_j))$.

(iii) Every polynomial h_{ij} is ω-homogeneous.

Proof. Fix an $i \in \{1, ..., s\}$. As the polynomials $in_\omega(G_\prec)$ form a Gröbner basis of $in_\omega(I)$ w.r.t $\prec$, the representation of m_i as in (27) can be obtained by dividing m_i by the elements of $in_\omega(G_\prec)$ w.r.t $\prec$. By Proposition 1.9 $in_\omega(G_\prec)$ is a Gröbner basis of $in_\omega(I)$ w.r.t $\prec$, therefore by the correctness of the division algorithm (cf. Theorem A.10), a representation in which (i) holds exists. (This is sometimes referred to as a *standard representation*).

Points *(ii)* and *(iii)* are consequences of the division algorithm in the specific setting in which both m_i and the polynomials $\{in_\omega(g_i), .., in_\omega(g_r)\}$ are ω-homogeneous. (This is the case here due to Theorem A.24, as both sets are reduced Gröbner bases of the homogeneous ideal $in_\omega(I)$.) To prove this rigorously, set $a := \deg_\omega(m_i)$ and $b_j := \deg_\omega(in_\omega(g_j))$ for $j \in \{1, ..., r\}$.

At the first iteration of the division algorithm, for some j, a polynomial of the form $c_\gamma x^\gamma in_\omega(g_j)$ is subtracted from m_i to eliminate the leading term $\mathrm{LT}_\prec(m_i)$, where

$$c_\gamma x^\gamma = \frac{\mathrm{LT}_\prec(m_i)}{\mathrm{LT}_\prec(in_\omega(g_j))}.$$

Thus, Remark 1.3 implies

$$\deg_\omega(c_\gamma x^\gamma) = a - b_j \quad \text{and} \quad \deg_\omega(c_\gamma x^\gamma in_\omega(g_j)) = a = \deg_\omega(m_i). \tag{28}$$

The term $c_\gamma x^\gamma$ is added to h_{ij}.
Observe that the "new" polynomial in the second iteration $m_i - c_\gamma x^\gamma(in_\omega(g_j))$ remains $\omega-$homogeneous of degree a. It is evident that at every subsequent step of the division algorithm, if the leading term of the "new" polynomial is divisible by $in_\omega(g_j)$, then a term of ω-degree $a - b_j$ must be multiplied with $in_\omega(g_j)$ and subtracted from m_i. This term is then added to h_{ij}.
Consequently, for each j, the polynomial h_{ij} is either zero or $\omega-$homogeneous with degree $a - b_j$. Combined with (28), this implies both *(ii)* and *(iii)*.

$\square$

We now use this representation of the elements of M to obtain a Gröbner basis of I w.r.t $\prec'_\omega$.

Theorem 2.9. *Let $G_\prec = \{g_1, ..., g_r\}$ be the reduced Gröbner basis of an ideal I w.r.t $\prec$, $\omega \in C_\prec$, and $M = \{m_1, ...m_s\}$ be the reduced Gröbner basis of $in_\omega(I)$ with respect to $\prec'_\omega$. For each $i \in \{1, ..., s\}$ we write*

$$m_i = \sum_{j=1}^{r} h_{ij} in_\omega(g_j) \ ,$$

where $h_{i1}, ...h_{ir} \in \mathbb{Q}[x_1, ..., x_n]$ have the properties from Lemma 2.8.
We subsequently define the "lifted" polynomial

$$f_i := \sum_{j=1}^{r} h_{ij} g_j \in I. \tag{29}$$

Then the following holds:

$$G := \{f_1, ..., f_s\} \quad \text{is a Gröbner basis of } I \text{ w.r.t } \prec'_\omega.$$

Proof. We start by observing that for $i \in \{1, ..., s\}$ the initial form $in_\omega(f_i)$ is precisely m_i:

$$in_\omega(f_i) = in_\omega(\sum_{j=1}^{r} h_{ij} g_j) = \sum_{j=1}^{r} h_{ij} in_\omega(g_j) = m_i \ , \tag{30}$$

where the second inequality holds due to points (ii) and (iii) of Lemma 2.8: all non-zero polynomials h_{ij} are homogeneous, and the polynomials $h_{ij} in_\omega(g)$ have equal $\omega-$degree. Thus, the terms of f_i that vanish upon taking $in_\omega(f_i)$ are precisely those of the $\omega-$tail of g_i.
To see that G forms a Gröbner basis, we observe that

$$\begin{aligned}
in_{\prec'_\omega}(G) &= \langle \{in_{\prec'_\omega}(f_1), ...in_{\prec'_\omega}(f_s)\}\rangle \\
&= \langle \{in_{\prec'}(in_\omega(f_1)), ...in_{\prec'}(in_\omega(f_s))\}\rangle \\
&= \langle \{in_{\prec'}(m_1), ..., in_{\prec'}(m_s)\}\rangle \qquad (31)\\
&= \langle \{in_{\prec'}(in_\omega(m_1)), ...in_{\prec'}(in_\omega(m_s))\}\rangle \\
&= in_{\prec'_\omega}(M) \\
&= in_{\prec'_\omega}(in_\omega(I)) \qquad (32)\\
&= in_{\prec'}(in_\omega(in_\omega(I))) = in_{\prec'_\omega}(I).
\end{aligned}$$

The equality (31) follows from the observation (30). (32) holds as M is a Gröbner basis of $in_\omega(I)$ w.r.t $\prec'_\omega$. Furthermore, we used that $in_{\prec'_\omega}(f) = in_{\prec'}(in_\omega(f))$ for all $f \neq 0$ (cf. (5)) multiple times. Per definition, G is a Gröbner basis of I w.r.t $\prec'_\omega$. □

We prove two properties of the lifted Gröbner basis G which will be needed later.

Corollary 2.10. *In the setting of Theorem 2.9, let $M = \{m_1, ..., m_r\}$ be the reduced Gröbner basis of $in_\omega(I)$ w.r.t $\prec'_\omega$. Then the lifted Gröbner basis $G = \{f_1, ..., f_r\}$ of I w.r.t $\prec'_\omega$ has the following properties:*

(i) G is inclusion minimal.

(ii) For all $i \in \{1, ..., r\}$: $in_{\prec'_\omega}(f_i) = in_{\prec'_\omega}(m_i)$.

Proof. For (i): Recall that $in_\omega(f_i) = m_i$ for all i. If we assume that G is not inclusion minimal then (w.log) $\langle\{f_2, ..., f_r\}\rangle = I$. This would imply $\langle\{in_\omega(f_2), ..., in_\omega(f_r)\}\rangle = \langle\{m_2, ..., m_r\}\rangle = in_\omega(I)$, contradicting the fact that M is reduced (and in particular inclusion minimal).

For (ii): $in_{\prec'_\omega}(f_i) = in_{\prec'}(in_\omega(f_i)) = in_{\prec'}(m_i) = in_{\prec'_\omega}(m_i)$ follows once again from (5). $\square$

The following corollary of Theorem 2.9 gives us a straightforward method for computing the lifted Gröbner basis:

Corollary 2.11. *Let $G = \{f_1, ..., f_r\}$ be the set of polynomials defined in* (29) *of Theorem 2.9. For each $i \in \{1, ..., r\}$, f_i may be written as*

$$f_i = m_i - \overline{m_i}^{\prec},$$

where $\overline{m_i}^{\prec}$ is the normal form of m_i w.r.t $\prec$ and I (cf. Definition A.14).

Proof. As in Theorem 2.9, let $G_\prec = \{g_1, ..., g_r\}$ be the reduced Gröbner basis of I w.r.t $\prec$. Our very first observation is that

$$\omega \in C_\prec \implies C_\prec = C_{\prec_\omega} \implies G_\prec = G_{\prec_\omega}$$

by Corollary 1.24 and Corollary 1.17. Consequently, dividing any polynomial by $\{in_\prec(g_1), ..., in_\prec(g_r)\}$ w.r.t $\prec$ is equivalent to dividing by $\{in_{\prec_\omega}(g_1), ..., in_{\prec_\omega}(g_r)\}$ w.r.t $\prec_\omega$. Because of this, we prove the equivalent statemtent $f_i = m_i - \overline{m_i}^{\prec_\omega}$ for all $i \in \{1, ..., s\}$.
Fix an $i \in \{1, .., r\}$. Rewriting m_i as in (27), we see that

$$m_i - f_i = \sum_{j=1}^{r} h_{ij} in_\omega(g_j) - \sum_{j=1}^{r} h_{ij} g_j = \sum_{j=1}^{r} -h_{ij}\,\mathrm{tail}_\omega(g_j), \tag{33}$$

where $\mathrm{tail}_\omega(g_j) := g_j - in_\omega(g_j)$. Thus, the statement we would like to prove is equivalent to showing

$$\overline{m_i}^{\prec_\omega} = \sum_{j=1}^{r} -h_{ij}\,\mathrm{tail}_\omega(g_j).$$

Proposition A.12 states that the residue $\overline{m_i}^{\prec_\omega}$ is uniquely determined by two properties:

(i) There exists a $p \in I$ such that $m_i = p + \overline{m_i}^{\prec_\omega}$.

(ii) No term of $\overline{m_i}^{\prec_\omega}$ is divisible by any of the monomials in the set $\{in_{\prec_\omega}(g_1), ...in_{\prec_\omega}(g_r)\}$.

Therefore, if we can show that $\sum_{j=1}^{r} -h_{ij}\,\mathrm{tail}_\omega(g_j)$ possesses both of these properties, we are done.

For (i), note that (similarly to above):

$$m_i = f_i + \sum_{j=1}^{r} -h_{ij}\,\mathrm{tail}_\omega(g_j) = \underbrace{\sum_{j=1}^{r} h_{ij} g_j}_{\in I} + \sum_{j=1}^{r} -h_{ij}\,\mathrm{tail}_\omega(g_j)$$

so $\sum_{j=1}^{r} -h_{ij}\,\mathrm{tail}_\omega(g_j)$ fulfills *(i)* .

For (ii), we first claim that $\sum_{j=1}^{r} h_{ij}\,\mathrm{tail}_\omega(g_j) = \overline{f_i}^{in_\omega(I),\prec}$, where $\overline{f_i}^{in_\omega(I),\prec}$ is the remainder obtained by dividing f_i by the marked Gröbner basis $\{in_\omega(g_1), ..., in_\omega(g_r)\}$ of $in_\omega(I)$. (Recall that $G_\prec = G_{\prec_\omega}$ implies that this set is a Gröbner basis of $in_\omega(I)$ w.r.t $\prec$ by Proposition 1.9.). In general, the division algorithm gives us

$$f_i = \sum_{j=1}^{r} p_{ij} in_\omega(g_j) + \overline{f_i}^{in_\omega(I),\prec}$$

for some $p_{i1}, ..., p_{ij} \in \mathbb{Q}[x_1, ..., x_n]$. However, we may take $p_{ij} = h_{ij}$, where the h_{ij} are a representation of m_i defined in (27) / Lemma 2.8. This follows somewhat technically from the multivariate division algorithm; in the first step of the algorithm, the leading term

$$in_{\prec_\omega}(f_i) = in_\prec(in_\omega(f_i)) = in_\prec(m_i)$$

is eliminated by subtracting some polynomial of the form $c_\gamma x^\gamma in_\omega(g_j)$ for $j \in \{1, ..., r\}$. This is precisely the multiple of $in_\omega(g_j)$ required to eliminate the term $in_\prec(m_i)$ of m_i when reducing m_i w.r.t $\prec$ (by dividing by elements of $in_\omega(G_\prec)$): thus, x^γ is a term of the polynomial h_{ij} from (27) for some $j \in \{1, ..., r\}$. A similar argument for subsequent steps of the division algorithm implies that setting $p_{ij} = h_{ij}$ obtains one possible representation of f_i. Consequently we may write

$$f_i = \sum_{j=1}^{r} h_{ij} in_\omega(g_j) + \overline{f_i}^{in_\omega(I),\prec} = m_i + \overline{f_i}^{in_\omega(I),\prec}. \tag{34}$$

Therefore (combining with (33)) we have that

$$\overline{f_i}^{in_\omega(I),\prec} = f_i - m_i = \sum_{j=1}^{r} h_{ij} \,\mathrm{tail}_\omega(g_j).$$

By correctness of the division algorithm, no term of the residue $\overline{f_i}^{in_\omega(I),\prec}$ is divisible by any of the monomials $\{in_\prec(in_\omega(g_1)), ..., in_\prec(in_\omega(g_r))\} = \{in_{\prec_\omega}(g_1), ..., in_{\prec_\omega}(g_r)\}$. Thus, $\sum_{j=1}^{r} h_{ij} \,\mathrm{tail}_\omega(g_j)$ has property (ii) as well. This concludes the proof.

□

We can now describe the steps of the "Lift" subroutine:

Algorithm 3 LIFT($G_\prec$, ω, τ)

Input: $G_\prec$ ▷ The starting marked Gröbner basis
ω ▷ A weight vector in $\mathbb{Q}^n_{\geq 0} \cap C_\prec$
τ ▷ The target vector (of which $\prec'$ is a refinement)

Output: G ▷ A Gröbner basis of I w.r.t $\prec'_\omega$

```
inwG ← in_ω(G_≺)
M ← Buchberger(inwG, ≺'_ω)          ▷ compute a Gröbner basis of in_ω(G) w.r.t ≺'_ω
for m in M do
    r ← m̄^{I,≺}                     ▷ subtract from each m ∈ M its normal form w.r.t G_≺
    m ← m - r
end for
return M
```

The correctness of this procedure is a direct consequence of Corollary 2.11.

2.4 Proof of correctness

We now have all the results necessary to prove the correctness of Algorithm 1.

Theorem 2.12. *Algorithm 1* StandardGroebnerWalk *is correct and terminates after finitely many iterations of the while loop.*

Proof. For readability, we present the algorithm once again on the following page. We prove the correctness of the algorithm via a series of case distinctions.

Trivial conversion: If $\prec'$ and $\prec$ are such that $C_\prec = C_{\prec'}$, then in particular $\tau \in C_\prec$. Thus, Lemma 2.4 implies GetNextW$(G_\prec,\ \sigma,\ \tau) = \tau$. Corollary 1.17 implies that $G_\prec = G_{\prec'} = G_{\prec'_\tau}$, therefore the lifting step makes no changes to $G_\prec$: the Gröbner basis of initial forms M of $in_\tau(I)$ from Algorithm 3 is $in_\tau(G_\prec)$ itself, and subtracting normal forms returns once again $G_\prec = G_{\prec'}$. After this, "done" gets set to True and the algorithm terminates, outputting $G_\prec$. Indeed, this is the desired output, as equality of the Gröbner cones implies $G_\prec = G_{\prec'}$ by Corollary 1.17.

Initialization: At the very first call of the while loop, either $\tau \in C_\prec$ holds (in which case Lemma 2.4 implies GetNextW$(G_\prec,\ \sigma,\ \tau) = \tau$) or the conditions for Lemma 2.3 are fulfilled. In either case, Remark 2.5 implies that from the second iteration of the while-loop onwards, the weight vector $\omega \in \overline{\sigma\tau}$ computed by "GetNextW" is distinct from the starting vector σ, and closer to τ. For the newly computed ω, one of the following two cases holds:

Case 1: $\omega \in C_{\prec'}$

Then Lemma 1.23 implies $C_{\prec'_\omega} = C_{\prec'}$. Thus, Lift$(G_\prec\ \omega,\ \tau)$ computes a Gröbner basis with respect to $\prec'$ and reducing it gives us $G_{\prec'_\omega} = G_{\prec'}$. The next iteration of the while loop is the "trivial conversion" setting described above: "GetNextW" returns τ, so "done" is set to True and the algorithm terminates.

Case 2: $\omega \notin C_{\prec'}$

In this case, the reduced Gröbner basis computed in the rest of the iteration is $G_{\prec'_\omega} \neq G_{\prec'}$ (due to Corollary 1.17), so we are not yet done. (Indeed, in this case $\omega \neq \tau$, so "done" remains false). Thus, the algorithm reitarates after updating the starting vector, order and Gröbner basis: The starting vector σ is set to ω and $G_\prec$ is set to $G_{\prec'_\omega}$ (i.e. the reduced Gröbner basis obtained upon applying a reduction algorithm to the output of Lift$(G_\prec,\ \omega,\ \tau)$). The matrix $A_\prec$ is updated to $A_{\prec'_\omega}$ which may be obtained by taking the matrix $A_{\prec'}$ and adding the first row ω (cf. Corollary 1.34)

Termination: As the Gröbner cones form a polyhedral fan, the line segment $\overline{\sigma\tau}$ has a non-empty intersection with finitely many full-dimensional cones $C_1, ..., C_k \in \mathrm{GFan}(I)$. Due to convexity (cf. Remark C.3), each intersection $\overline{\sigma\tau} \cap C_i$ is either a line segment or a point. This (combined with the fact that $\hat{u} \geq 0$ holds at every call of GetNextW) implies that upon exiting a cone C_i, our path along $\overline{\sigma\tau}$ does not reenter it at a later stage. Thus, τ is reached after finitely many steps, implying that the algorithm terminates.

□

Algorithm 1 STANDARDGROEBNERWALK($G_{\prec}$, $A_{\prec}$, $A_{\prec'}$)

Input: $G_{\prec}$ ▷ the marked Gröbner basis of I w.r.t $\prec$
$A_{\prec}$ and $A_{\prec'}$ ▷ monomial order matrices for $\prec$ and$\prec'$

Output: $G_{\prec'}$ ▷ the marked Gröbner basis of I w.r.t $\prec'$

```
σ ← (A_≺)_{1,·}                          ▷ a weight vector refined by ≺
τ ← (A_≺')_{1,·}                         ▷ a weight vector refined by ≺'
done ← "False"
while done = "False" do

    ω ← GetNextW(G_≺, σ, τ)              ▷ the weight vector at which \overline{στ} exits C_≺
    G' ← Lift(G_≺, ω, τ)                 ▷ outputs a Gröbner basis of I w.r.t ≺'_ω
    G' ← Reduce(G')                      ▷ interreduction

    if ω = τ then
        done ← "True"                    ▷ if we have reached τ, terminate and output G'

    else                                 ▷ otherwise, update the starting Gröbner basis/order to ≺'_ω
        σ ← ω
        G_≺ ← G'
        A_≺ ← A_≺'_ω                     ▷ the new starting monomial order is ≺'_ω
    end if
end while
return G'
```

3 First Examples

To see the algorithm in action, we carry out several Gröbner basis conversions using Algorithm 1 "StandardGroebnerWalk" on the ideal

$$I = \langle x^2 + yz, xy + z^2 \rangle \lhd \mathbb{Q}[x, y, z]. \tag{35}$$

3.1 Converting to Lex

In this subsection, we pick the graded reverse lexicographic order as our starting monomial order. We apply Buchberger's algorithm to the generators and find that the marked Gröbner basis with respect to this order is

$$G_{\prec} = G_{grevlex} = \{\underline{xy} + z^2, \underline{x^2} + yz, \underline{y^2z} - xz^2\}.$$

A monomial order matrix for $\prec$ is given by

$$A_{\prec} = \begin{pmatrix} 1 & 1 & 1 \\ 0 & 0 & -1 \\ 0 & -1 & 0 \end{pmatrix}. \tag{36}$$

We want to use Algorithm 1 to compute the marked Gröbner basis of I with respect to the lexicographic order, which we represent with the monomial order matrix

$$A_{\prec'} = \begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix}. \tag{37}$$

We go through the algorithm step by step:

- the initialization sets $\sigma = (1, 1, 1)$, $\tau = (1, 0, 0)$, and *done* = "False".
- We then call GetNextW($G_{\prec}$, σ, τ). This involves computing the bounding vectors $\mathrm{BV}(G_{\prec})$, which are
$$\mathrm{BV}(G_{\prec}) = \{(1, 1, -2), (2, -1, -1), (-1, 2, -1)\} =: \{v_1, v_2, v_3\}. \tag{38}$$
  We observe that $\langle \tau, v_i \rangle < 0$ holds only for $i = 3$. So we set
$$\hat{u} = \frac{\langle \sigma, v_3 \rangle}{\langle \sigma, v_3 \rangle - \langle \tau, v_3 \rangle} = \frac{0}{0 - 1} = 0\ .$$
  It follows that our "next" weight vector ω is once again σ. (This is an example of the situation described in remark 2.5.)
- Next, we call Lift($G_{\prec}$, ω, τ). This first computes the set of initial forms
$$in_{\omega}(G_{\prec}) = \{xy + z^2, x^2 + yz, y^2z - xz^2\}$$
  which coincides with $G_{\prec}$ itself; this is no coincidence, as our ideal I is $(1, 1, 1)$-homogeneous. "Lift" then computes the Gröbner basis of $in_{\omega}(G_{\prec})$ with respect to the monomial order $\prec'_{\sigma}$, which is encoded as the matrix
$$A_{\prec'_{\sigma}} = \begin{pmatrix} 1 & 1 & 1 \\ 1 & 0 & 0 \\ 0 & 1 & 0 \end{pmatrix}.$$
  This is
$$M = \{\underline{x^2} + yz, \underline{xy} + z^2, \underline{xz^2} - y^2z, \underline{y^3z} + z^4\} =: \{m_1, m_2, m_3, m_4\}.$$

We then subtract $\overline{m_i}^{\prec}$ from each m_i and reduce (in this case, both of these steps are trivial) and obtain

$$G' = M = \{\underline{x^2} + yz, \underline{xy} + z^2, \underline{xz^2} - y^2z, \underline{y^3z} + z^4\}.$$

Because the weight vector computed in this iteration is $\omega = (1, 1, 1) \neq \tau$, we are not yet done. Therefore, we update the starting vector/Gröbner basis to $\sigma := \omega = (1, 1, 1)$ and $G_\prec := G'$ and go again.

- In this iteration $\text{GetNextW}(G_\prec, \sigma, \tau)$ first computes the bounding vectors

$$\text{BV}(G') = \{(2, -1, -1), (1, 1, -2), (1, -2, 1), (0, 3, -3)\} =: \{v_1, v_2, v_3, v_4\}.$$

  This time we observe that $\langle \tau, v_i \rangle \geq 0$ for all i, and therefore $\hat{u} = 1$, implying $\omega = \tau$.

- $\text{Lift}(G_\prec, \tau, \tau)$ starts by computing the initial forms

$$in_\omega(G_\prec) = \{xy, x^2, xz^2, y^3z + z^4\}\ ,$$

  which already form a Gröbner basis of the initial ideal w.r.t $\prec'_\omega$. Therefore $M = in_\omega(G')$ and lifting to I gives us once again

$$G' = G_\prec = \{\underline{x^2} + yz, \underline{xy} + z^2, \underline{xz^2} - y^2z, \underline{y^3z} + z^4\},$$

  which is also already reduced w.r.t $\prec'_\omega$.

- As $\omega = \tau$, *done* is set to "True" and the algorithm terminates, outputting

$$G' = \{\underline{x^2} + yz, \underline{xy} + z^2, \underline{xz^2} - y^2z, \underline{y^3z} + z^4\} = G_{lex}.$$

The point of this first example is more to verify the correctness of StandardGroebnerWalk than to underline any particular advantage of the walk over Buchberger's algorithm: the path followed was almost trivial, and the computation of M in the first call of "Lift" is tantamount to converting $G_{grevlex}$ to G_{lex} with Buchberger's algorithm. The fact that the algorithm in this case was, in effect, "Buchberger with extra steps" is in part because the corresponding cones $C_{grevlex}$ and C_{lex} share a face of codimension 2, and our chosen starting vector σ lies on this face. The goal of the next example is to illustrate the importance of the path chosen.

3.2 Changing paths

Let us now carry out the same conversion with a different path. That is, we start with the same starting Gröbner basis as before:

$$G_\prec = G_{grevlex} = \{\underline{xy} + z^2, \underline{x^2} + yz, \underline{y^2z} - xz^2\} =: \{g_1, g_2, g_3, g_4\}.$$

Our goal is once again to compute a Gröbner basis with respect to *lex*. However, now we choose to represent these monomial orders by the matrices

$$A_\prec = \begin{pmatrix} 1 & 1 & 0 \\ 1 & 1 & 1 \\ 0 & 0 & -1 \end{pmatrix} \quad \text{and} \quad A_{\prec'} = \begin{pmatrix} 3 & 1 & 0 \\ 1 & 0 & 0 \\ 0 & 1 & 0 \end{pmatrix}$$

respectively. Because of this, at initialization $\sigma := (1, 1, 0)$ and $\tau := (3, 1, 0)$. These are vectors lying in the interiors of $C_\prec$ and $C_{\prec'}$ respectively. We present a less verbose description of the steps of StandardGroebnerWalk below.

- At initialization, $\sigma = (1, 1, 0)$, $\tau = (3, 1, 0)$, *done* = "False".

- GetNextW($G_\prec$, σ, τ) computes the same set $\mathrm{BV}(G_\prec) = \{v_1, v_2, v_3\}$ as in (38). This time $\langle \tau, v_i \rangle$ is only negative for $i = 3$, at which $\langle \tau, v_i \rangle = -1$. Consequently $\hat{u} = \frac{1}{2}$ and GetNextW($G_\prec$, σ, τ) $= (2, 1, 0) = \omega$.
- Lift($G_\prec$, ω, τ) first computes the initial forms
$$in_\omega(G_\prec) = \{xy, x^2, y^2 - xz^2\}$$
and converts them to the following Gröbner basis of $in_\omega(I)$ with respect to $\prec'_\omega$:
$$M = \{xy, x^2, y^2z - xz^2, y^3z\} = \{m_1, m_2, m_3, m_4\}.$$
- We observe that for $i \in \{1, 2, 3\}$, $m_i = in_\omega(g_i) \in in_\omega(G_\prec)$. Therefore subtracting the normal forms $\overline{m_i}^{\prec}$ amounts to retrieving the corresponding elements $g_i \in G_\prec$.
For m_4, we have the following:
$$m_4 = z^2(xy) + y(y^2z - xz^2) = z^2 in_\omega(g_1) + y(in_\omega(g_3).$$
Thus, the fourth element of the Gröbner basis w.r.t $\prec'_\omega$ is
$$z^2g_1 + yg_3 = z^2(xy + z^2) + y(y^2z - xz^2) = y^3z + z^4.$$
So ultimately GetNextW($G_\prec$, σ, τ) outputs the marked Gröbner basis
$$G' = \{g_1, g_2, g_3, g_4\} = \{\underline{xy} + z^2, \underline{x^2} + yz, \underline{xz^2} - y^2z, \underline{y^3z} + z^4\}.$$
- As $\omega \neq \tau$, we reiterate with starting vector $\sigma := \omega = (2, 1, 0)$ and starting Gröbner basis $G_\prec := G'$
- In this iteration, GetNextW($G_\prec$, $(2, 1, 0)$, $(3, 1, 0)$) computes the same bounding vectors as in the second iteration of the previous example:
$$\mathrm{BV}(G_\prec) = \{(1, 1, -2), (2, -1, -1), (1, -2, 1), (0, 3, -3)\} =: \{v_1, v_2, v_3, v_4\}$$
As $\langle \tau, v_i \rangle \geq 0$ for all $i = 1, ..., 4$, GetNextW($G_\prec$, σ, τ) $= \tau$, and therefore *done* gets set to "True".
- The remaining computations are now identical to the previous example (Sharp-sighted readers will have noticed that G' is precisely G_{lex}) and StandardGroebnerWalk terminates after one more while loop, outputting the same Gröbner basis as the one computed in Section 3.1.

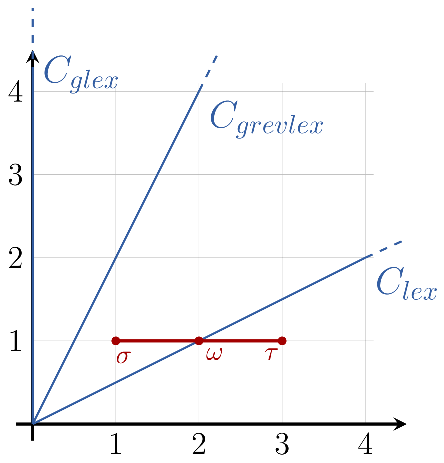


Figure 4: The "new" path from $\sigma = (1, 1, 0)$ to $\tau = (3, 1, 0)$ viewed in the $\omega_3 = 0$ plane. The line segment crosses into the target cone C_{lex} at $\omega = (2, 1, 0)$.

We observe that modifying the path changes the computations in a non-trivial way. With both choices of path, Buchberger's algorithm is called once on a set of the form $in_\omega(G_\prec)$. In the first example, this amounted to calling it on $G_\prec$ itself, whereas in the second case the initial forms are "proper" and consist almost entirely of monomials, implying a more efficient conversion (and subsequent reduction) of M. Clearly the computational advantage in this example is negligible; however, it is enough to illustrate the principle behind the Gröbner walk's strategy: instead of one "heavy" conversion with Buchberger's algorithm we carry out many "light" intermediate conversions, which amount to Buchberger's algorithm on sets that consist mainly of monomials. A discussion of how to choose the path such that the lengths of the initial forms are minimal is the topic of Section 4.

3.3 Same ideal, different steps

We perform one last computation on the same ideal to illustrate an example where the path crosses into more than one Gröbner cone. To this end, we consider once again

$$I = \langle x^2 + yz, xy + z^2 \rangle,$$

but now take the lexicographic ordering *lex* to be the starting monomial order. That is,

$$G_\prec = \{\underline{x^2} + yz, \underline{xy} + z^2, \underline{xz^2} - y^2z, \underline{y^3z} + z^4\}$$

is our starting Gröbner basis. In this example, we convert $G_\prec$ to the Gröbner basis with respect the monomial order $glex_{(1,3,0)}$ (the graded lexicographic order, refined by the matrix $(1,3,0)$) using StandardGroebnerWalk. Following Proposition 1.33he standard choice of monomial order matrix for the refinement order $\prec'$ is

$$A_{\prec'} = \begin{pmatrix} 1 & 3 & 0 \\ 1 & 1 & 1 \\ 1 & 0 & 0 \end{pmatrix}. \tag{39}$$

We report only the outputs of the subroutines.

- Upon intialization, $\sigma := (1,0,0)$, $\tau := (1,3,0)$ and *done* = "False" .
- GetNextW($G_\prec$, σ, τ) $= (1, \frac{1}{2}, 0) =: \omega$.
- Lift computes $in_\omega(G_\prec) = \{x^2, xy, xz^2 - y^2z, y^3z\}$ and converts it to the following Gröbner basis M of $in_\omega(I)$ w.r.t $\prec'_\omega$, which has the monomial order matrix

  $$A_{\prec'_\omega} = \begin{pmatrix} 1 & \frac{1}{2} & 0 \\ 1 & 3 & 0 \\ 1 & 1 & 1 \end{pmatrix}.$$

  This is

  $$M = \{\underline{y^2z} - xz^2, xy, x^2\} =: \{m_1, m_2, m_3\},$$

  which is subsequently lifted to a basis of I by computing $f_i := m_i - \overline{m_i}^\prec$ for each i. This gives us

  $$G' = \{\underline{y^2z} - xz^2, \underline{xy} + z^2, \underline{x^2} + yz\}$$

  which is already a marked Gröbner basis of I w.r.t $\prec'_\omega$.
- *done* $\neq$ "True", so we update our starting parameters to $\sigma := \omega$ and $G_\prec := G'$ and go again.
- GetNextW($G_\prec$, σ, τ) $= (1, 2, 0) =: \omega$

- "LIFT" computes $in_\omega(G_\prec) = \{y^2z, xy, x^2 + yz\}$ and converts it to

$$M' := \{x^3, xy, \underline{x^2} + yz\},$$

which is a Gröbner basis of $in_{\omega'}(I)$ w.r.t the monomial order $\prec'$, refined by ω'. The corresponding matrix is:

$$A_{\prec'_{\omega}{}'} = \begin{pmatrix} 1 & 2 & 0 \\ 1 & 3 & 0 \\ 1 & 0 & 0 \end{pmatrix}.$$

(Importantly, we note that the weight vector $(1, \frac{1}{2}, 0)$ computed in the previous iteration is no longer a row of this matrix.)

- Subtracting the remainders $\overline{m_i}^{G_\prec}$ from the m_is in this case retrieves

$$G' = \{\underline{x^3} - z^3, \underline{xy} + z^2, \underline{x^2} + yz\},$$

which is already reduced w.r.t $\prec'_\omega$. $done \neq$ "False" still holds, so we set reiterate after updating $G_\prec := G'$ and $\sigma := \omega$.

- GETNEXTW($G_\prec$, ω, τ) $= \tau$, therefore we set $done =$ "True" and perform the final computations:

Now, $in_\tau(G_\prec) = \{x^3, xy, yz\}$ consists of monomials, and it is quick to check that it is the reduced Gröbner basis of $in_\tau(G_\prec)$ w.r.t $\prec'_\tau$. In particular $G_{\prec'} = G_{\prec'_\tau}$ therefore (analogously to the last step of the previous example) subtracting the normal forms retrieves $G_{\prec'}$ once again. Reducing is trivial up to updating the marking $in_{\prec'}(g_3) =: yz$, after which the algorithm terminates, outputting

$$G_{\prec'} = \{\underline{x^3} - z^3, \underline{xy} + z^2, \underline{yz} + x^2\}.$$

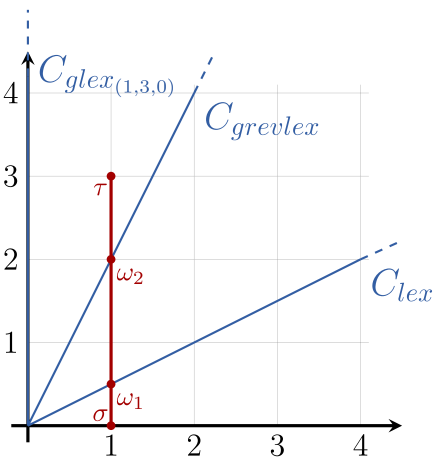


Figure 5: The path in the Gröbner fan from $\sigma = (1, 0, 0)$ to $\tau = (1, 3, 0)$ viewed in the $\omega_3 = 0$ plane, as well as the intermediate weight vectors ω_1 and ω_2

.

3.4 A first modification: integer weight vectors

In certain computer algebra software, including `Macaulay2` and `OSCAR`, monomial orders may only be specified by matrices/weight vectors with integer entries. Because of this, a common modification in implementations of StandardGroebnerWalk is to scale the output of GetNextW by the greatest common divisor of its entries. ([AGK97] refers to this as the "Zig-Zag" walk.)

Algorithm 4 GetNextIntegerW($G_\prec$, σ, τ)

Input: $G_\prec$ ▷ a marked Gröbner basis w.r.t $\prec$
σ ▷ a starting weight vector in $C_\prec$
τ ▷ A target weight vector
Output: ω ▷ The next weight vector in the walk, scaled so that it is in $\mathbb{Z}^n_{\geq 0}$

BV $\leftarrow$ BV($G_\prec$)
$\hat{u} \leftarrow 1$
for v_j in BV **do**
  if $\langle \tau, v_j \rangle < 0$ **then**
    $u_j \leftarrow \frac{\langle \sigma, v_j \rangle}{\langle \sigma, v_j \rangle - \langle \tau, v_j \rangle}$
    if $u_j < \hat{u}$ **then**
      $\hat{u} \leftarrow u_j$
    end if
  end if
end for
$\omega \leftarrow (1 - \hat{u})\sigma + \hat{u}\tau$ ▷ compute next ω along $\overline{\sigma\tau}$
$\omega \leftarrow \frac{1}{gcd(\omega)}\omega$ ▷ scale for integer entries
return ω

"$gcd(\omega)$" is the greatest common divisor of the entries of ω and may be computed in $O(\log n)$ time. For $\omega \in \mathbb{Q}^n_{\geq 0}$ this factor is greater than zero, implying that $\frac{1}{gcd(\omega)}\omega$ lies in precisely the same Gröbner cones as ω. An important consequence is that the modified path transverses exactly the same Gröbner cones as the original path as a combination of Proposition 1.20 and Proposition 1.25. This means that the intermediate conversions using Buchberger's algorithm are exactly same in the standard and Zig-Zag walks. The only downside is that the modification implies that intermediate weight vectors no longer lie on the line segment $\overline{\sigma\tau}$.

Example 3.1. In the conversion from Section 3.3, replacing "GetNextW" with "GetNextIntegerW" yields the intermediate weight vectors $\omega_1 = (2, 1, 0)$ and $\omega_2 = (1, 2, 0)$.

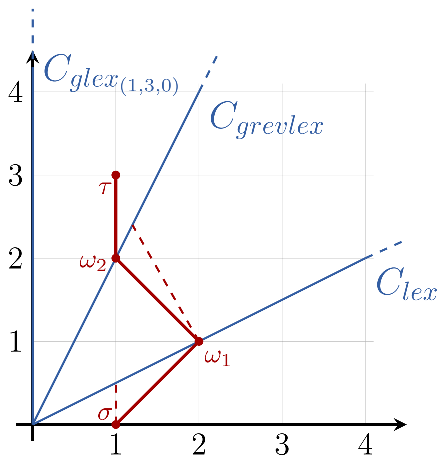


Figure 6: The path the algorithm takes in the conversion from Section 3.3 after replacing GetNextW with GetNextIntegerW. The dashed lines correspond to the paths to τ traced prior to scaling.

4 Path perturbation

The goals of this section are to witness the advantages of the Gröbner walk in action and introduce further improvements made possible by optimizing the choice of path. The statement and proof of Proposition 4.1, as well as the description and proof of correctness of the algorithms in Section 4.3 are novel contributions. The results from Section 4.1 and Section 4.2 are taken primarily from [Tra00] and [Fuk+07]; we added more detail to the proofs, and stated them with notation consistent with ours.

Throughout this chapter, we shall consider the following task as a running example:

Given the ideal

$$I = \langle\ 6 + 3x^3 + 16x^2z + 14x^2y^3,\ \ 6 + y^3z + 17x^2z^2 + 7xy^2z^2 + 13x^3z^2\ \rangle \tag{40}$$

compute the reduced Gröbner basis of I w.r.t the lexicographic ordering with $x \succ y \succ z$.

Despite consisting of only two polynomials over three variables, solving this problem by applying Buchberger's algorithm directly on the generators of I is slow. The main bottleneck consists of the computation and reduction of S-pairs; we encounter S-pairs of degree up to 46, which each have up to 466 terms and coefficients of order up to 10^{590}. Using the in-built method `gb` in `Macaulay2` with our benchmarking machine (cf. Section 6) took approximately 80 seconds.

In contrast, the computation of the reduced basis w.r.t *grevlex* with `gb` took just 0,28 seconds. An intuitive approach to the task using the Gröbner walk is therefore to first compute $G_{grevlex}$ and then walk from $C_{grevlex}$ to C_{lex}. This can be done by calling
StandardGroebnerWalk($G_{grevlex}$, $A_{\prec}$, $\prec'$), with $A_{\prec}$ and $A_{\prec'}$ defined as in (36)/(37). The algorithm terminates in approximately 10 seconds, almost one order of magnitude faster than pure Buchberger. There are 17 iterations of the while loop, and although each of these iterations entails a Gröbner basis conversion, these conversions are significantly faster due to the fact that they are performed on a set of truncated polynomials of the form $in_\omega(G)$ instead of G itself.

Upon closer inspection, one notices that the majority of the time expended for Algorithm 1 is spent at the last two iterations of the while loop: Subsequently, the lifting step computes a Gröbner basis with respect to *lex* consisting of polynomials with up to 466 terms. The most computationally expensive step is the subsequent reduction of this lifted Gröbner basis which involves reducing S-pairs with between 44 and 466 terms. This expends more than half of the total computation time. In comparison, the maximal length of the initial forms encountered in the previous intermediate conversions was 8, implying that these reductions were computationally much lighter (the complexity of Buchberger depends on the length of the support of the polynomials). We shall see that a contributing factor to these heavy final conversions is that the target vector $\tau = (1, 0, 0)$ lies on a face of codimension ≥ 2.

As seen in the previous chapter, the initial forms $in_\omega(g_i)$ encountered at each intermediate conversion depend on the choice of path between $C_{\prec}$ and $C_{\prec'}$. The intermediate conversions are at their lightest when the path from $C_{\prec}$ to $C_{\prec'}$ is chosen in such a way that the length of the initial forms $in_\omega(G_{\prec})$ is minimal. We know from Proposition 1.20 that if $\omega \in \mathrm{int}(C_{\prec})$ then
$in_\omega(G_{\prec}) = \{in_{\prec}(g_1), ..., in_{\prec}(g_s)\}$. In particular, the set $in_\omega(G)$ consists solely of monomials. The following result may be considered as a generalization of this statement to weight vectors which lie on the boundary of Gröbner cones.

Proposition 4.1. *Let I be an ideal, $\prec$ a monomial order, $G_\prec = \{g_1, ..., g_s\}$ the marked Gröbner basis of I w.r.t $\prec$, $C_\prec$ the corresponding Gröbner cone, and $\omega \in \partial C_\prec \cap \mathbb{Q}^n_{\geq 0}$ be a weight vector on the boundary of $C_\prec$.*
The length of the initial forms $in_\omega(G_\prec) = \{in_\omega(g_1), ...in_\omega(g_s)\}$ (i.e. $|\operatorname{supp}(in_\omega(g))|$) is minimal on the interior of the <u>*facets*</u> *of $C_\prec$ in the following sense:*

> *If $\omega \in \operatorname{relint}(F)$ for some face F of $C_\prec$ of codimension ≥ 2, then there exists a facet F' such that $F \subset F'$ and for all $\omega' \in \operatorname{relint}(F')$ we have that*
>
> $$in_\omega(g_i) = in_{\omega'}(g_i) + h_i \quad \text{for all} \quad i = 1, ..., s$$
>
> *where h_i is a (possibly empty) sum of terms in the $\omega-$tail of g_i. Futhermore, $h_i \neq 0$ holds for at least one $i \in \{1, ..., s\}$.*

Proof. Let F be a face of $C_\prec$ of codimension ≥ 2 and $\omega \in \operatorname{relint} F$. We may reformulate the claim in terms of inclusions of supports. It suffices to show that there exists a facet F' with $F \subset F'$ such for all $\omega' \in \operatorname{relint}(F)$:

$$\operatorname{supp}(in_{\omega'}(g_i)) \subseteq \operatorname{supp}(in_\omega(g_i)) \quad \text{for all} \quad i = 1, .., s\ ,$$

where strict inclusion holds for at least one i.
Let $\operatorname{BV}(G_\prec) = \{v_1, ..., v_m\}$ be the bounding vectors of $G_\prec$. By Proposition 1.20, the bounding vectors form an H-description of $C_\prec$ in the sense that

$$C_\prec = \big\{x \in \mathbb{R}^n : \langle x, v_i\rangle \geq 0 \quad \text{for all} \quad i \in \{1, ..., m\}\big\},$$

where some of the inequalities may be redundant.

We may assume without loss of generality that the first k elements of $\operatorname{BV}(G_\prec)$ give a minimal H-description of $C_\prec$ [1]. That is, the matrix with rows consisting of the first k bounding vectors.

$$M := \begin{bmatrix} v_1 \\ v_2 \\ \vdots \\ v_k \end{bmatrix} \in \mathbb{Z}^{k,n}$$

is of full rank (as $C_\prec$ is full-dimensional) and the Gröbner cone $C_\prec$ may be written as

$$C_\prec = \{x \in \mathbb{R}^n : Mx \in \mathbb{R}^k_{\geq 0}\}.$$

In this setting, any facet $F' \subset C_\prec$ may be described as

$$F' = \{x \in \mathbb{Q}^n_{\geq 0} : Mx \in \mathbb{R}^k_{\geq 0}\} \cap H_{v_j}$$

for some $j \in \{1, .., k\}$, where $H_{v_j} := \{x \in \mathbb{R}^n : \langle x, v_j\rangle = 0\}$ is the hyperplane with normal vector v_j. Similarly, any proper face $F \subset F'$ may be obtained by intersecting with further hyperplanes of this form. That is,

$$F = F' \cap H_{v_{i1}} \cap H_{v_{i2}} \cap ... \cap H_{v_{il}}$$

for some $i_1, ..., i_l \in \{1, ..., k\} \setminus \{j\}$. For simplicity, we consider the case $l = 1$ and denote $v_{i1} := v_l$. In other words, we assume F is a face of codimension 2 contained in F' (The $l > 1$ case follows analogously). Then for any $\omega \in \operatorname{relint} F$, we have that

$$\langle \omega, v_l\rangle = \langle \omega, v_j\rangle = 0 \quad \text{and} \quad \langle \omega, v_i\rangle > 0 \quad \text{for all} \quad i \in \{1, ..., k\} \setminus \{j, l\}. \tag{41}$$

[1] The task of determining a minimal H-description is dual to that of determining the vertex set of the convex hull of a point configuration.

Due $v_l \in \mathrm{BV}(G_\prec)$, there exists (by Definition 1.18) a polynomial $g \in G_\prec$ such that $v_l = \alpha - \beta$, where $in_\prec(g) = x^\alpha$ and $\beta \in \mathrm{supp}(g) \setminus \{\alpha\}$. The statement (41) implies $\langle \omega, \alpha \rangle = \langle \omega, \beta \rangle$, which in turn implies $\beta \in in_\omega(g_i)$.

On the other hand, for any $\omega' \in \mathrm{relint}(F')$ we have that

$$\langle \omega', v_j \rangle = 0 \quad \text{and} \quad \langle \omega', v_i \rangle > 0 \quad \text{for all} \quad i \in \{1, ..., k\} \setminus \{j\}. \tag{42}$$

In particular, $\langle \omega', v_l \rangle > 0$, and it follows that $\langle \omega', \alpha \rangle > \langle \omega', \beta \rangle$.
Thus, $\beta \in \mathrm{supp}(in_\omega(g)) \setminus \mathrm{supp}(in_{\omega'}(g))$. By comparing the conditions (41) and (42) we see that

$$\mathrm{supp}(in_{\omega'}(g)) \subset \mathrm{supp}(in_\omega(g)) \quad \text{for all} \quad g \in G_\prec\ ,$$

and from $\beta \in \mathrm{supp}(in_\omega(g)) \setminus \mathrm{supp}(in_{\omega'}(g))$ it follows that strict inclusion holds for at least one g. This completes the proof. $\square$

For any two full-dimensional cones C_1 and C_2 in $\mathbb{R}^n$ which intersect at a common facet F, a generic line segment between two points $x \in \mathrm{int}\, C_1$ and $y \in \mathrm{int}\, C_2$ intersects F in its relative interior $\mathrm{relint}(F)$. This, combined with Proposition 4.1 suggests that it may be advantageous to initialize the Gröbner walk in such a way that the starting and target vectors σ and τ fulfill $\sigma \in \mathrm{int}(C_\prec)$ and $\tau \in \mathrm{int}(C_{\prec'})$, in order to ensure intersections on low-codimension faces. This practice is called *path perturbation* and was first suggested in [AGK97]. Their ideas were later expanded on by [Tra00] and [Fuk+07].

In the example (40), modifying the steps of the Gröbner walk from $C_{grevlex}$ to C_{lex} by setting $\sigma = (6, 3, 2) \in \mathrm{int}(C_{grevlex})$ and $\tau = (445, 32, 1) \in \mathrm{int}(C_{lex})$ changed the nature and computations of StandardGroebnerWalk in the following way:

Starting vector σ	Target vector τ	number of conversions	maximal $\lvert\mathrm{supp}(in_\omega(g))\rvert$	maximal coeff. length	time (s., approx)
$(1,1,1)$	$(1,0,0)$	17	44	130	11.01
$(1,1,1)$	$(445,32,1)$	54	2	1147	4.26
$(6,3,2)$	$(1,0,0)$	27	44	138	11.42
$(6,3,2)$	$(445,32,1)$	56	2	1147	3.92

Table 1: Properties and durations of StandardGroebnerWalk in `Macaulay2` with varying choices of starting and target vectors.

We observe that the significant improvements on performance are seen when we modify the target vector. This is in accordance with our observations at the beginning of this chapter, as this modification avoids the bottlenecks which occur at the very last conversion. In our running example, the interior vectors $(6, 3, 2) \in \mathrm{int}(C_{grevlex})$ and $(445, 32, 1), \in \mathrm{int}(C_{lex})$ were computed after computing corresponding Gröbner cones. The aim of the following two sections is to discuss how such interior weight vectors may be computed *without* the Gröbner cones being known ahead of time.

4.1 Perturbed weight vectors

In the setting of StandardGroebnerWalk, the marked Gröbner basis with respect to the starting monomial order $\prec$ is known at every iteration. Via the bounding vectors $\mathrm{BV}(G_\prec)$ we obtain an H- description of the corresponding cone $C_\prec$. An interior starting vector $\sigma \in \mathrm{int}(C_\prec)$ can then be computed with linear methods (By Proposition 1.20, it suffices to determine a vector $\sigma \in \mathbb{Q}^n_{\geq 0}$ such that $\langle \sigma, v \rangle > 0$ for all $v \in \mathrm{BV}(G_\prec)$).

Conversely, the Gröbner cone of the target monomial order is not known in advance. Therefore we would like a method of determining an interior point of a Gröbner cone without its H-description being given. This is possible via the monomial order matrix of $\prec'$.

For a matrix $A \in \mathbb{Q}^{k \times n}$ of full row rank, recall the *matrix ordering* on $\mathbb{Q}^n$ defined via the following relation (cf. Definition B.3):

$$\text{For } \; u, v, \in \mathbb{Q}^n, \qquad u <_A v \iff Au <_{lex} Av.$$

Lemma 4.2. *Let $A \in \mathbb{Q}^{k \times n}$ be a matrix of full row rank. We denote the rows of A by $\omega_i \in \mathbb{Q}^n$ for $i \in \{1, .., k\}$. Let $V \subset \mathbb{Q}^n$ be a finite set of non-zero vectors such that:*

$$0 <_A v \quad \text{ for all } \; v \in V.$$

Then there exists a $\delta > 0$ such that for all $\varepsilon \in (0, \delta)$, the following holds:

$$\langle \omega_1 + \varepsilon\omega_2 + \varepsilon^2\omega_3 + ... + \varepsilon^{k-1}\omega_k, v\rangle > 0 \;\text{ for all }\; v \in V.$$

Proof. We fix n and prove the claim by induction over k.
If $A = (\omega_1) \in \mathbb{Q}^{1,n}$ then the left-hand side of (4.2) is ω_1 itself. In this case $\langle \omega_1, v\rangle > 0$ for all $v \in V$ follows directly from the assumption $Av >_{lex} 0$.
Now assume the statement holds for $k-1$. We start by defining the set

$$V' := \big\{ v \in V \;:\; \langle \omega_{k-1}, v\rangle > 0 \big\}.$$

We write $V' = \{v_1, ..., v_j\}$. For each $i \in \{1, ...j\}$ there exists a $\delta_i > 0$ such that for all $\varepsilon \in (0, \delta_i)$:

$$\langle \omega_{k-1} + \varepsilon\omega_k, v_i\rangle > 0. \tag{43}$$

Let $\tilde{\delta} := \min_i\{\delta_i\}$. We fix a $\varepsilon \in (0, \tilde{\delta})$ and define the matrix A' with rows

$$A' = \begin{pmatrix} \omega_1 \\ \omega_2 \\ \vdots \\ \omega_{k-1} + \varepsilon\omega_k \end{pmatrix} \in \mathbb{Q}^{k-1,n}.$$

$$\boxed{\underline{\text{Claim:}}\; A' \text{ has full row rank and } 0 <_{A'} v \text{ holds for all } v \in V.}$$

proof of claim: The fact that A' has full row rank follows directly from the initial assumption that A has full row rank. Now let $v \in V$. Then one of the following three cases hold:

Case 1: $\langle \omega_{k-1}, v\rangle < 0$. Then the assumption $0 <_A v$ implies $Av >_{lex} 0$, meaning that

$$\langle w_1, v\rangle = ... = \langle w_{j-1}, v\rangle = 0 \text{ and } \langle \omega_j, v\rangle > 0 \text{ for some } \; j \in \{1, ..., k-2\}.$$

In particular, $A'v >_{lex} 0$ also holds, as a tie is broken before $k-1$. This implies $0 <_{A'} v$.

Case 2: $\langle \omega_{k-1}, v\rangle = 0$. Then $Av >_{lex} 0$ implies either $\langle \omega_j, v\rangle > 0$ for some $j \in \{1, ..., k-2\}$ or $\langle \omega_j, v\rangle = 0$ for all $j \in \{1, .., k-1\}$ and $\langle \omega_k, v\rangle > 0$. In either case we have in particular that $A'v > 0 \iff 0 <_{A'} v$.

Case 3: $\langle \omega_{k-1}, v\rangle > 0$. Then by construction $\langle \omega_{k-1} + \varepsilon\omega_k, v\rangle > 0$. This together with $Av >_{lex} 0$ implies $A'v > 0$ as the tie is broken at the latest at the last row of A'.
Taken together, the three cases prove the claim.

Due to the claim, we may now apply the induction hypothesis to the matrix A'. There exists a $\delta' > 0$ such that for all $\varepsilon' \in (0, \delta')$ the following holds:

$$\langle \omega_1 + \varepsilon'\omega_2 + (\varepsilon')^2\omega_3 + ... + (\varepsilon')^{k-3}\omega_{k-2} + (\varepsilon')^{k-2}(\omega_{k-1} + \varepsilon'\omega_k), v\rangle > 0 \text{ for all } v \in V. \tag{44}$$

Finally, upon setting $\delta := \min(\tilde{\delta}, \delta')$ and combining (43) with (44), we obtain

$$\langle \omega_1 + \varepsilon\omega_2 + \varepsilon^2\omega_3 + ... + \varepsilon^{k-1}\omega_k, v\rangle > 0 \quad \text{for all } \ v \in V \ , \ \varepsilon \in (0, \delta)$$

which proves the lemma. □

Proposition 4.3. *Let $A_\prec = [\omega_1, ..., \omega_n] \in \mathbb{Q}^{n,n}_{\geq 0}$ be a monomial order matrix of $\prec$ with non-negative entries, where each $\omega_i \in \mathbb{Q}^n_{\geq 0}$ denotes a row of $A_\prec$. For an ideal I, let $C_\prec$ denote its Gröbner cone with respect to $\prec$. There exists a $\varepsilon > 0$ such that*

$$\omega_1 + \varepsilon\omega_2 + \varepsilon^2\omega_3 + ... + \varepsilon^{n-1}\omega_n \in \operatorname{int}(C_\prec).$$

Proof. Let $G_\prec = \{g_1, ..., g_s\}$ and $v \in \mathrm{BV}(G_\prec)$ be a bounding vector. Then v is of the form $v = \alpha_i - \beta_i$, for some $i \in \{1, ..., s\}$, where $x^{\alpha_i} = in_\prec(g_i)$ and $\beta_i \in \operatorname{supp}(g_i - in_\prec(g_i))$. In particular $v \neq 0$, and $\mathrm{BV}(G_\prec)$ has finite cardinality.
We observe that

$$\begin{aligned} x^{\beta_i} \prec x^{\alpha_i} &\iff A_\prec\alpha_i >_{lex} A_\prec\beta_i \\ &\iff A_\prec(\alpha_i - \beta_i) >_{lex} 0 \\ &\iff A_\prec v >_{lex} 0 \\ &\iff 0 <_{A_\prec} v. \end{aligned}$$

Therefore the set $\mathrm{BV}(G_\prec)$ and the matrix $A_\prec$ fulfill the conditions for Lemma 4.2. It follows that there exists a $\delta > 0$ such that for all $\varepsilon \in (0, \delta)$:

$$\langle \omega_1 + \varepsilon\omega_2 + \varepsilon^2\omega_3 + ... + \varepsilon^{n-1}\omega_n, v\rangle > 0 \quad \text{for all } \ v \in \mathrm{BV}(G_\prec). \tag{45}$$

It now follows from the second part of Proposition 1.20 that $\omega_1 + \varepsilon\omega_2 + \varepsilon^2\omega_3 + ... + \varepsilon^{n-1}\omega_n \in \operatorname{int}(C_\prec)$, as desired.

□

The vector in (45) will appear several times throughout the following chapters, so we give it a name.

Definition 4.4. Let $A_\prec = [\omega_1, ..., \omega_n]$ be a monomial order matrix, where ω_i denotes once again a row of $A_\prec$. For $\varepsilon > 0$, the **ε-perturbed weight vector** of $A_\prec$ (of degree n) is

$$\omega_\varepsilon := \omega_1 + \varepsilon\omega_2 + \varepsilon^2\omega_3 + ... + \varepsilon^{n-1}\omega_n.$$

Using this and recalling Definition 1.14 of the Gröbner cone we can reformulate Proposition 4.3 as follows:

Corollary 4.5. *In the setting of Proposition 4.3, for sufficiently small $\varepsilon > 0$, the $\varepsilon-$perturbed vector ω_ε represents $\prec$. Equivalently,*

$$in_{\omega_\varepsilon}(I) = in_\prec(I).$$

Example 4.6. Consider $I = \langle x^2 + yz, xy + z^2\rangle \lhd \mathbb{Q}[x, y, z]$ and the graded reverse lexicographic monomial order *grevlex*. As seen in section 3.1, reduced Gröbner of I with respect to this ordering is

$$G_{grevlex} := G_\prec = \{\underline{xy} + z^2, \underline{x^2} + yz, \underline{y^2z} - xz^2\}$$

which gives us the bounding vectors

$$\mathrm{BV}(G_\prec) = \{(1,1,-2),(2,-1,-1),(-1,2,-1)\} =: \{v_1,v_2,v_3\}.$$

A monomial order matrix for $\prec$ is given by

$$A_\prec = \begin{pmatrix} 1 & 1 & 1 \\ 0 & 0 & -1 \\ 0 & -1 & 0 \end{pmatrix} =: \begin{pmatrix} \omega_1 \\ \omega_2 \\ \omega_3 \end{pmatrix}$$

and indeed for $\varepsilon := \frac{1}{3}$ (and thus $\omega_\varepsilon = (1, \frac{8}{9}, \frac{6}{9})$) we observe that

$$\langle \omega_\varepsilon, v_1 \rangle = \frac{5}{9}\ ,\ \langle \omega_\varepsilon, v_2 \rangle = \frac{4}{9}\ ,\ \text{and}\ \ \langle \omega_\varepsilon, v_3 \rangle = \frac{1}{9}\ ,$$

which implies $\omega_\varepsilon \in \mathrm{int}(C_\prec)$. Consequently w_ε represents *grevlex*.

4.2 Deterministic perturbation

In Example 4.6, the perturbed vector ω_ε represents *grevlex* for I for all $\varepsilon < \frac{1}{2}$. The perturbed vector for $\varepsilon := \frac{1}{2}$ does not lie in the interior of $C_\prec$ as $\langle \omega_\varepsilon, v_3 \rangle = 0$. A general method for the computing such an ε in the general case was first introduced by Quoc-Nam Tran in [Tra00].

Theorem 4.7. *([Tra00, Theorem 3.1]) Let I be an ideal, $\prec$ be a monomial order with matrix $A_\prec \in \mathbb{Q}^{n,n}_{\geq 0}$. We denote the rows of $A_\prec$ by $\omega_1, ..., \omega_n \in \mathbb{Q}^n_{\geq 0}$ and its individual entries by a_{ij}. (In accordance with Remark 1.31 we assume that all entries of $A_\prec$ are non-negative.) Let $G_\prec = \{g_1, ..., g_s\}$ be the reduced Gröbner basis of I with respect to $\prec$. We define the quantities*

$$M_1 := \max_{1\leq i,j\leq n} \{|a_{ij}|\}\ ,\ M_2 := \max_{1\leq i\leq s}\{\max_{\beta\in supp(g_i)} \sum_{k=1}^{n} |\beta_k|\}\ \ \textit{and}\ \ \varepsilon := \frac{1}{M_1 M_2}.$$

Then:

$$\omega_1 + \varepsilon\omega_2 + \varepsilon^2\omega_3 + ... + \varepsilon^{n-1}\omega_n = \omega_\varepsilon \in \mathrm{int}(C_\prec).$$

In particular, ω_ε represents $\prec$ for I.

Proof. We start by showing the following claim:

<u>*Claim:*</u> For any $d \in \mathbb{N}$ with $d \geq M_1 M_2$, the vector

$$\tilde{w} := d^{n-1}\omega_1 + d^{n-2}\omega_2 + ... + d\omega_{n-1} + \omega_n \tag{46}$$

represents $\prec$. In particular $\tilde{\omega} \in \mathrm{int}(C_\prec)$.

Once again, we invoke Proposition 1.20: that is, we show $\langle \tilde{w}, v \rangle > 0$ for all $v \in \mathrm{BV}(G_\prec)$. Let $v \in \mathrm{BV}(G_\prec)$ be a bounding vector of the form $\alpha - \beta$, where $x^\alpha = in_\prec(g)$ and $\beta \in \mathrm{supp}(g - in_\prec(g))$ for some $g \in G_\prec$. Our goal is to show $\langle \tilde{w}, \alpha \rangle > \langle \tilde{w}, \beta \rangle$.

$in_\prec(g) = x^\alpha$ implies $x^\beta \prec x^\alpha$ and therefore $\beta <_{A_\prec} \alpha$. Thus, there exists a $k \in \{1, ..., n\}$ such that

$$\langle \omega_k, \alpha \rangle > \langle \omega_k, \beta \rangle \quad \text{and} \quad \langle \omega_i, \alpha \rangle = \langle \omega_i, \beta \rangle \quad \forall i \in \{1, ..., k-1\}.$$

If $k = n$ then $\langle \tilde{w}, \alpha \rangle > \langle \tilde{w}, \beta \rangle$ follows immediately.

For the case that $k < n$, we have that both

$$\langle \omega_k, \alpha \rangle \geq \langle \omega_k, \beta \rangle + 1 \tag{47}$$

and (by construction of d)

$$\langle \omega_i, \beta \rangle \leq M_1 \sum_{j=1}^{n} \beta_j \leq M_1 M_2 \leq d-1 \quad \text{for } \underline{\text{all}} \; i \in \{1, ..., n\}. \tag{48}$$

Equation (48) implies the following:

$$\sum_{i=k+1}^{n} d^{n-i} \langle \omega_i, \beta \rangle \leq \sum_{i=k+1}^{n} d^{n-i}(d-1) = d^{n-k} - 1$$

Combining this with (47) yields

$$\begin{aligned} d^{n-k}\langle \omega_k, \alpha \rangle &\geq d^{n-k}(\langle \omega_k, \beta \rangle + 1) \\ &= d^{n-k}\langle \omega_k, \beta \rangle + d^{n-k} \\ &> d^{n-k}\langle \omega_k, \beta \rangle + \sum_{i=k+1}^{n} d^{n-i}\langle \omega_i, \beta \rangle = \sum_{i=k}^{n} d^{n-i}\langle \omega_i, \beta \rangle. \end{aligned}$$

This in turn implies

$$\begin{aligned} \langle \tilde{w}, \alpha \rangle &= \sum_{i=1}^{n} d^{n-i}\langle \omega_i, \alpha \rangle \\ &= \sum_{i=1}^{k-1} d^{n-i}\langle \omega_i, \alpha \rangle + d^{n-k}\langle \omega_k, \alpha \rangle + \sum_{i=k+1}^{n} d^{n-i}\langle \omega_i, \alpha \rangle \\ &> \sum_{i=1}^{k-1} d^{n-i}\langle \omega_i, \beta \rangle + \sum_{i=k}^{n} d^{n-i}\langle \omega_i, \beta \rangle + \underbrace{\sum_{i=k+1}^{n} d^{n-i}\langle \omega_i, \alpha \rangle}_{\geq 0^*} \qquad \geq \langle \tilde{w}, \beta \rangle, \end{aligned}$$

where * follows from our assumption that the rows of $A_\prec$ are non-negative. This proves the claim.

The statement of Theorem 4.7 now follows from the claim by setting $d = M_1 M_2$ and observing that for $\varepsilon = \frac{1}{d}$ we have the relationship $\tilde{w} = d^{n-1}\omega_\varepsilon$. Therefore for all $v \in \mathrm{BV}(G_\prec)$

$$\langle \tilde{w}, v \rangle > 0 \iff \langle \frac{1}{d^{n-1}} \tilde{w}, v \rangle > 0 \iff \langle \omega_\varepsilon, v \rangle > 0$$

holds, implying $\omega_\varepsilon \in \mathrm{int}(C_\prec)$ by Proposition 1.20. $\square$

Remark 4.8. It may not be the case that the quantity $\frac{1}{M_1 M_2}$ from the previous theorem is a valid choice of δ in Lemma 4.2; that is, there may be quantites $\varepsilon < \frac{1}{M_1 M_2}$ for which $\omega_\varepsilon \notin \mathrm{int}(C_\prec)$. However, in practical applications we are interested in a concrete choice of ε, and the Theorem 4.7 delivers this. In fact, for the same reasons behind the integer weight vector modification described in Section 3.4, it may be advantageous to choose $\tilde{w}$ (as defined in (46)) instead ω_ε as our starting vector, as both represent $\prec$, but only the latter has integer entries. In this setting, we refer to the quantity d as the *perturbation factor* of the weight vector $\tilde{w}$.

The ε from the previous result depends on the maximal total degree of a Gröbner basis element $g \in G_\prec$. In practice, we therefore run once again into the difficulty of not knowing this quantity in advance during the perturbation the target vector. The following result introduces an a priori bound for this quantity given only some other generating set of I.

Lemma 4.9. *([Tra00, Lemma 2.1]) Let $F \subset k[x_1, ..., x_n]$ be a finite collection of polynomials. We define*

$$M := \max_{f \in F}\{\max_{\beta \in supp(f)} \sum_{k=1}^{n} |\beta_k|\}.$$

Let $I = \langle F \rangle$ and $\prec$ be a monomial order. the total degree of the polynomials in the reduced Gröbner basis $G_\prec$ of I w.r.t $\prec$ is bounded above by

$$(M^2 + 2M)^{2^{n-1}}. \tag{49}$$

Proof. consult [Dub90]. □

4.3 The perturbed Gröbner walk

An algorithm for the Gröbner walk with deterministically perturbed starting and target vectors is given below:

Algorithm 5 dPerturbedWalk($G_\prec, A_\prec, A_{\prec'}$)

Input: $G_\prec$ ▷ the marked Gröbner basis of I w.r.t $\prec$
$A_\prec$ and $A_{\prec'}$ ▷ monomial order matrices for $\prec$ and$\prec'$ with non-negative entries

Output: $G_{\prec'}$ ▷ the marked Gröbner basis of I w.r.t $\prec'$

$M_\prec \leftarrow$ max(entries($A_\prec$)) ▷ Compute the largest entries of the matrices
$M_{\prec'} \leftarrow$ max(entries($A_{\prec'}$))

$M_2 \leftarrow$ DubéBound($G_\prec$) ▷ compute the Dubé bound

$d_\prec \leftarrow M_\prec \cdot M_2$ ▷ The perturbation factors
$d_{\prec'} \leftarrow M_{\prec'} \cdot M_2$

$\tilde{\sigma} \leftarrow$ Perturb($A_\prec, d_\prec$) ▷ Compute the perturbed vectors
$\tilde{\tau} \leftarrow$ Perturb($A_{\prec'}, d_{\prec'}$)

$A_\prec \leftarrow \begin{pmatrix} \tilde{\sigma} \\ A_\prec \end{pmatrix}$ ▷ New matrices for $\prec$ and $\prec'$

$A_{\prec'} \leftarrow \begin{pmatrix} \tilde{\tau} \\ A_{\prec'} \end{pmatrix}$

return StandardGroebnerWalk($G_\prec, A_\prec, A_{\prec'}$)

---

where "DubéBound" is the quantity d from Lemma 4.9 and "Perturb" is the following subroutine:

Algorithm 6 PERTURB(A, d)

Input: $A \in \mathbb{Q}^{k,n}$ ▷ a $k \times n$ matrix with rows $\omega_1, ..., \omega_k \in \mathbb{Q}^n_{\geq 0}$
$d \in \mathbb{N}$ ▷ a non-negative integer

Output: $\tilde{w}$ ▷ a perturbed weight vector with perturbation factor d

OUTPUT $\leftarrow$ ZEROS(n) ▷ An $n-$dimensional vector with all zero entries
for i in $\{1, ..., k\}$ **do**
  OUTPUT $\leftarrow$ OUTPUT $+ d^{i-1}\omega_i$
end for
return OUTPUT

The correctness of dPERTURBEDWALK is an almost direct consequence of the correctness of STANDARDGROEBNERWALK (cf. Theorem 2.12). Strictly speaking, the modification to the target monomial matrix in the penultimate line implies that the output of dPERTURBEDWALK is the marked Gröbner basis with respect to the ordering $\prec'$, refined by $\tilde{\tau}$. However, Theorem 4.7 implies $\tilde{\tau} \in \text{int}(C_{\prec'})$. Therefore (by Corollary 1.17), it follows that $G_{\prec'_{\tilde{\tau}}} = G_{\prec'}$ as desired.

The fact that all intersections of the perturbed path with boundaries of cones generically occur on facets implies minimality of the length of initial forms by Proposition 4.1,which was our desired outcome. However, the following example illustrates a major difficulty which the deterministically perturbed walk encounters in practice.

Example 4.10. Let $I = \langle x^2 + yz, xy + z^2 \rangle$ be the ideal from the examples in Section 3. The deterministic degree bound obtained by applying Lemma 4.9 to I is

$$(2^2 + 2 * 2)^{2^{3-1}} = 8^4 = 4096.$$

However, the actual highest degree of an element of any Gröbner basis of I is 4. Using the deterministic degree bound to compute d for the algorithm above leads to intermediate weight vectors of order 10^8, which unnecessarily slows down computations.

`Macaulay2` does not allow for monomial orders to be specified by weight vectors with entries of order greater than 10^{10}; thus, the perturbed walk using the deterministic Dubé bound is not only ill-advised, but indeed impossible to implement. A possible way around these unnecessarily long weight vectors is *heuristic perturbation*: here, we perturb the target vector by some prescribed factor d and execute the walk with this $\tilde{\tau}$ as the target vector. Before terminating, we assert whether the outputted Gröbner basis $G_{\prec'_{\tilde{\tau}}}$ has the properties we desire. This approach may be especially applicable in problems of *elimination* (cf. Section 6), in which a wider class of Gröbner bases would solve the problem at hand.

Algorithm 7 HPERTURBEDWALK($G_{\prec}, A_{\prec}, A_{\prec'}, d$)

Input: $G_{\prec}$ ▷ the marked Gröbner basis of I w.r.t $\prec$
$A_{\prec}$ and $A_{\prec'}$ ▷ monomial order matrices for $\prec$ and $\prec'$ with non-negative entries
d ▷ A positive integer

Output: $G_{\prec'}$ ▷ the marked Gröbner basis of I w.r.t $\prec'$

$\tilde{\tau} \leftarrow$ PERTURB($A_{\prec'}, d$) ▷ perturb τ by the prescribed factor

$A_{new} \leftarrow \begin{pmatrix} \tilde{\tau} \\ A_{\prec'} \end{pmatrix}$
$G_{new} \leftarrow$ STANDARDGROEBNERWALK($G_{\prec}, A_{\prec}, A_{new}$)
if ISGROEBNERBASIS($G_{new}, A_{\prec'}$) **then** ▷ checks if G_{new} is the desired basis
return G_{new}
else ▷ If not, walk from G_{new} to $G_{\prec'}$
return STANDARDGROEBNERWALK($G_{new}, A_{new}, A_{\prec'}$)
end if

There are two reasons we do not perturb the starting vector σ in the heuristically perturbed walk. The first is that we cannot guarantee that the perturbed vector lies in $C_{\prec}$, which may lead to issues at initialization. The second reason is empirically, computational bottlenecks involving coefficient swell occur at the final conversions, as this is where both the number of terms of the initial forms and their coefficients are at their largest. If we can ensure that we enter the final cone at a facet, the impact of the large coefficients is minimized due to the truncation of initial forms.

Example 4.11. Calling HPERTURBEDWALK with a perturbation factor of $d = 64$ on the ideal from the running example (40) yielded a Gröbner basis w.r.t *lex* after 53 conversions, whereby no additional conversions were necessary after computing G_{new}. This took a total of 2.9 seconds, which is an improvement on STANDARDGROEBNERWALK by a factor of 3.

Further discussion/comparison of the perturbed Gröbner walk is present in Section 6. For now, we turn our attention to another variant of the Gröbner walk: the *generic Gröbner walk*.

5 Symbolic perturbation: the generic Gröbner walk

The generic Gröbner walk was first presented in [Fuk+07] and takes the idea of path perturbation one step further. Here, the perturbed line segment $\overline{\sigma\tau}$ is replaced by a *symbolic* line segment in such a way that intersections at facets are guaranteed at every iteration. Crucially, this happens in such a way that intermediate computations are independent of the weight vectors encountered. It can be viewed as an attempt to preserve the advantages of the deterministically perturbed walk, without the computational costs of weight vectors with large entries. The material of this section is based on [Fuk+07]: our contributions consist of the formal proof of correctness of the generic walk and its subroutines, and the in-depth discussion of the 3-dimensional example in Section 5.4.

Let I be an ideal. To simplify notation in the upcoming sections, we rename the matrices $A_{\prec}$ and $A_{\prec'}$ to S and T respectively. In accordance with Remark 1.31, we assume that both matrices are invertible $n \times n$ matrices with non-negative entries. We denote their rows by σ_i and τ_i respectively. Summing up, we write

$$S := A_{\prec} = \begin{pmatrix} \sigma_1 \\ \sigma_2 \\ \vdots \\ \sigma_n \end{pmatrix} \quad \text{and} \quad T := A_{\prec'} = \begin{pmatrix} \tau_1 \\ \tau_2 \\ \vdots \\ \tau_n \end{pmatrix}, \text{ where } \; S, T \in GL_n(\mathbb{Q}) \text{ and } \sigma_i, \tau_i \in \mathbb{Q}^n_{\geq 0} \text{ for all } i.$$

By Theorem 4.7, there exist small enough positive quantities δ and ε such that

$$\sigma_\delta \in \operatorname{int}(C_{\prec}) \quad \text{and} \quad \tau_\varepsilon \in \operatorname{int}(C_{\prec'}),$$

where σ_δ and τ_ε are the $\delta-$ and $\varepsilon-$ perturbed vectors of S and T respectively (cf. definition 4.4). We now consider the Gröbner walk from $C_{\prec}$ to $C_{\prec'}$ along the *δ-ε-perturbed path*, which we define as the line segment

$$\ell(u) := (1-u)\cdot\sigma_\delta + u\cdot\tau_\varepsilon \text{ , where } u \in [0,1].$$

Like StandardGroebnerWalk and dPerturbedWalk, the generic Gröbner walk works by following this line segment, computing a reduced Gröbner basis every time it enters a new full-dimensional cone by lifting a basis of initial forms with "Lift" (cf. Algorithm 3). The main point of difference lies in the method by which the initial forms are computed at every step: instead of computing the actual weight vector ω at which the line enters a new cone, $in_\omega(I)$ is computed by identifying the inner facing facet normal of the face on which ω lies. Crucially, all computations end up being independent of the quantities ε and δ. In order to identify these facet normals, we describe a construction introduced by [Fuk+07] called the *facet preorder*.

5.1 The facet preorder

Definition 5.1. Let $I \lhd \mathbb{Q}[x_1,...,x_n]$ be an ideal. We define the **Bounding vectors** of I as

$$\mathrm{BV}(I) := \bigcup_{\prec} \mathrm{BV}(G_{\prec}) \text{ ,}$$

where $\prec$ varies across all monomial orders.

Remark 5.2. By Corollary A.19 and Definition 1.18, $\mathrm{BV}(I)$, consists of finitely many vectors in $\mathbb{Z}^n$.

Let $<_S$ and $<_T$ be the strict total orderings on $\mathbb{R}^n$ induced by the matrices S and T (cf Definition B.3).

Definition 5.3. For two monomial orders $\prec$ and $\prec'$ represented by matrices S and T, the *path region* with respect to $\prec$ and $\prec'$ (or, equivalently S and T) is the set

$$R_{\prec,\prec'} := R_{S,T} := \{x \in \mathbb{R}^n : x >_S 0 \text{ and } x <_T 0\}. \tag{50}$$

Example 5.4. In the example in Section 3.3 $\prec$ is the lexicographic order on $\mathbb{Q}[x, y, z]$ and $\prec'$ is the graded lexicographic order, refined by $(1, 3, 0)$. The path region with respect to these orderings is

$$\begin{aligned} R_{S,T} =& \Big\{ x \in \mathbb{R}^3 : x >_S 0 \quad \text{and} \quad x <_T 0 \Big\} \\ =& \left\{ x \in \mathbb{R}^3 : \begin{pmatrix} x_1 \\ x_2 \\ x_3 \end{pmatrix} >_{lex} \begin{pmatrix} 0 \\ 0 \\ 0 \end{pmatrix} \quad \text{and} \quad \begin{pmatrix} x_1 + 3x_2 \\ x_1 + x_2 + x_3 \\ x_1 \end{pmatrix} <_{lex} \begin{pmatrix} 0 \\ 0 \\ 0 \end{pmatrix} \right\} \\ =& \Big\{ x \in \mathbb{R}^3 : x_1 > 0 \vee (x_1 = 0 \wedge x_2 > 0) \vee (x_1 = x_2 = 0 \wedge x_3 > 0) \Big\} \\ & \cap \Big\{ x \in \mathbb{R}^3 : x_1 + 3x_3 < 0 \vee (x_1 + 3x_2 = 0 \wedge x_1 + x_2 + x_3 < 0) \vee (x_1 + 3x_3 = x_1 + x_2 + x_3 = 0 \wedge x_3 < 0) \Big\}. \end{aligned}$$

This is an unbounded, impure, half-open polyhedral region in $\mathbb{R}^3$.

Remark 5.5. The relevance of the previous two definitions to our setting is that all of the facets intersected by the generic line segment ℓ have inner normal vectors lying in $R_{S,T} \cap \mathrm{BV}(I)$.

To identify the normal vectors of the facets which the perturbed path ℓ intersects, we introduce a relation on the elements of $\mathrm{BV}(I) \cap R_{S,T}$. To do this, we must define yet another type of ordering. Given a strict total ordering $<$ on $\mathbb{Q}^n$, we may construct an ordering $\tilde{<}$ on $\mathbb{Q}^{n\times n}$ via lexicographic extension: that is, given two matrices $A, B \in \mathbb{Q}^{n\times n}$, we compare rows with $<$ until a tie is broken.

Definition 5.6. Let $v_i, v_j \in \mathrm{BV}(I) \cap R_{S,T}$. The **facet preorder** $<_F$ is the relation given by

$$v_i <_F v_j \quad \text{if and only if} \quad T v_i v_j^t \mathrel{\tilde{<}_S} T v_j v_i^t, \tag{51}$$

where $\tilde{<}_S$ denotes the lexicographic extension of $<_S$ onto $\mathbb{Q}^{n\times n}$.

It is clear that $<_F$ is a strict preorder; anti-reflexivity and transitivity follow directly from the anti-reflexivity/transitivity properties of $<_S$ (cf. Lemma B.4). The facet preorder is instrumental in determining the facets intersected by ℓ. The following theorem and its proof are a reformulation of the observations from [Fuk+07, pg.10].

Theorem 5.7. *Let $\prec$, $\prec'$ be monomial orders and assume that $C_\prec \neq C_{\prec'}$. For small enough $\delta, \varepsilon > 0$, the perturbed line segment $\ell = \overline{\sigma_\delta \tau_\varepsilon}$ exits $C_\prec$ at a facet which has an inner-facing normal vector which is minimal in the set $\mathrm{BV}(G_\prec) \cap R_{S,T}$ with respect to $<_F$.*

Proof. We start by associating each bounding vector $v_i \in \mathrm{BV}(G_\prec)$ with the quantity

$$u_i := \frac{\langle \sigma_\delta, v_i \rangle}{\langle \sigma_\delta, v_i \rangle - \langle \tau_\varepsilon, v_i \rangle}.$$

Let δ and ε be small enough such that by Proposition 4.3, $\sigma_\delta \in \mathrm{int}(C_\prec)$ and $\tau_\varepsilon \in \mathrm{int}(C_{\prec'})$. In particular, $C_\prec \neq C_{\prec'}$ implies $\tau_\varepsilon \notin C_\prec$. With this choice of ε and δ, the following two statements are consequences of Lemma 4.2 and Proposition 1.20:

$$\langle \sigma_\delta, v \rangle > 0 \iff 0 <_S v \quad \text{for all } v \in \mathrm{BV}(G_\prec)\ , \tag{52}$$

and

$$\text{There exists a} \quad v \in \mathrm{BV}(G_\prec) \text{ such that } \langle \tau_\varepsilon, v \rangle < 0 \iff v <_T 0. \tag{53}$$

Thus, $\mathrm{BV}(G_\prec) \cap R_{S,T} \neq \emptyset$ and (analogously to the proof of Proposition 2.2), the point at which the line segment $\overline{\sigma_\delta \tau_\varepsilon}$ exits $C_\prec$ is precisely $\ell(\hat{u})$, where

$$\hat{u} = \min\Big\{ u_i \ \Big|\ v_i \in \mathrm{BV}(G_\prec) \cap R_{S,T} \Big\}. \tag{54}$$

The theorem is a consequence of the following claim:

$$\boxed{\underline{\textit{Claim:}} \qquad u_i < u_j \iff v_i <_F v_j \qquad \text{for all } v_i \in \mathrm{BV}(I)\ ,\ v_j \in \mathrm{BV}(I) \cap R_{S,T}.}$$

Proof of claim: We start by defining the set

$$N_1 := \{\langle \tau_k, v_i \rangle v_j : k \in \{1, ..., n\}, v_i, v_j \in \mathrm{BV}(I)\},$$

where we remind the reader that τ_k is the $k-$th row of the monomial order matrix T of $\prec'$. Lemma 4.2 implies that there exists a small enough $\delta > 0$ such that for all elements $x, y \in N_1$:

$$x <_S y \iff \langle \sigma_\delta, x \rangle < \langle \sigma_\delta, y \rangle. \tag{55}$$

This follows by applying the lemma to the set of vectors $\{x - y \mid x, y \in N_1, x >_S y\}$.
For a $\delta > 0$ fulfilling (55) we subsequently define

$$N_\delta := \{\langle \sigma_\delta, v_i \rangle v_j : v_i, v_j \in \mathrm{BV}(I)\},$$

and observe that (again by Lemma 4.2) there exists an $\varepsilon > 0$ such that for all $w, z \in N_\delta$:

$$w <_T z \iff \langle \tau_\varepsilon, w \rangle < \langle \tau_\varepsilon, z \rangle. \tag{56}$$

We pick δ and ε small enough such that (52), (53), and (55), (56) are all fulfilled. Then for all $v_i \in \mathrm{BV}(I)$ and $v_j \in \mathrm{BV}(I) \cap R_{S,T}$ the following holds:

$$\begin{aligned}
u_i < u_j &\iff \frac{\langle \sigma_\delta, v_i \rangle}{\langle \sigma_\delta, v_i \rangle - \langle \tau_\varepsilon, v_i \rangle} < \frac{\langle \sigma_\delta, v_i \rangle}{\langle \sigma_\delta, v_j \rangle - \langle \tau_\varepsilon, v_j \rangle} \\
&\iff \frac{\langle \tau_\varepsilon, v_i \rangle}{\langle \sigma_\delta, v_i \rangle} < \frac{\langle \tau_\varepsilon, v_j \rangle}{\langle \sigma_\delta, v_j \rangle} \\
&\iff \langle \tau_\varepsilon, \langle \sigma_\delta, v_j \rangle v_i \rangle < \langle \tau_\varepsilon, \langle \sigma_\delta, v_i \rangle v_j \rangle \\
&\overset{(56)}{\iff} \langle \sigma_\delta, v_j \rangle v_i <_T \langle \sigma_\delta, v_i \rangle v_j.
\end{aligned}$$

Where the last equivalence holds because $\langle \sigma_\delta, v_j \rangle v_i, \langle \sigma_\delta, v_i \rangle v_j \in N_\delta$. Now note that for any row τ_k of the matrix T we have that due to the fact that δ fulfills (55):

$$\begin{aligned}
\langle \tau_k, \langle \sigma_\delta, v_j \rangle v_i \rangle < \langle \tau_k, \langle \sigma_\delta, v_i \rangle v_j \rangle &\iff \langle \sigma_\delta, \langle \tau_k, v_i \rangle v_j \rangle < \langle \sigma_\delta, \langle \tau_k, v_i \rangle \rangle \\
&\overset{(55)}{\iff} \langle \tau_k, v_i \rangle v_j <_S \langle \tau_k, v_j \rangle v_i.
\end{aligned}$$

The vectors in the last inequality are precisely the kth rows of the matrices $Tv_iv_j^t$ and $Tv_jv_i^t$ respectively. Thus, the comparisons in the last line above are tantamount to comparing rows of $Tv_iv_j^t$ and $Tv_jv_i^t$ with the lexicographic order on $\mathbb{Q}^n$. Combining this with the definition of of $<_F$ (cf. Definition 5.6), we obtain

$$u_i < u_j \iff Tv_iv_j^t \,\tilde{<}_S\, Tv_jv_i^t \iff v_i <_F v_j. \tag{57}$$

This completes the proof of the claim.

Let v_i be minimal in $\mathrm{BV}(G_\prec)$ w.r.t $<_F$. It follows from (54) and Proposition 2.2 that $\ell(u_i) = \ell(\hat{u})$ is the point of $\partial C_\prec$ at which the line segment leaves $C_\prec$. It remains to show that the corresponding vector v_i is facet-defining. To this end, if we assume that $v_i = v_j$ for two bounding vectors $v_i, v_j \in \mathrm{BV}(G_\prec) \cap R_{S,T}$, it follows from the converse of (57) that $Tv_iv_j^t = Tv_jv_i^t$. As T is invertible, this implies $v_iv_j^t = v_jv_i^t$, which in turn means that $v_j \in \mathrm{posHull}(v_i)$. Thus, any two such vectors define the same face of $C_\prec$, which for dimension reasons must be a facet. $\square$

An important observation is that the facet preorder is dependent only on S and T. Crucially, it is independent of the values of δ and ε; this leads to the main theoretical advantage of the generic walk. For two distinct cones $C_\prec$ and $C_{\prec'}$, an algorithm for computing the inner facet normal of the facet on which $\omega := \ell(\hat{u})$ lies can be described as follows:

Algorithm 8 GETNEXTV($G_\prec$, $\tilde{v}$, S, T)

Input: $G_\prec$ ▷ The marked Gröbner basis w.r.t $\prec$
$\tilde{v}$ ▷ The bounding vector from the previous iteration ("$-\infty$" at initialization)
S ▷ A monomial order matrix for $\prec$ with non-neg entries
T ▷ A monomial order matrix for $\prec'$ with non-neg entries

Output: v ▷ The element of BV($G_\prec$) corresponding to the facet intersecting with ℓ

```
BV ← BV(G_≺)
V ← { }
for v in BV do
    if ṽ <_F v and v ∈ R_{S,T} then
        V ← V ∪{v}                  ▷ Collect all v ∈ BV(G_≺) ∩ R_{S,T} with ṽ <_F v
    end if
end for
if V == {} then                     ▷ if no such vectors exist, signal termination
    return +∞
else
    return min_{<_F}(V)             ▷ Return the minimal element of V w.r.t <_F
end if
```

The correctness of Algorithm 8 is a direct consequence of Theorem 5.7. The following lemma is important for the correctness/termination of the generic walk.

Lemma 5.8. *In the setting of Algorithm 8,*

$$\text{GETNEXTV}(G_\prec,\ \tilde{v},\ S,\ T) = +\infty \iff G_\prec = G_{\prec'}\ .$$

Proof. "$\Longleftarrow$" Assume $G_\prec = G_{\prec'}$. Then (with $\varepsilon > 0$ small enough such that Theorem 5.7 holds) $\tau_\varepsilon \in \operatorname{int}(C_{\prec'})$. It follows that $\langle \tau_\varepsilon, v\rangle > 0 \iff 0 <_T v$ for all $v \in \mathrm{BV}(G_\prec)$. In particular, $v \notin R_{S,T}$ by definition of $R_{S,T}$. Consequently, the set V in Algorithm 8 is empty, and so
GETNEXTV($G_\prec$, $\tilde{v}$, S, T) outputs $+\infty$.
"$\Longrightarrow$" If GETNEXTV($G_\prec$, $\tilde{v}$, S, T)$\neq +\infty$, then by Algorithm 8 there exists a $v \in \mathrm{BV}(G_\prec) \cap R_{S,T}$ such that $v >_F \tilde{v}$. In particular, $v \in R_{S,T}$ implies that $v <_T 0 \iff \langle \tau_\varepsilon, v\rangle < 0$, which implies $\tau_\varepsilon \notin C_\prec$ by Proposition 1.20. This in turn means that $G_\prec \neq G_{\prec'}$ due to Corollary 1.17. □

5.2 The generic lifting step

Let $\omega := \ell(u_i)$ be the point along the perturbed line segment at which it exits $C_\prec$ (We assume $C_\prec \neq C_{\prec'}$). In the setting of the generic walk, we know an inner facet normal $v_i \in \mathrm{BV}(G_\prec)$ for the facet on which ω lies, but not ω itself. Because of this, we first require a method of obtaining a generating set of the initial forms $in_\omega(I)$ when only v_i is known.

Lemma 5.9. *Let $G_\prec = \{g_1, ..., g_s\}$ be the reduced Gröbner basis of I with respect to $\prec$, $v \in \mathrm{BV}(G_\prec)$ be an inner facing facet-defining normal vector to $C_\prec$, i.e*

$$\langle \omega, v\rangle \geq 0 \quad \textit{for all} \ \ \omega \in C_\prec \ , \textit{and} \ \ F_v := C_\prec \cap H_v = \{\omega \in C_\prec \ : \langle \omega, v\rangle = 0\} \ \textit{is a facet.}$$

For any weight vector $\omega \in \mathbb{Q}^n_{\geq 0} \cap \mathrm{relint}(F_v)$ and $i \in \{1, ..., s\}$, the initial form $in_\omega(g_i)$ may be written as

$$in_\omega(g_i) = x^\alpha + \sum_{\beta \in S'_{g_i}} c_\beta x^\beta,$$

where $x^\alpha = in_\prec(g_i)$, c_β is the coefficient of x_β in g, and

$$S'_{g_i} = \{\beta \in \mathrm{supp}(g_i) : \alpha - \beta \in \mathrm{posHull}(v)\}.$$

Proof. We fix an i and a $\omega \in \mathrm{relint}(F_v)$ show the set equality $\mathrm{supp}(in_\omega(g_i)) = S'_{g_i} \cup \{\alpha\}$.

"$\subseteq$": $\alpha \in \mathrm{supp}(in_\omega(g_i))$ holds as a direct consequence of $\omega \in C_\prec$ and Corollary 1.24. For any $\beta \in S'_{g_i}$, if $\alpha - \beta$ and u are positive multiples of each other, then clearly

$$\langle \omega, \alpha - \beta\rangle = 0 \iff \langle \omega, u\rangle = 0.$$

Thus, β is a term for which $\langle \omega, \beta\rangle$ is maximal, and $\beta \in in_\omega(g_i)$ follows.

"$\supseteq$": If $\beta \in \mathrm{supp}(in_\omega(g_i))$ then $\langle \alpha - \beta, \omega\rangle = 0$, implying $\alpha - \beta \in \omega^\perp$. Thus, $\alpha - \beta$ and the normal vector v are positive multiples of each other, and $\beta \in S'_{g_i}$ follows. (We know that $\alpha - \beta$ and v are *positive* multiples of each other as assuming the contrary would yield $\langle \alpha - \beta, v\rangle < 0$, contradicting $in_\prec(g_i) = x^\alpha$ by Proposition 1.20.)

□

A modified routine for obtaining $in_\omega(G_\prec) = \{in_\omega(g_1), ..., in_\omega(g_s)\}$ my now be described as follows:

Algorithm 9 GenericInitialForms($G_\prec$, v)

Input: $G_\prec$ ▷ The marked starting Gröbner basis
v ▷ a facet-defining bounding vector such that $\omega \in \mathrm{relint}(F_v)$

Output: $in_\omega(G_\prec) = \{in_\omega(g_1), .., in_\omega(g_s)\}$ ▷ the set of initial forms w.r.t ω

```
G ← List(G_≺)
OUTPUT ← {}
for g in G do
    α ← ExponentVector(in_≺(g))
    p ← x^α                          ▷ add in_≺(g) to in_ω(g) and keep the marking
    for c_β x^β in Terms(g) do
        if α − β ∈ posHull(v) then   ▷ Check if ⟨α, v⟩ = ⟨β, v⟩
            p ← p + c_β x^β          ▷ If this is the case, add the term to in_ω(g)
        end if
    end for
    OUTPUT ← OUTPUT ∪{p}
end for
return OUTPUT
```

The correctness follows from Lemma 5.9. Due to Lemma 2.7, upon keeping the markings of the input Gröbner basis $G_\prec$ we obtain a marked Gröbner basis $in_\omega(G_\prec)$ of $in_\omega(I)$.

In every non-trivial iteration of StandardGroebnerWalk, the lifting step (cf. Algorithm 3) works by computing a Gröbner basis of $in_\omega(I)$ w.r.t $\prec'_\omega$ using Buchberger's algorithm. As the exact weight vector ω remains unknown throughout the generic Gröbner walk, we describe a subroutine for obtaining the lifted basis which is independent of ω and then prove its correctness. Interreduction is not optional in the generic Gröbner walk, so we include it as part of this subroutine.

Algorithm 10 GenericLift($G_\prec$, u, S, T)

Input: $G_\prec$ ▷ the marked Gröbner basis of I w.r.t $\prec$
u ▷ the bounding vector computed in the previous iteration
S
T ▷ Monomial order matrices for $\prec$ and $\prec'$

Output: G ▷ a Gröbner basis of I w.r.t $(\prec')_\omega$

```
inwG ← GenericInitialForms(G_≺, u)
M ← markedGB(inwG, MonomialOrder = T)     ▷ convert in_ω(G) to a marked GB w.r.t ≺'
for m in M do
    r ← markedGBNormalForm(G_≺, p)        ▷ compute m̄^{G_≺} of r
    m ← m - r                              ▷ subtract normal forms from each m ∈ M
end for

G ← markedGBReduce(M)                      ▷ reduce and keep markings of M
return G
```

Despite the similarities of this procedure with the standard Lift from section 2 (cf. Algorithm 3), there are a couple of important subtleties which must be noted. Firstly, the subroutines "markedGBNormalForm" and "markedGBReduce" are necessary: these are similar to their conventional counterparts, but differ in that they only require a *marked* Gröbner basis G as input. They do *not* require an explicit description of the monomial order with respect to which G is a Gröbner basis. This is crucial in our setting because the intermediate weight vectors (and therefore the intermediate refinement monomial orders) encountered along the generic path are not known. Pseudocode for these subroutines can be found in Section A.4.

Proposition 5.10. *Algorithm 10 "GenericLift" is correct in the sense that its output G is a marked Gröbner basis of I with respect to the refinement monomial order $(\prec')_\omega$.*

Proof. As $in_\omega(G_\prec)$ is ω-homogeneous, its reduced Gröbner basis M with respect to $\prec'$ is equal to that with respect to the refinement order $(\prec')_\omega$. It follows that output of Algorithm 10 is set-theoretically equal to that of Algorithm 3. In both algorithms, the markings of the lifted basis are those of their respective "M"s. As $in_{\prec'_\omega}(m_i) = in_{\prec'}(m_i)$ for all i, the markings of the two bases are also equal. Thus (by correctness of Algorithm 3), the output of Algorithm 10 is indeed a marked Gröbner basis of I with respect to $(\prec')_\omega$, obtained by applying markedGBReduce to the inclusion minimal Gröbner basis. □

5.3 The generic Gröbner walk

Using the subroutines defined in the previous two sections, we can now describe the generic Gröbner walk as follows:

Algorithm 11 GENERICWALK($G_\prec$, S, T)

Input: $G_\prec$ ▷ the marked Gröbner basis of I w.r.t $\prec$
S
T ▷ Matrices for the starting and target orders with entries in $\mathbb{Q}^n_{\geq 0}$

Output: $G_{\prec'}$ ▷ the marked Gröbner basis of I w.r.t $\prec'$

```
G ← G_≺
v ← GETNEXTV(G, −∞, S, T)                ▷ Initialize
while v ≠ +∞ do
    G ← GENERICLIFT(G, v, S, T)
    v ← GETNEXTV(G, v, S, T)
end while
return G
```

In the expressions "$v = \pm\infty$", $+\infty$ and $-\infty$ are placeholders for integer vectors which are strictly greater/smaller than all bounding vectors $\mathrm{BV}(I) \cap R_{S,T}$ w.r.t the facet preorder $<_F$ and are solely relevant for the initialization/termination of the algorithm. We also note that the matrices S and T do not get updated as we pass from one Gröbner basis to the next as the facet preorder is dependent solely on the initial inputs $\prec$ and $\prec'$.

Proposition 5.11. *Algorithm 11* GENERICWALK *is correct and terminates after finitely many steps.*

Proof. The correctness of the lifting step follows from the correctness of the standard lifting step (Algorithm 3) and Proposition 5.10. It remains to show that the algorithm is non-stationary and terminates after finitely many steps.
Every call of "GETNEXTV" returns a vector which is strictly greater than the input vector w.r.t $<_F$. If this vector is $+\infty$, then this is the signal that we are done, and the output of the generic walk is $G_{\prec'}$ by Lemma 5.8. If the output of GETNEXTV(G, v, S, T) is some $\tilde{v} \neq +\infty$, then this indicates that we are not yet done. However, due to the claim (5.1) from Theorem 5.7, we have that $u < \tilde{u}$. In particular the point $\ell(\tilde{u})$ along the perturbed line segment ℓ is closer to τ_ε than $\ell(u)$, implying that the algorithm is non-stationary. Termination after finitely many steps follows similarly to the proof of Theorem 2.12; the line segment ℓ intersects only finitely many Gröbner cones.

□

5.4 An example

To see GENERICWALK in action, we perform the same computation from Section 3.3, this time using Algorithm 11. Let

$$I = \langle x^2 + yz, xy + z^2 \rangle$$

and

$$G_\prec = \{\underline{x^2} + yz, \underline{xy} + z^2, \underline{xz^2} - y^2 z, \underline{y^3 z} + z^4\}.$$

Our task is to convert $G_\prec$ to the reduced Gröbner basis $G_{\prec'}$ basis w.r.t the monomial order $\prec' = glex_{(1,3,0)}$, using Algorithm 11.

Recall that the monomial order matrix for $\prec = lex$ can be taken to be $S := I_3$, and $\prec'$ has monomial order matrix

$$T := \begin{pmatrix} 1 & 3 & 0 \\ 1 & 1 & 1 \\ 1 & 0 & 0 \end{pmatrix}.$$

We set $G := G_\prec$ Our first task is to compute GETNEXTV($G_\prec$, $-\infty$, S, T). To do this, first compute the path region. This is

$$R_{S,T} = \left\{ x \in \mathbb{R}^3 : x >_S 0 \quad \text{and} \quad x <_T 0 \right\}$$
$$= \left\{ x \in \mathbb{R}^3 : \begin{pmatrix} x_1 \\ x_2 \\ x_3 \end{pmatrix} >_{lex} \begin{pmatrix} 0 \\ 0 \\ 0 \end{pmatrix} \quad \text{and} \quad \begin{pmatrix} x_1 + 3x_2 \\ x_1 + x_2 + x_3 \\ x_1 \end{pmatrix} <_{lex} \begin{pmatrix} 0 \\ 0 \\ 0 \end{pmatrix} \right\}.$$

The bounding vectors of G are

$$\mathrm{BV}(G) = \left\{ \begin{pmatrix} 0 \\ 3 \\ -3 \end{pmatrix}, \begin{pmatrix} 1 \\ -2 \\ 1 \end{pmatrix}, \begin{pmatrix} 1 \\ 1 \\ -2 \end{pmatrix}, \begin{pmatrix} 2 \\ -1 \\ -1 \end{pmatrix} \right\} =: \{v_1, v_2, v_3, v_4\}.$$

Out of these, only the v_2 and v_4 lie in $R_{S,T}$, therefore GetNextV(G, $-\infty$, S, T) returns the vector out of these which is minimal w.r.t the facet preorder. To determine this we first compute

$$Tv_2v_4^t = \begin{pmatrix} -10 & 5 & 5 \\ 0 & 0 & 0 \\ 2 & -1 & -1 \end{pmatrix} \quad \text{and} \quad Tv_4v_2^t = \begin{pmatrix} -2 & 2 & -1 \\ 0 & 0 & 0 \\ 2 & -4 & 2 \end{pmatrix}.$$

Comparing these matrices w.r.t $\tilde{<}_S$ equates to comparing their rows w.r.t $<_{lex}$.

As $(-10, 5, , 5) <_{lex} (-2, 2, 1)$, we have that $Tv_2v_4^t \tilde{<}_S Tv_4v_2^t$, and therefore $v_2 <_F v_4$. So GetNextV(G, $-\infty$, S, T) $= v_2 =: v$

Calling "GenericInitialForms(G, v)" returns the Gröbner basis of initial forms $in_\omega(G)$ by calculating, for each $g \in G$, which terms of g have exponent vectors $\beta \in \mathrm{supp}(g)$ fulfilling $\alpha - \beta \in \mathrm{posHull}(v)$. Doing this yields the set

$$in_\omega(G) = \{x^2, xyxz^2 - y^2z, y^3z\}.$$

In the lifting step, $in_\omega(G)$ is converted to the reduced Gröbner basis of $in_\omega(I)$ w.r.t $\prec'$. This is

$$M := \{\underline{x^2}, \underline{xy}, \underline{y^2z} - xz^2\}.$$

M is subsequently lifted to a Gröbner basis of I by subtracting the normal form $\overline{m}^G$ from each $m \in M$:

$$G_{new} = \{\underline{x^2} + yz, \underline{xy} + z^2, \underline{y^2z} - xz^2\}.$$

We keep the markings of M, and note that G_{new} is already reduced, as no term of any of the polynomials is divisible by any of the marked terms.

As $v \neq +\infty$, we update by setting $G := G_{new}$ and reiterate. We have

$$\mathrm{BV}(G) = \left\{ \begin{pmatrix} 2 \\ -1 \\ -1 \end{pmatrix}, \begin{pmatrix} 1 \\ 1 \\ -2 \end{pmatrix}, \begin{pmatrix} -1 \\ 2 \\ -1 \end{pmatrix} \right\} = \{\tilde{v_1}, \tilde{v_2}, \tilde{v_3}\}.$$

Of these, only $\tilde{v_1}$ lies in the region $R_{S,T}$. GetNextV(G, v, S, T) checks if $\tilde{v_1}$ is greater than our previous bounding vector $v = (1 - 2, 1)$ w.r.t $<_F$: indeed (similarly to before) $T\tilde{v_1}v^t >_{lex} Tv\tilde{v_1}^t$, therefore $\tilde{v_1} >_F v$, so GetNextV(G, v, S, T) $= \tilde{v_1} =: v \neq \infty$, so we perform another conversion.

The new set of initial forms is

$$in_\omega(G) := \{x^2 + yz, xy, y^2z\},$$

which is converted to

$$M := \{\underline{yz} + x^2, \underline{xy}, \underline{x^3}\}$$

and lifted to

$$G_{new} := \{\underline{yz} + x^2, \underline{xy} + z^2, \underline{x^3} - z^3\},$$

which is also reduced. (Again, the markings of G_{new} are those of M.)
Now we have

$$\mathrm{BV}(G_{new}) = \left\{ \begin{pmatrix} -2 \\ 1 \\ 1 \end{pmatrix}, \begin{pmatrix} 1 \\ 1 \\ -2 \end{pmatrix}, \begin{pmatrix} 3 \\ 0 \\ -3 \end{pmatrix} \right\},$$

and observe that $\mathrm{BV}(G_{new}) \cap R_{S,T} = \emptyset$ which (by Lemma 5.8) implies that we are already in the desired cone. Indeed, $\textsc{GetNextV}(G_{new}, v, S, T) = +\infty$ and the algorithm terminates, outputting $G_{new} = G_{\prec'}$.

Upon comparing these computations with those from the same conversion using StandardGroebnerWalk (see Section 3.3), we see that the intermediate Gröbner bases computations are exactly same in both algorithms. The advantages of GenericWalk over StandardGroebnerWalk are twofold; firstly, no weight vectors are explicitly computed. Secondly, it is guaranteed that the perturbed line segment intersects the boundaries of cones in *facets*, thus minimizing the length of the polynomials on which Buchberger's algorithm is called due to Proposition 4.3.

We conclude this section by revisiting the running example (40) from Section 4, adding runtimes for the generic Gröbner walk:

Algorithm	number of conversions	maximal $\lvert\mathrm{supp}(in_\omega(g))\rvert$	maximal coeff. length	time (s., approx)
StandardGroebnerWalk	17	44	130	11.01
dPerturbedWalk	56	2	1147	6.08
hPerturbedWalk	53	2	1147	2.92
GenericWalk	50	3	1147	2.31

Table 2: A comparison of the Gröbner walk algorithms on the conversion from (40). The perturbed walks were executed with perturbation factor $d = 64$, and both yielded the correct basis. The perturbed and generic walks constitute an improvement on StandardGroebnerWalk by avoiding long initial forms, in particular at the last conversion.

6 Experiments in `Macaulay2`

In this section we provide a survey of problems involving Gröbner basis computation for which the Gröbner walk algorithms may be particularly well-suited. All of the tests in this section were performed in `Macaulay2` on a machine with an intel i5-5300U processor with 16MB cache and 8GB of DDR3 RAM running `Macaulay2` version 1.22 and Linux Mint 19.0.3. We compare the built-in function `gb` (which adopts a series of heuristics and optimizations for the handling of S-pairs [RS12]) with the implementations of StandardGroebnerWalk and GenericWalk already present in `Macaulay2` as part of the `GroebnerWalk` package[2]. For the FGLM algorithm (cf. [Fau+93]) we use the `fglm` method from the package "`FGLM`". We compare our results with a version of hPerturbedWalk which was implemented by us, based on the source code of the `GroebnerWalk` package. We also wrote the two methods `verboseWalk` and `verbosegenericWalk`, which execute the standard and generic walks, outputting extra information about the size of the polynomials and coefficients encountered. The source code for these methods and for the experiments can be found in the `modifiedGWalk` repository ([Now24]).

6.1 Lexicographic Gröbner bases

One of the main motivating factors behind the introduction of Gröbner bases [Buc06] was the solution of systems of polynomial equations. Given a set of multivariate polynomials $f_1, ..., f_r \in \mathbb{Q}[x_1, .., x_n]$ we would like to determine its *affine variety*, defined as

$$\mathbf{V}(\{f_1, ..., f_s\}) := \{x \in \mathbb{C}^n \ : \ f_i(x) = 0 \ \text{ for all } \ i \in \{1, ..., r\}\}.$$

A common technique for doing this is *elimination*, which may be seen as a generalization of the Gaussian elimination/triangulation to the non-linear case. The goal here is to obtain a generating set of the ideal $I := \langle f_1, ..., f_s\rangle$ which is "well-behaved" in the sense that the elements of the Gröbner basis have terms where the variables are isolated in such a way that the system may be solved by backwards elimination. The *elimination theorem* states that this is precisely the case for the lexicographic Gröbner basis with $x_1 \succ x_2 \succ ... \succ x_n$.

Theorem 6.1 (Elimination Theorem).
Let $I \lhd \mathbb{Q}[x_1, ..., x_n]$ be an ideal and G_{lex} be a Gröbner basis of I with respect to the lexicographic ordering with $x_1 \succ x_2 \succ ... \succ x_n$. For every $0 \leq l \leq n$, the set

$$G_l := G \cap \mathbb{Q}[x_{l+1}, ..., x_n]$$

*is a Gröbner basis of the l-**th elimination ideal***

$$I_l := I \cap \mathbb{Q}[x_{l+1}, ..., x_n].$$

Proof. Consult [CLO15, pg. 121]. □

Example 6.2. The *cyclic 3-roots* problem consists of solving the following system of polynomials:

$$\begin{cases} f_1(x,y,z) = x + y + z \\ f_2(x,y,z) = xy + yz + zx, xyz - 1 \\ f_3(x,y,z) = xyz - 1 \end{cases}$$

The lexicographic Gröbner basis of $I = \langle f_1, f_2, f_3, \rangle \lhd \mathbb{Q}[x, y, z]$ is

$$G_{lex} = \{x + y + z, y^2 + yz + z^2, z^3 - 1\} =: \{g_1, g_2, g_3\}.$$

[2]The source code for this package may be found here.

The system may now be solved by backwards substitution comparable to the linear case. The condition $g_3 = 0$ implies that any solution $(x, y, z) \in \mathbf{V}(I)$ fulfills $z^3 - 1$. Substituting this in $g_2 = 0$ and subsequently $g_1 = 0$, we obtain that there are 6 complex solutions to the system, corresponding to the permutations of the third roots of unity.

Example 6.2 may be generalized in the intuitive way to the *cyclic n-roots* problem in $\mathbb{Q}[x_1, ..., x_n]$ (cf. [BF91]), which is commonly used in the benchmarking of polynomial solvers.
Systems of polynomial equations with only isolated solutions are also referred to as *zero-dimensional*; this is because the variety of the corresponding ideal is a finite set of points. For the solution of such systems, a multitude of "non-Gröbner basis" approaches have been developed: most notably numerical methods such as homotopy continuation ([BT18]). Here, we choose to compare the standard, perturbed and generic walks only with the Gröbner basis methods `gb` and FGLM. Our results are presented below.

System	Runtime in `Macaulay2` (s., approx.)									
	`gb`		Standard walk		Perturbed walk		Generic walk		FGLM	
	$\mathbb{Q}$	$\mathbb{F}_p$	$\mathbb{Q}$	$\mathbb{F}_p$	$\mathbb{Q}$	$\mathbb{F}_p$	$\mathbb{Q}$	$\mathbb{F}_p$	$\mathbb{Q}$	$\mathbb{F}_p$
`cyclic5`	0.06	0.06	0.09	0.049	0.25	0.26	0.59	0.71	0.19	0.20
`cyclic6`	3.69	$6 \cdot 10^{-}5$	2.44	0.61	2.87	1.86	11.34	12.67	1.24	0.71
`chap4`	81.02	0.17	11.01	0.39	2.92	1.01	2.31	0.85	n.a	n.a.
`katsura6`	m.o	m.o.	m.o.	m.o.	16.5	4.56	37.83	45.01	26.80	0.614

Table 3: The performance of the various Gröbner walk algorithms in a series of lexicographic conversion problems over the fields $\mathbb{Q}$ and $\mathbb{F}_p$, for $p = 32003$. "m.o." stands for "memory overload", meaning that the algorithm did not terminate. More information on the polynomial systems may be found in Appendix D.

Firstly, we observe that the performance of the standard walk is generally better than the other variants of the walk. Exceptions occur in `chap4` and `katsura6`. In `chap4` (the running example (40) from sections 4 and 5) this is because of the large coefficient swell (cf. section 5.4). In `katsura6` the second while loop of StandardGroebnerWalk is tantamount to applying Buchberger's algorithm to the ideal I: thus, the algorithm hangs for the same reasons that `gb` does.
Compared to `gb`, the performance of StandardGroebnerWalk was generally better. Where applicable, FGLM was the fastest method. However, it may only be called on zero-dimensional ideals. Passing to the finite field $\mathbb{F}_p$ generally sped up computations: the most drastic improvement is seen in `chap4`, where the large coefficients of the polynomials encountered in the intermediate bases no longer constitute a bottleneck.

6.2 Implicitization of parametric surfaces

Consider a surface in $\mathbb{R}^3$ parametrized by polynomial equations in two variables:

$$S \equiv \begin{cases} x = f_1(t_1, t_2) \\ y = f_2(t_1, t_2) \\ z = f_3(t_1, t_2) \end{cases} \quad \text{where} \quad f_1, f_2, f_3 \in \mathbb{Q}[t_1, t_2] \ , \ t_1, t_2 \in \mathbb{R}.$$

The task of *implicitization* is to express S solely in terms of zero sets of polynomials in $\mathbb{Q}[x, y, z]$. More specifically, we would like to determine the (set-theoretically) smallest ideal $I \triangleleft \mathbb{Q}[x, y, z]$ such that $S \subseteq \mathbf{V}(I)$.

Over infinite fields, this task can be reformulated as an elimination problem:

Theorem 6.3 (Implicitization theorem)**.**
Let

$$F : \mathbb{R}^2 \to \mathbb{R}^3 \ , \ (t_1, t_2) \mapsto (f_1(t_1, t_2), f_2(t_1, t_2), f_3(t_1, t_2))$$

be a parametrization of a surface $S \subset \mathbb{R}^n$*, and*

$$I := \langle x - f_1(t_1, t_2), y - f_2(t_1, t_2), z - f_3(t_1, t_2)\rangle \lhd \mathbb{R}[t_1, t_2, x, y, z]. \tag{58}$$

The variety $\mathbf{V}(I_{x,y,z})$ *of the* ***elimination ideal***

$$I_{x,y,z} := I \cap \mathbb{R}[x, y, z]$$

is the smallest variety in $\mathbb{R}^3$ *containing* S*.*

Proof. consult [CLO15, pg. 134]. $\square$

The theorem implies that one can implicitize a surface S given in parametric form by computing a reduced Gröbner basis G of the ideal I defined in (58) w.r.t a lexicographic term ordering with $t_1 \succ t_2 \succ x \succ y \succ z$. By the elimination theorem, the polynomials $G_{x,y,z} := G \cap \mathbb{R}[x, y, z]$ are a minimal implicit representation of the surface.

Example 6.4. Let $S \subset \mathbb{R}^3$ be the surface parametrized by

$$F : \mathbb{R}^2 \to \mathbb{R}^3 \ , \ (t_1, t_2) \mapsto (t_1 t_2, t_1^2, t_2^2).$$

To calculate its implicit representation, we compute the reduced Gröbner basis of the ideal

$$I = \langle x - t_1 t_2, y - t_1^2, z - t_2^2\rangle$$

with respect to the lexicographic ordering with $t_1 \succ t_2 \succ x \succ y \succ z$. This is

$$G = \left\{\mathbf{x^2} - \mathbf{yz}, zu - xv, xu - yv, v^2 - z, uv - x, u^2 - y\right\},$$

where only the polynomial in bold lies in $\mathbb{Q}[x, y, z]$. Thus, the variety $\mathbf{V}(\langle x^2 - yz\rangle)$ is the smallest variety containing S.

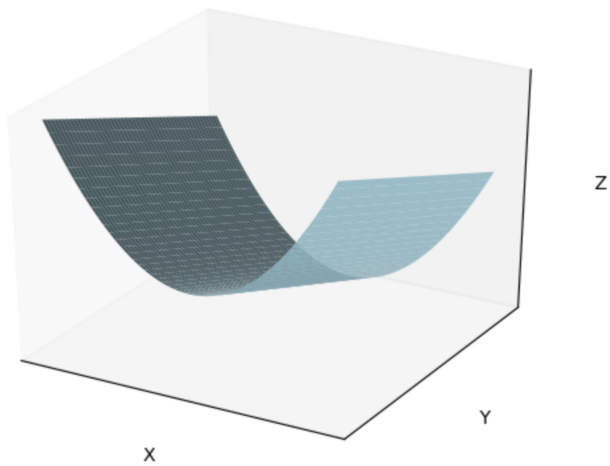


Figure 7: The surface $\mathbf{V}(\langle x^2 - yz\rangle)$ from Example 6.4 in $\mathbb{R}^3$

Implicitization is an application for which the Gröbner walk is especially well-suited. Its performance for the types of computations involved is consistently better than Buchberger's algorithm, and the paper in which the deterministically perturbed Gröbner walk was introduced ([Tra00]) mentioned implicitization as the primary application. Furthermore, in this setting the ideals upon which the computations are being performed are always of positive dimension; because of this, FGLM is not applicable. An overview of the performance of the Gröbner walk and its variants in `Macaulay2` is presented below:

System	Runtime in `Macaulauy2` (s., approx.)							
	`gb`		Standard walk		Perturbed walk		Generic walk	
	$\mathbb{Q}$	$\mathbb{F}_p$	$\mathbb{Q}$	$\mathbb{F}_p$	$\mathbb{Q}$	$\mathbb{F}_p$	$\mathbb{Q}$	$\mathbb{F}_p$
`newellp1`	52.2	1.81	24.1	3.89	18.02	9.27	47.1	18.63
`newellp4`	0.02	0.03	0.2	0.12	0.37	0.12	0.24	0.21
`newellp25`	m.o	8.01	72.04	9.37	132.9	28.36	m.o	59.16
`agk4`	5.94	0.34	3.19	0.94	6.86	2.49	6.55	4.61
`agk8`	51.00	0.92	9.2	2.14	15.86	6.51	80	21.64

Table 4: The performance of the various Gröbner walk algorithms in a series of implicitization problems with coefficients in $\mathbb{Q}$ and $\mathbb{F}_p$ with $p = 32003$. "m.o" stands for memory overload, meaning that the algorithm did not terminate.

We observe that the performance of StandardGroebnerWalk is consistently better than that of `gb` over $\mathbb{Q}$. Although passing to $\mathbb{F}_p$ sped up computations, upon doing this `gb` was again the fastest method. This suggests that the computational bottleneck in these examples is coefficient swell. On the other hand, the perturbed and generic walks did not perform better than the standard walk, implying that the intermediate conversions on these examples are not significantly faster when executed on facets.

6.3 Integer programming and toric ideals

Let $a_1, ..., a_n, b \in \mathbb{N}$ and consider the problem of determining whether the equation

$$x_1a_1 + x_2a_2 + ... + x_na_n = b$$

has a non-negative integer solution $x = (x_1, ..., x_n) \in \mathbb{N}^n$. This may be solved by the following linear program:

$$\textit{Minimize} \quad t \in \mathbb{N} \quad , \quad \textit{subject to} \;\; t + x_1a_1 + x_2a_2 + ... + x_na_n \leq b \;\; \textit{and} \;\; t, x_1, ..., x_n \geq 0. \tag{59}$$

Using standard algebraic methods, this linear program can be solved using Gröbner bases.

Proposition 6.5. *Let $b, a_1, ..., a_n \in \mathbb{N}$. We define the corresponding ideal*

$$I := \langle x_1 - t^{a_1}, x_2 - t^{a_2}, ..., x_n - t^{a_n} \rangle \lhd \mathbb{Q}[t, x_1, ..., x_n]. \tag{60}$$

A solution to (59) *is given by the exponent vector of the monomial*

$$\overline{t^b}^{I,\prec} , \tag{61}$$

where $\prec$ is any ordering such that $t \succ x_1 \succ ... \succ x_n$. In particular, the problem has an optimal solution if the variable t does not appear in (61).

Proof. Consult [Stu95, page 43]. □

In some settings (for example in the coin change problem), it may be desirable minimize the total entries of the solution $\sum |x_i|$. This can be done by ordering the 'a_i's from largest to smallest and computing a graded lexicographic Gröbner basis.

Example 6.6. Let $a_1 = 5$, $a_2 = 12$ and $a_3 = 20$. We wish to solve the linear program in (59) with $b = 43$. The corresponding toric ideal is

$$I = \langle x_1 - t^5, x_2 - t^12, x_3 - t^{20} \rangle. \tag{62}$$

We compute a marked Gröbner basis of I_A w.r.t any monomial order $\prec$ which prioritizes t over x_i (for example, the lexicographic ordering $lex_{t \succ x_1 \succ x_2 \succ x_3}$). This is

$$G_\prec = \left\{ \underline{x_1^4} - x_3,\ \underline{x_2^5} - x_3^3,\ \underline{tx_3^2} - x_1 x_2^3,\ \underline{tx_2^2} - x_1 x_3,\ \underline{tx_1^3 x_3} - x_2^3,\ \underline{t^2 x_3} - x_1^2 x_2,\ \underline{t^2 x_1^2} - x_2,\ \underline{t^3 x_2} - x_1^3,\ \underline{t^5} - x_1 \right\}$$

The normal form of the monomial $t^b = t^{43}$ w.r.t G is $tx_1^2 x_2 x_3$ and indeed

$$43 = 1 + 2 \cdot 5 + 1 \cdot 12 + 1 \cdot 20 = 1 + 2 \cdot a_1 + 1 \cdot a_2 + 1 \cdot a_3.$$

According to [Fuk+07] the generic Gröbner walk constitutes a significant improvement on Buchberger's algorithm for conversions of this type. Using `Macaulay2`, we were unable to replicate the results which they report in section 6 ([Fuk+07, pg.14]) on their set of knapsack problems, which originally come from [AL04]. In our experiments, the generic Gröbner walk did not terminate in less than 10,000 seconds in any of their "`cuww`" examples. Because of this, we decided to instead perform experiments on smaller ideals of the form (62): we chose $n = 5$, and random a_i between 1 and 10^4.

System		Timings in `Macaulay2` (s., approx.)			
	$\lvert G_{glex} \rvert$	`gb`	Buchberger	Standard walk	Generic walk
`randomknap1`	215	0.16	483.15	3.28	5.31
`randomknap2`	251	0.67	$> 10,000$	2.24	4.56
`randomknap3`	161	0.055	5483.41	1.27	1.81
`randomknap4`	271	0.77	55.72	6.15	10.07
`randomknap5`	230	0.52	2053.95	16.21	25.11

Table 5: The performance of the various Gröbner basis algorithms in knapsack problems with random a_i. For Buchberger's algorithm we used the method `gfanBuchberger` from the "`gfanInterface`" package.

While the results confirm that Gröbner walk methods are a significant improvement on a naive implementation of Buchberger's algorithm, using `gb` is clearly the method of choice. In this case, this is because the underlying ideal is both toric and homogeneous with respect to the weight vector

$$\omega = \left(\prod_{i=1}^{n} a_i \ ,\ \prod_{i=2}^{n} a_i \ ,\ \ldots ,\ \prod_{i \neq k}^{n} a_i \ ,\ \ldots ,\ \prod_{i=1}^{n-1} a_i \right), \tag{63}$$

where the variables are ordered as t, x_1, x_2, ..., x_n.

Further analysis of the intermediate computations with the `verboseWalk` methods explain the inferior performance of the generic walk. While the generic walk always entailed more intermediate conversions, there is no advantage over the standard Gröbner walk in this particular setting: in both methods, all of the initial forms encountered are either monomials or binomials, meaning. Our implementation of hPerturbedWalk was not applicable on these ideals in `Macaulay2` as the degree of the polynomials is too high. However, we do not expect the perturbed walk to perform better than the standard walk for the reasons we have just stated.

The ideals of the form (62) belong to a class of ideal which are ubiquitous in applications: toric ideals. Specialized Gröbner basis methods for these ideals (for example those based on [CT91]) exist and are integrated in the `gb` function by default. To compare the algorithms on toric ideals, we generated random matrices with 8 columns and 4 rows and entries up to 16, and constructed the corresponding toric ideal with the functions `toricMarkov` and `toBinomial`. We then used the methods `gb`, StandardGroebnerWalk and GenericWalk and `gfanBuchberger` to compute lexicographic Gröbner bases.

System	Timings in `Macaulay2` (s., approx.)			
	`gb`	Buchberger	Standard walk	Generic walk
`toric1`	0.004	0.35	20.67	51.63
`toric2`	1.71	16.7	87.06	659.98
`toric3`	0.09	0.59	55.32	89.07
`toric4`	0.06	0.27	16.13	28.18
`toric5`	0.92	0.24	11.78	28.92

Table 6: The performance of the various Gröbner basis algorithms for the computation of lexicographic bases of toric ideals with 4 generators over 8 variables.

The performance of `gb` and the Gröbner walk algorithms is comparable to the knapsack case. The dramatic improvement of Buchberger's algorithm may be attributed to the lower degree of the polynomials encountered. While the Gröbner walk approach is clearly not the method of choice for toric ideals for computing a single Gröbner basis, it may still be of interest if one wants information about the intermediate bases and/or more generally, the Gröbner fan. In the knapsack case, each intermediate Gröbner basis computation corresponds to solving the linear program (59) with some other b, meaning that it may be beneficial to cache them.

We conclude with some heuristics for the use of Gröbner basis algorithms in `Macaulay2` based on our observations:

- For computing lexicographic bases of 0-dimensional ideals, FGLM is the preferred method.
- For computing lexicographic bases of ideals of positive dimension, the standard or generic walk may perform well on ideals where the `gb` method hangs.
- StandardGroebnerWalk is the method of choice for Gröbner basis approaches to problems of implicitization.
- For toric ideals, `gb` and other specialized algorithms are the methods of choice.
- Any benefit of the perturbed or generic walks over the StandardGroebnerWalk is highly situational. The theoretical advantages of the former methods are rarely reflected in performance.

7 Implementation in `OSCAR`: The `GroebnerWalk` package

We implemented the standard, generic and perturbed Gröbner walks in the `OSCAR` computer algebra system ([24]). In its current state, the code is a stand-alone `julia` package with dependency on `OSCAR` called `GroebnerWalk`, and it was developed in collaboration with Kamillo Ferry. Special thanks go to Jordi Welp (University of Oldenburg), who laid the groundwork of the code as part of his master's thesis. It is our hope that the code will be included in an upcoming stable version of `OSCAR` as an option for the function `groebner_basis`. In this section, we provide a brief description of the functionality of the code, as well as some preliminary benchmark results. The source code (including the examples mentioned in section 7.2) may be found here.

7.1 Functionality

To demonstrate the functionality of `GroebnerWalk` we revisit the ideal (35) from Section 3. Let

$$I = \langle x^2 + yz, xy + z^2 \rangle \lhd \mathbb{Q}[x, y, z],$$

and suppose we are tasked with computing a lexicographic Gröbner basis. The entire functionality of the package is accessed via the `groebner_walk` function, as demonstrated below.

```
using Oscar, GroebnerWalk # load necessary packages

R, (x,y,z) = polynomial_ring(QQ, ["x", "y", "z"]) # define the ring...

I = ideal([x^2 + y*z, x*y + z^2]) # ...and the ideal

groebner_walk(I, lex(R)) # compute the Groebner basis
```

By default, the starting basis is the reduced Gröbner basis w.r.t the graded reverse lexicographic order. If we would like to do a Gröbner walk with a different starting order, this can be specified by adding a third argument. Furthermore, the intermediate weight vectors computed may be retrieved using `set_verbosity_level`. We demonstrate these features by executing the walk described in Section 3.3:

```
set_verbosity_level(:groebner_walk, 1) # set verbosity level

start_order = lex(R) # specify starting...

target_order = matrix_ordering(R, [1 3 0; 1 1 1; 1 0 0 ]) # ...and target orders

groebner_walk(I, target_order, start_order) # walk from start to target order
```

Running this gives the following output:

```
Results for standard_walk
Crossed Cones in:
ZZRingElem[1, 0, 0]
ZZRingElem[2, 1, 0]
ZZRingElem[1, 2, 0]
Cones crossed: 3
Groebner basis with elements
1 -> y*z + x^2
2 -> x^3 - z^3
3 -> x*y + z^2
with respect to the ordering
matrix_ordering([x, y, z], [1 3 0; 1 1 1; 1 0 0])
```

The selection of the algorithm used is done by an (optional) keyword argument `algorithm=:`. At time of writing, the three options available are `standard`, `generic` and `perturbed`. Calling `groebner_walk(I, target_order, start_order, algorithm =:generic)` yields the following output:

```
Results for generic_walk
Facets crossed for:
ZZRingElem[1, -2, 1]
ZZRingElem[2, -1, -1]
Cones crossed: 2
Groebner basis with elements
1 -> y*z + x^2
2 -> x^3 - z^3
3 -> x*y + z^2
with respect to the ordering
matrix_ordering([x, y, z], [1 3 0; 1 1 1; 1 0 0])
```

7.2 Preliminary benchmarks

We executed benchmarks on selected computational tasks already used for the experiments in Section 6. We ran the comparisions on the same machine described in Section 6 running Linux Mint 19.03, Julia 1.10.4 and `OSCAR` 1.0.3. The results are presented below:

Table 7: The results obtained for several systems from section 6, taken over $\mathbb{Q}$ and $\mathbb{F}_p$ with $p = 32003$. The entry "m.o" is present when the algorithm did not terminate due to memory overload

	Runtime in `OSCAR` (s., approx)					
Ideal	`groebner_basis`		Standard walk		Generic walk	
	$\mathbb{Q}$	$\mathbb{F}_p$	$\mathbb{Q}$	$\mathbb{F}_p$	$\mathbb{Q}$	$\mathbb{F}_p$
`cyclic5`	0.07	0.05	0.08	0.07	0.91	0.879
`cyclic6`	5928.51	0.24	0.93	0.61	17.85	28.41
`agk4`	1407.76	$4 \cdot 10^{-5}$	23.79	5.33	214.75	201.69
`newellp1`	2081.76	0.40	25.51	17.33	6042.97	4597.38
`randomknap4`	0.16	0.14	97.30	77.38	19.98	49.81
`chap4`	m.o	2.78	1.49	3.46	59.60	48.96

The results show that in many cases the standard Gröbner walk constitutes a significant improvement on the state of the art for ideals defined over $\mathbb{Q}$. This, combined with the fact, at the time of writing, not all widely-used Gröbner basis algorithms have been fully implemented in `OSCAR`[3], means that our implementation of the standard walk may be preferable to `groebner_basis` in a variety of contexts.

Comparing the computation timings with those from the `Macaulay2` experiments from the previous chapter, we notice that the computations here are generally slower, sometimes by several orders of magnitude. We attribute this to the fact that `groebner_basis` (which uses methods from `Singular`) contains fewer of the heuristics present in the `gb` method in `Macaulay2`. Furthermore, it is a known issue that the reduction of polynomials with the multivariate division algorithm is excessively memory-intensive in `OSCAR`. This would help to explain to the comparatively poor performance of the generic

[3]Faugere's F4 algorithm [Fau99] may only be called over finite fields, whereas the F5 algorithm is as of yet not implemented. The "Hilbert-driven" Buchberger algorithm [Sim14] threw an error in our larger examples for reasons we have yet to investigate.

walk, in which many more conversions/reductions generally occur.

Table 8: Side-by-side comparison of the default Gröbner basis methods and the standard and generic walks in OSCAR and Macaulay2. All of these computations were carried out over $\mathbb{Q}$. "m.o" stands for memory overload.

Ideal	Runtime (s., approx)					
	groebner_basis / gb		Standard walk		Generic walk	
	OSCAR	M2	OSCAR	M2	OSCAR	M2
cyclic5	0.07	0.06	0.08	0.09	0.91	0.59
cyclic6	5928.51	3.69	0.93	2.44	17.85	11.34
agk4	1407.76	5.94	23.79	3.19	214.75	0.24
newellp1	2081.76	52.2	25.51	24.1	6042.97	47.10
randomknap4	0.16	0.77	97.30	6.15	19.98	10.07
chap4	m.o	81.02	1.49	11.01	59.60	2.31

7.3 Outlook

We plan to perform further tests in the upcoming weeks to investigate the bottlenecks encountered by groebner_basis and help interpret our results. The GroebnerWalk package remains a work in progress. We intend to optimize our code for more efficient interactions with both the Singular backend, as well as the subroutines for the computation and lifting of initial forms already implemented in OSCAR as part of the groebner_fan method.

8 Concluding remarks

We started by providing a comprehensive resource for the theory behind the Gröbner walk as introduced by [CKM97], as well as the variations proposed in [Tra00] and [Fuk+07]. This includes a formalization of these algorithms (Algorithm 1, Algorithm 8, Algorithm 11) and rigorous proofs of correctness (Theorem 2.12, Theorem 2.9, Theorem 5.7). The formulation and proof of these results, as well as those of Proposition 4.1 and Lemma 2.8, and all of the examples provided are novel contributions, so feedback and corrections on these parts are especially welcome.

On the implementation side, the experiments from Section 6 and the `OSCAR` implementation are novel contributions. We made efforts to ensure replicability, and invite the reader to conduct the experiments themselves; to this end, we would welcome any alternative results and/or interpretations of our data. The decision to conduct our experiments primarily in `Macaulay2` instead of `OSCAR` was a practical one: many of our heavier examples in Section 6 did not terminate reasonable time in `OSCAR` using any Gröbner basis conversion algorithm. We are currently investigating the reasons for this.

In `OSCAR`, our implementation of the standard Gröbner walk appears to be an improvement on the state of the art. The `GroebnerWalk` package has been staged for inclusion into `experimental`. We hope to obtain suggestions and feedback at the upcoming MEGA 2024 conference in Leipzig, where we will hold a computer presentation presenting the package. A jupyter notebook containing a 4-variable example (with a corresponding 3-dimensional visualization using polymake) is planned for the presentation, and will be made available at the repository ([Now24]) once completed. We also plan to submit a short paper about the package to JSAG.

Avenues for further work include:

- Implementation and experiments involving other variants of the Gröbner walk such as the "fractal" ([AGK97]) and "parametric" walks ([HDB17]).
- Discussion of the relevance of the Gröbner walk for computing Gröbner fans ([FJT07]) and in the tropical setting ([Jen07, Chapter 7]).
- A further investigation of the computational bottlenecks encountered in the generic walk, particularly in `OSCAR`.

Questions, corrections, and comments may be sent to `nowell@tu-berlin.de`.

A Ideals and Gröbner bases

We provide an overview of the theory on ideals and Gröbner bases required to introduce the Gröbner walk algorithm. For proofs and more context, we refer to chapter 2 of the book "Ideals, varieties and algorithms" ([CLO15], pg. 49). Throughout this section, K denotes a field and $K[x_1, ..., x_n]$ is the polynomial ring in n variables with coefficients over K. We denote the set of all monomials in $K[x_1, ..., x_n]$ by $\mathrm{Mon}_n(x)$. Although the results below hold for arbitrary fields unless otherwise stated, we are most interested in the case $K = \mathbb{Q}$.

A.1 Ideals

Definition A.1. An **ideal** I in $K[x_1, ..., x_n]$ is a subset of $K[x_1, \dots, x_n]$ fulfilling the following three conditions

(i) I contains the zero polynomial: $0 \in I$.

(ii) I is closed under addition: $f, g \in I \implies f + g \in I$.

(iii) I is closed under multiplication with any element of $K[x_1, ..., x_n]$, i.e.

$$\sum_{i=1}^{s} f_i h_i \in I \qquad \text{for any} \quad f_1, ..., f_s \in I \quad , \quad h_1, ..., h_s \in K[x_1, ..., x_n].$$

If $I \subset K[x_1, ..., x_n]$ is an ideal, we write $I \lhd K[x_1, ..., x_n]$. Notable examples of ideals include the ring itself $I = K[x_1, ..., x_n]$ the zero ideal $I = \{0\}$, kernels of ring homomorphisms and vanishing ideals of affine varieties.

Definition A.2. Let $F \subset K[x_1, ..., x_n]$.

- The set
$$\langle F \rangle := \left\{ \sum_{i=1}^{s} f_i h_i \ : f_i \in F \ , \ h_i \in K[x_1, ..., x_n] \ , \ s \in \mathbb{N} \right\}$$
is called the ideal **generated** by F. It is the smallest ideal containing F.
- For an ideal I, we call a set F such that $\langle F \rangle = I$ a **generating set** (or basis) of I.
- We call I a **monomial ideal** if it has a generating set consisting solely of monomials.

The following lemma is instrumental for the proof of several classical results.

Lemma A.3. *Let $I = \langle x^\alpha \ , \ \alpha \in A \rangle$ be a monomial ideal (where $A \subset \mathbb{N}^n$). Then for any monomial $x^\beta \in \mathrm{Mon}_n(x)$ the following holds:*

$$\begin{aligned} x^\beta \in I &\iff x^{\alpha'} | x^\beta \text{ for some } \alpha' \in A \\ &\iff \text{There exist vectors } \alpha' \in A \ , \ \gamma \in \mathbb{N}^n \quad \text{such that} \quad x^\beta = x^{\alpha'} x^\gamma = x^{\alpha' + \gamma}. \end{aligned}$$

Proof. Consult [CLO15, pg. 70]. □

For example, Lemma A.3 is an ingredient in the proof of the Hilbert basis theorem:

Theorem A.4. *(Hilbert basis theorem) Any ideal $I \lhd K[x_1, ..., x_n]$ has a finite generating set $\{f_1, ..., f_s\}$.*

The Theorem can be proven either by deriving it as an immediate corollary of the claim that $K[x_1, ..., x_n]$ is a Noetherian ring, or by introducing a multivariate division algorithm and using this in combination with Lemma A.3.

A.2 Monomial orders and the multivariate division algorithm

The multivariate division algorithm is crucial in solving systems of polynomial equations. To introduce it, we first need to introduce a special kind of ordering relation on the set of all monomials in n variables $\mathrm{Mon}_n(x)$

Definition A.5. A **monomial order** is a relation $\prec$ on the set $\mathrm{Mon}_n(x)$ with the following three properties:

(i) $\prec$ is a strict total ordering, i.e.

$$\text{for all } \alpha, \beta \in \mathbb{N}^n \text{ with } \alpha \neq \beta, \text{ either } x^\beta \prec x^\alpha \text{ or } x^\alpha \prec x^\beta.$$

(ii) $\prec$ is admissible, i.e.

$$\text{for all } \alpha, \beta, \gamma \in \mathbb{N}^n : x^\beta \prec x^\alpha \implies x^{\beta+\gamma} \prec x^{\alpha+\gamma}$$

(iii) $\prec$ is a well-order, i.e.

$$\text{for any non-empty collection of monomials } M \subset \mathrm{Mon}_n(x), \text{ there exists an } x^\alpha \in M \text{ such that } x^\alpha \prec x^\beta \text{ for all } x^\beta \in M.$$

Example A.6. Some well-known examples of monomial orders are:

- the **lexicographic order** *lex*:

$$x^\beta \prec_{lex} x^\alpha \iff \text{the leftmost non-zero entry of } \alpha - \beta \text{ is positive.}$$

- the **graded reverse lexicographic** order *grevlex*:

$$x^\beta \prec_{grevlex} x^\alpha \iff \begin{array}{l} \sum_{i=1}^n \beta_i < \sum_{i=1}^n \alpha_i \ \ \textbf{or} \\ \sum_{i=1}^n \beta_i = \sum_{i=1}^n \alpha_i \text{ and the rightmost non-zero entry of } \alpha - \beta \text{ is negative.} \end{array}$$

- the **refinement monomial order** of a weight vector $\omega \in \mathbb{Q}^n_{\geq 0}$ w.r.t a given monomial order $\prec$. This is the relation $\prec_\omega$ defined as

$$x^\beta \prec_\omega x^\alpha \quad :\iff \quad \langle \omega, \beta \rangle < \langle \omega, \alpha \rangle \quad \text{or} \quad \big(\langle \omega, \beta \rangle = \langle \omega, \alpha \rangle \quad \text{and} \quad x^\beta \prec x^\alpha\big).$$

Refinement orders appear numerous times in the Gröbner walk algorithm. Therefore, we provide a proof of the fact that they are indeed monomial orders. To do this, we first recall a helpful characterization of the well-ordering property for admissible total orders.

Proposition A.7. *If $\prec$ is an admissable total ordering on $\mathrm{Mon}_n(x)$ (i.e. a relation already fulfilling (i) and (ii) from the Definition A.5), then the following holds:*

$$\prec \textit{ is a well-order} \quad \iff \quad 1 = x^{(0,\dots,0)} \prec x^\alpha \quad \textit{for all } \alpha \in \mathbb{N}^n \setminus \{0\}.$$

Proof. [CLO15] proves this as a corollary of Dickson's lemma, which is the preliminary version of the Hilbert basis theorem for monomial ideals (consult pg. 73). ☐

Proposition A.8. *Let $\prec$ be a monomial order on $\mathrm{Mon}_n(x)$ and $\omega \in \mathbb{Q}^n_{\geq 0}$. Then the following holds:*

$$\prec_\omega \textit{ is a monomial order.}$$

Proof. We show that $\prec_\omega$ has the properties *(i)* - *(iii)* of Definition A.5.

(i): Let $\alpha, \beta \in \mathbb{N}^n$, $\alpha \neq \beta$. Then either $x^\alpha \prec x^\beta$ or $x^\beta \prec x^\alpha$ hold (as $\prec$ is a total order). Thus, either $x^\alpha \prec_\omega x^\beta$ or $x^\beta \prec_\omega x^\alpha$, regardless of whether $\langle \omega, \alpha \rangle = \langle \omega, \beta \rangle$ holds or not.

(ii) follows from admissibility of $\prec$ together with the observation that

$$\langle \omega, \alpha \rangle < \langle \omega, \beta \rangle \implies \langle \omega, \alpha + \gamma \rangle < \langle \omega, \beta + \gamma \rangle \quad \text{for any } \alpha, \beta, \gamma \in \mathbb{N}^n.$$

(iii) follows from Proposition A.7, as $\omega \in \mathbb{Q}^n_{\geq 0}$ implies $\langle \omega, 0 \rangle = 0 < \langle \omega, \alpha \rangle$ and consequently

$$1 \prec x^\alpha \quad \text{for all } \alpha \in \mathbb{N}^n,\ \alpha \neq (0, ..., 0).$$

$\square$

Once we have fixed a monomial order $\prec$, we may define leading terms/coefficients of a polynomial in an analogous manner to the 1-variable case.

Definition A.9. Let $\prec$ be a monomial order, $f \in K[x_1, ..., x_n]$, $f \neq 0$. We may write the terms of f in descending order w.r.t $\prec$. That is,

$$f = c_{\alpha_1} x^{\alpha_1} + c_{\alpha_2} x^{\alpha_2} + ... + c_{\alpha_k} x^{\alpha_k}$$

where $c_{\alpha_i} \in K \setminus \{0\}$ and $x^{\alpha_{i+1}} \prec x^{\alpha_i}$ for all i.

- The **leading term** of f w.r.t $\prec$ is

$$\mathrm{LT}_\prec(f) := c_{\alpha_1} x^{\alpha_1}.$$

- The **initial monomial** (or leading monomial) of f w.r.t $\prec$ is

$$in_\prec(f) := x^{\alpha_1}.$$

- The **leading coefficient** of f w.r.t $\prec$ is

$$\mathrm{LC}_\prec(f) := c_{\alpha_1}.$$

These are all of the notions necessary to introduce the multivariate division algorithm.

Theorem A.10. *Let $\prec$ be a monomial order on $\mathrm{Mon}_n(x)$, $F = (f_1, ..., f_s)$ be an ordered set of non-zero polynomials in $K[x_1, .., x_n]$. For each $f \in K[x_1, ..., x_n]$ there exist polynomials $r, q_1, ..., q_s \in K[x_1, ..., x_n]$ such that the following three statements hold:*

(i) $f = q_1 f_1 + q_2 f_2 + ... + q_s f_s + r$.

(ii) Either $r = 0$ holds, or $r \neq 0$ and no term of r is divisible by $\mathrm{LT}_\prec(f_i)$ for all $i \in \{1, .., s\}$.

(iii) For all i, either $in_\prec(q_i f_i) = in_\prec(f)$ or $in_\prec(q_i f_i) \prec in_\prec(f)$.

Proof. The proof is constructive: the polynomials q_i and r are the quotients and remainder respectively, obtained upon dividing f by F with the *multivariate division algorithm.* For a description of this algorithm as well as a proof of the result we refer to [CLO15, pg.64]. $\square$

It is clear that in the setting above, if $r = 0$ then $f \in \langle F \rangle$. The converse doesn't hold for general generating sets of $\langle F \rangle$ (Consult [CLO15, pg. 68] for an example of this.). This motivates the introduction of a special kind of generating set: Gröbner bases.

A.3 Gröbner bases

Definition A.11. Let $\prec$ be a monomial order.

- For a set $F \subset K[x_1, ..., x_n]$, the **initial ideal** of F w.r.t $\prec$ is the ideal generated by the leading monomials of elements of F:

$$in_{\prec}(I) := \langle \{ in_{\prec}(f), \quad \text{where} \quad f \in F \} \rangle$$

- A finite set of polynomials G is called a **Gröbner basis** of an ideal $I \triangleleft K[x_1, ..., x_n]$ w.r.t $\prec$, if

$$in_{\prec}(G) = in_{\prec}(I).$$

Every ideal I has a Gröbner basis w.r.t $\prec$; this follows constructively from the results in the following section. Furthermore, Gröbner bases solve the *ideal membership problem*, i.e the task of determining whether $f \in I$ holds for some $I \triangleleft K[x_1, ..., x_n]$, $f \in K[x_1, ..., x_n]$. We state this fact as a modified version of Theorem A.10.

Proposition A.12. *Let $\prec$ be a monomial order, $I \triangleleft K[x_1, ..., x_n]$ be an ideal and $G = \{g_1, ..., g_s\}$ be a Gröbner basis of I w.r.t $\prec$. For any non-zero polynomial $f \in K[x_1, ..., x_n]$ there exists a unique $r \in K[x_1, ..., x_n]$ such that:*

(i) No term of r is divisible by any $\mathrm{LT}_{\prec}(g_i)$, $i \in \{1, ..., r\}$.

(ii) There exists a $g \in I$ such that $f = g + r$.

Proof. Consult [CLO15, pg.83]. □

Remark A.13. By Lemma A.3, condition (ii) is equivalent to requiring that no term of r lies in $in_{\prec}(I)$. Monomials not in $in_{\prec}(I)$ are sometimes referred to as *standard monomials*. Using this terminology, we can reformulate (ii) as the requirement that r be a (possibly empty) linear combination of standard monomials.

Definition A.14. The polynomial r from the Proposition A.12 is called the **residue** (or alternatively, the remainder or normal form) of f w.r.t I and $\prec$. We denote it by $\overline{f}^{I,\prec}$.
When the underlying ideal is clear from context, we may simply write $\overline{f}^{\prec}$.

Proposition A.12 states that $\overline{f}^{I,\prec}$ is unique, depends only on the monomial order $\prec$ and I, and is a linear combination of standard monomials. Crucially, it is not dependent on the choice of generating set of I. For any fixed ideal I and a monomial order $\prec$, there exist infinitely many Gröbner bases of I. To introduce a notion of uniqueness, we turn our attention to Gröbner bases with extra properties.

Definition A.15. Let $\prec$ be a monomial order, $I \triangleleft K[x_1, ..., x_n]$ an ideal.
A Gröbner basis $G = \{g_1, ..., g_r\}$ is called **reduced**, if it has the following additional properties:

(i) G is minimal w.r.t inclusion, i.e.

$$in_{\prec}(G \setminus \{g_i\}) \subsetneq in_{\prec}(I) \quad \text{for all } i \in \{1, ..., r\}.$$

(ii) G is monic, i.e.

$$\mathrm{LC}_{\prec}(g_i) = 1 \qquad \text{for all } i \in \{1, ..., r\}.$$

(iii) G is reduced, i.e.

For all $i, j \in \{1, ..., r\}$ with $i \neq j$, no term of g_i is divisible by $in_{\prec}(g_j)$.

An ideal may have the same reduced Gröbner basis w.r.t two distinct monomial orders $\prec_1$ and $\prec_2$. This can be the case even if $in_{\prec_1}(I) \neq in_{\prec_2}(I)$ For example, it is clear that $\{x+y\}$ is a reduced Gröbner basis of the ideal $I = \langle x+y \rangle \lhd \mathbb{Q}[x,y]$ w.r.t both $lex_{x \succ y}$ and $lex_{y \succ x}$. This motivates the notion of a *marked* Gröbner basis.

Definition A.16. We call a reduced Gröbner basis $G = \{g_1, ..., g_s\}$ of an ideal I w.r.t $\prec$ with the leading terms identified a **marked** Gröbner basis of I w.r.t $\prec$. Formally, it is defined as the set of ordered pairs $(g_i, in_\prec(g_i))$. In our notation, we denote these bases by $G_\prec$ and identify the leading monomial of each polynomial by underlining.

$$G_\prec = \left\{ \underline{in_\prec(g_i)} + \sum \text{remaining terms of } g_i \;\mid\; i \in \{1, ..., s\} \right\}.$$

(We note that due to the monicity property, we have $\mathrm{LT}_\prec(g_i) = in_\prec(i)$ for all $i \in \{1, ..., s\}$.)

Proposition A.17. *Let $I \lhd K[x_1, ..., x_n]$ be an ideal. The set of all marked Gröbner bases is in 1-1 correspondence with the set of all initial ideals.*

Proof. Let $in_\prec(I)$ be an initial ideal of I w.r.t some monomial order $\prec$, and let

$$G = \left\{ \underline{in_\prec(g_i)} + \sum \text{remaining terms of } g_i \;\mid\; i \in \{1, ..., s\} \right\} \text{ and}$$
$$\tilde{G} = \left\{ \underline{in_\prec(\tilde{g_j})} + \sum \text{remaining terms of } \tilde{g_j} \;\mid\; j \in \{1, ..., r\} \right\}$$

be two marked Gröbner bases of I w.r.t $\prec$. That is,

$$in_\prec(I) = \langle \{in_\prec(g_1), ..., in_\prec(g_s)\} \rangle = \langle \{in_\prec(\tilde{g_1}), ...in_\prec(\tilde{g_r}) \rangle \}$$

holds (by definition of a Gröbner basis) and G and $\tilde{G}$ both have the properties (i)-(iii) of Definition A.15. Our goal is to show $G = \tilde{G}$.
Lemma A.3 implies that for each $i \in \{1, ..., s\}$, $in_\prec(g_i)$ is divisible by $in_\prec(\tilde{g_j})$ for some unique $j \in \{1, ..., r\}$ and, by the same argument, each $in_\prec(\tilde{g_j})$ is divisible by some $in_\prec(g_i)$. (Uniqueness follows from the fact that G and $\tilde{G}$ are both reduced. This implies $r = s$ and, upon reordering, we may assume that $in_\prec(g_i) = in_\prec(\tilde{g_i})$ holds for all $i \in \{1, ..., s\}$.
It remains to show that $g_i = \tilde{g_i}$ holds for all $i \in \{1, ..., s\}$. Assuming $g_i \neq \tilde{g_i}$ for some i, then $g_i - \tilde{g_i} \in I \setminus \{0\}$, and therefore $x^\beta := in_\prec(g_i - \tilde{g_i}) \in in_\prec(I)$. By Lemma A.3, x^β is divisible by some $in_\prec(g_i)$ and some $in_\prec(\tilde{g_j})$. However, as $in_\prec(g_i - \tilde{g_i})$ is a term of either g_i or $\tilde{g_i}$ this would contradict the reducedness of G and $\tilde{G}$. Thus, $g_i = \tilde{g_i}$, completing the proof. □

An immediate consequence of this result is that marked Gröbner bases are unique in the following sense:

Corollary A.18. *Let $I \lhd K[x_1, ..., x_n]$ be a monomial ideal. If $\prec$ and $\prec'$ are monomial orders such that $in_\prec(I) = in_{\prec'}(I)$, then $G_\prec = G_{\prec'}$.*

Finally, it is a well-known result that, given an ideal $I \lhd K[x_1, ..., x_n]$, there are only finitely many initial ideals. (For a proof of this, consult [Stu95, pg.1]) Thus:

Corollary A.19. *For some ideal $I \lhd K[x_1, ..., x_n]$ The set of all marked Gröbner bases*

$$\{ G_\prec \;,\; \prec \text{ is a monomial order } \}$$

is finite.

A.4 Computing Gröbner bases

Let F be an arbitrary finite generating set of an ideal $I \lhd K[x_1, ..., x_n]$ and $\prec$ be a monomial order. F may be converted to a Gröbner basis of I w.r.t. $\prec$ using *Buchberger's Algorithm* (cf. [Buc06]). This algorithm works by computing and reducing so-called *S-pairs* of elements of F. We provide a general description of the algorithm below.

Definition A.20. Let $f, g \in K[x_1, ..., x_n]$ be two non-zero polynomials and $\prec$ be a monomial order. We denote the least common multiple of the the monomials $in_\prec(f)$ and $in_\prec(g)$ by x^γ.
The **S-pair** of f and g with respect to $\prec$ is the polynomial

$$S(f,g) := \frac{x^\gamma}{\mathrm{LT}_\prec(f)} \cdot f - \frac{x^\gamma}{\mathrm{LT}_\prec(g)} \cdot g. \tag{64}$$

For two polynomials $f, g \in F$ in the generating set of I, the leading term of $S(f,g)$ is by construction neither $\mathrm{LT}_\prec(f)$ nor $\mathrm{LT}_\prec(g)$. If this polynomial does not reduce to zero upon division by F, then $\mathrm{LT}_\prec(S(f,g)) \notin in_\prec(I)$. In this case, Buchberger's algorithm adds this remainder to F. This process is repeated until all S-pairs reduce to zero, which is precisely the case when F is a Gröbner basis w.r.t $\prec$. For a complete description of the algorithm and a proof of correctness, we refer to [CLO15, pg. 90].

The Gröbner basis G of I w.r.t $\prec$ computed with Buchberger's algorithm is generally not reduced in the sense of Definition A.15. However, this can be ensured by replacing each $g \in G$ by the remainder obtained by dividing g by $G \setminus \{g\}$ w.r.t $\prec$, and scaling each of these polynomials by $\frac{1}{\mathrm{LC}_\prec(g)}$. Upon identifying the leading terms of this new basis, we obtain the (unique) *marked* Gröbner basis $G_\prec$ (cf. Definition A.16).

On the implementation level, the Generic walk (cf. Section 5) presents a new challenge, namely the reduction of polynomials with respect to a marked Gröbner basis G without a description of the monomial order with respect to which G is a Gröbner basis being given. This motivates the introduction of the MARKEDGB data type. These are sets consisting of pairs of polynomials and monomials of the form (g, x^α), where $\alpha \in \mathrm{supp}(g)$. We refer to x^α as the *marking* of g. (Intuitively, one can think of x^α as the initial monomial of g w.r.t some fixed "hidden" monomial order.) We say that a polynomial p is *reduced* with respect to a MARKEDGB when no term of p is divisible by any of the markings of the elements of G. A subroutine for the reduction of p by a MARKEDGB G is given on the next page. The output is exactly $\overline{p}^\prec$ (cf. definition A.14), where $\prec$ is a monomial order with respect to which G is a Gröbner basis.

Another computational task in the Generic walk is to transform a non-reduced Gröbner basis with markings to a reduced Gröbner basis (again, with respect to some unknown monomial order). This can be done by replacing each $g \in G$ in the marked basis with its normal form respect to $G \setminus \{g\}$. If G is inclusion minimal (which is the case for the "lifted" basis computed in the generic walk (cf. Corollary 2.11, Proposition 5.10) then the markings of output basis can be taken to be those of the input. In the `GroebnerWalk` package, the MARKEDGB data type and the algorithms described below can be found in the `markedGB.jl` file.

Algorithm 12 MARKEDGBNORMALFORM(G, p)

Input: G ▷ A set of marked polynomials of the form (g, x^α)

p ▷ a polynomial

Output: $\overline{p}^G$ ▷ A reduced polynomial such that $p - \overline{p}^G \in \langle G \rangle$

```
r ← 0

while p ≠ 0 do
    c_γ x^γ ← first(TERMS(p))                    ▷ iterate over terms of p
    DIVISIONOCCURRED ← False

    for (g, x^α) in G do
        if x^α divides x^γ then
            p ← p − x^(γ−α) · g                  ▷ if some x^α divides x^γ, use g to eliminate this term
            DIVISIONOCCURRED ← True
        end if
    end for

    if not DIVISIONOCCURRED then                 ▷ If no x^α divides x^γ
        r ← r + c_γ · x^γ                        ▷ push x^γ to the remainder
        p ← p − c_γ · x^γ
    end if
end while
return r
```

Algorithm 13 MARKEDGBREDUCE(G)

Input: G ▷ An inclusion minimal Gröbner basis with markings

Output: G ▷ The marked Gröbner basis G

```
for (g, x^α) ∈ G do
    G′ ← G \ {g}
    g ← (1/c_α) · MARKEDGBNORMALFORM(G′, g)      ▷ Replace g with its normal form and normalize
end for
return G                                         ▷ Return G, keeping the original markings
```

A.5 Homogeneous ideals

The ideals encountered at faces in the Gröbner fan are of a special kind: they are *homogeneous.*

Definition A.21. Let $\omega \in \mathbb{R}^n$.

1. A polynomial $f \in K[x_1, ..., x_n,]$ is said to be ω**-homogeneous** if $in_\omega(f) = f$.

2. An ideal $I \triangleleft K[x_1, .., x_n]$ is $\omega-$**homogeneous** if it has a generating set consisting of ω-homogeneous polynomials.

3. We call I **homogeneous** if it is ω-homogeneous for some $\omega \in \mathbb{Q}^n_{\geq 0}$.

Remark A.22. Let $I \lhd K[x_1, ..., x_n]$ be an ideal and $\omega \in \mathbb{Q}^n_{\geq 0}$ be a weight vector. The following holds:

- The initial ideal $in_\omega(I)$ is ω-homogeneous.
- Any f can be decomposed into a sum of ω-homogeneous components, grouped by ω-degree.

Example A.23. The ideal

$$I = \langle x_1 - t^5, x_2 - t^{12}, x_3 - t^{20} \rangle \lhd \mathbb{Q}[t, x_1, x_2, x_3] \tag{65}$$

defined in Example 6.6 is $\omega-$homogeneous with respect to the weight vector

$$\omega = \big(1200, 240, 100, 60\big) .$$

This is a special case of the general observation (63).

In Section 2, we need the following characterization of homogeneous ideals.

Theorem A.24. *Let I be an ideal, $\omega \in \mathbb{R}^n_{\geq 0}$. The following statements are equivalent:*

(i) I is ω-homogeneous.

(ii) For all $f \in I$, the summands of its ω-homogeneous decomposition all lie in I.

(iii) $in_\omega(I) = I$.

(iv) The reduced Gröbner basis of I w.r.t any monomial order $\prec$ consists of ω-homogeneous polynomials, i.e.

$$G_\prec = \{g_1, ..., g_s\} = \{in_\omega(g_1), ...in_\omega(g_s)\}.$$

Proof. Consult [CLO15, pg. 407]. □

B Matrix orderings and monomial order matrices

The results in this section describe the correspondence between monomial orders and matrices. We start by recalling the lexicographic term order on $\mathbb{Q}^n$.

Definition B.1. The **lexicographic order** on $\mathbb{Q}^n$ is the relation $<_{lex}$ defined as follows:

$$\text{For } \ u, v, \in \mathbb{Q}^n, \qquad u <_{lex} v \iff \bigvee_{i=1}^{k} \left(\bigwedge_{j=1}^{i-1} u_j = v_j \right) \wedge u_i < v_i.$$

Expressed in words, $<_{lex}$ compares entries of u and v w.r.t the standard ordering on $\mathbb{Q}$ until a tie is broken.

Lemma B.2. *$<_{lex}$ is a strict total order on $\mathbb{Q}^n$*

Proof. It is almost trivial to check that $<_{lex}$ properties $(i) - (iv)$ of a strict total order:

(i) *irreflexive:* $\neg(u <_{lex} u)$.

(ii) *asymmetric:* $u <_{lex} v \implies \neg(v <_{lex} u)$.

(iii) *transitive:* $u <_{lex} v$ and $v <_{lex} w \implies v <_{lex} w$.

(iv) *connected:* $u \neq v \implies u <_{lex} v$ or $v <_{lex} u$.

For all $u, v \in \mathbb{Q}^n$. □

Via $<_{lex}$, any matrix $A \in \mathbb{Q}^{k \times n}$ defines a partial order on $\mathbb{Q}^n$.

Definition B.3. For a matrix $A \in \mathbb{Q}^{k\times n}$ the **matrix order** defined by A is the relation $<_A$ on $\mathbb{Q}^n$ defined as follows:

$$\begin{aligned} \text{For} \quad u,v, \in \mathbb{Q}^n, \qquad u <_A v &\iff Au <_{lex} Av \\ &\iff (\langle a_1,u\rangle, \langle a_2,u\rangle, ..., \langle a_k,u\rangle) <_{lex} (\langle a_1,v\rangle, \langle a_2,v\rangle, ..., \langle a_k,v\rangle) \\ &\iff \bigvee_{i=1}^{k} \left(\bigwedge_{j=1}^{i-1} \langle a_i,u\rangle = \langle a_j,v\rangle \right) \wedge \langle a_i,u\rangle < \langle a_i,v\rangle. \end{aligned}$$

where $<_{lex}$ is the lexicographic order on $\mathbb{Q}^k$, and a_i is the $i-$th column of A.

Lemma B.4. *If $A \in \mathbb{Q}^{k\times n}$ is a matrix with $k \geq n$ and $rk(A) = n$, then $<_A$ is a strict total order on $\mathbb{Q}^n$.*

Proof. The first three properties of a strict total order follow directly from the fact that $<_{lex}$ is a strict total order. Connectedness follows from $rk(A) = n$; as the columns of A are linearly independent, $u \neq v$ implies $Au \neq Av$, and therefore (by connectedness of $<_{lex}$), $Au <_{lex} Av$ or $Av <_{lex} Au$ holds. □

If we further require that the first row a_1 of A be non-negative, then $<_A$ defines a monomial order:

Proposition B.5. *Let $k \geq n$ and $A \in \mathbb{Q}^{k,n}$ be a matrix of rank n such that its first row $a_1 \in \mathbb{Q}^n$ is not the zero vector and has non-negative entries. The relation $\prec_A$ on the set of all monomials of $K[x_1, ..., x_n]$ defined by*

$$x^\beta \prec_A x^\alpha \quad :\iff \quad \beta <_A \alpha \qquad \textit{for all} \;\; \alpha, \beta \in \mathbb{N}^n.$$

is a monomial order.

Proof. We prove that $\prec_A$ fulfills the conditions $(i) - (iii)$ from Definition A.5.
(i): As $<_A$ is a strict total order on $\mathbb{Q}^n$, so is its restriction on $\mathbb{N}^n$. Thus, by definition, so is $\prec_A$.
(ii): Admissibility follows directly from the the distributive law for matrix-vector multiplication: For any $\alpha, \beta, \gamma \in \mathbb{N}^n$

$$A(\beta + \gamma) <_{lex} A(\alpha + \gamma) \iff A\beta <_{lex} A\alpha.$$

(iii): As (i) and (ii) hold, we prove (iii) using the condition in Proposition A.7. For any $\alpha \in \mathbb{N}^n \setminus \{0\}$, we have that

$$\langle a_1, \alpha\rangle > 0 = \langle a_1, 0\rangle \implies x^0 = 1 \prec x^\alpha$$

as desired. □

Thus, specific kinds of matrices define monomial orders. Conversely, any monomial order may be represented by a matrix in the following sense:

Proposition B.6. *Let $\prec$ be a monomial order. Then there exists a matrix $A \in \mathbb{Q}^{m\times n}$ such that for all monomials x^α, $x^\beta \in K[x_1, ..., x_n]$:*

$$x^\alpha \prec x^\beta \iff x^\alpha \prec_A x^\beta.$$

Proof. Consult [Ovc02, pg.239]. □

C Cones and fans

In this section, we provide a refresher of the notions from polyhedral geometry which appear in this text. For more details, we refer to Chapter 1 of [Zie95].

Definition C.1. Let $u \in \mathbb{R}^n \setminus \{0\}$ and $d \in \mathbb{R}_{>0}$.

- The **hyperplane** in $\mathbb{R}^n$ with normal vector u is

$$H_u := \{ x \in \mathbb{R}^n \ : \ \langle x, u \rangle = 0 \ \}.$$

- The $n-$ dimensional (closed, positive) **half-space** defined by u is

$$H_u^+ := \{ x \in \mathbb{R}^n \ : \ \langle x, u \rangle \geq 0 \ \}.$$

Definition C.2. For $m \in \mathbb{N}$, let $V = \{v_1, ..., v_m\} \subset \mathbb{R}^n$. The (polyhedral) **cone** over V is the set of all non-negative linear combinations of elements of V. We write

$$cone(V) := \{ x \in \mathbb{R}^n \ : \ x = \sum_{i=1}^{m} \lambda_i v_i \ \text{ for some } \lambda_1, ..., \lambda_m \in \mathbb{R}_{\geq 0} \} \subset \mathbb{R}^n. \tag{66}$$

The **dimension** of $cone(V)$ is the dimension of the span of V as a vector space. The vectors $v_1, ..., v_m$ are often referred to as the *rays* of the cone.

Remark C.3. $cone(V)$ is a convex set.

The following theorem provides an important equivalent characterization.

Theorem C.4 ("Main" theorem for cones)**.** *Let $C \subset \mathbb{R}^n$.*
C is a cone in the sense of Definition C.2 if and only if there exists a finite collection of half-spaces H_{u_1}, ..., H_{u_k} such that

$$C = H_{u_1}^+ \cap H_{u_1}^+ \cap H_{u_2}^+ \cdots \cap H_{u_k}^+. \tag{67}$$

Proof. Consult [Zie95, pg. 30]. □

For a polyhedral cone C, we refer to a representation of C as in (66) as a $V-$ **description**, and a representation as in (67) as an H**-description**. Faces of cones arise by intersecting with hyperplanes.

Definition C.5. Let $C := cone(V)$ be a polyhedral cone in $\mathbb{R}^n$.

- A **supporting hyperplane** to C is a hyperplane H_u with $u \in \mathbb{R}^n \setminus \{0\}$ such that

$$C = C \cap H_u^+ \qquad \text{(or equivalently, } \langle u, x \rangle \geq 0 \text{ for all } x \in C).$$

- A (proper) **face** of C is a set F of the form

$$F = C \cap H_u \qquad \text{, where } H_u \text{ is a supporting hyperplane of } C.$$

- The **dimension** of a face F is the dimension of F as a cone (cf. remark C.7).
- Faces of C of dimension $\dim(C) - 1$ are called **facets**.

Example C.6. Let

$$V = \left\{ \begin{pmatrix} 2 \\ 1 \\ 0 \end{pmatrix} \ , \begin{pmatrix} 1 \\ 1 \\ 1 \end{pmatrix} \ , \ \begin{pmatrix} 0 \\ 2 \\ 1 \end{pmatrix} \right\} =: \{v_1 \ , \ v_2 \ , \ v_3\} \subset \mathbb{R}^3.$$

The cone over V $cone(V)$ is depicted below. Its $H-$description is

$$cone(V) = H^+_{u_1} \cap H^+_{u_2} \cap H^+_{u_3},$$

where

$$u_1 = \begin{pmatrix} 1 \\ -2 \\ 4 \end{pmatrix} \ , \ u_2 = \begin{pmatrix} -1 \\ 2 \\ -1 \end{pmatrix} \ , \ u_3 = \begin{pmatrix} 1 \\ 1 \\ -2 \end{pmatrix}.$$

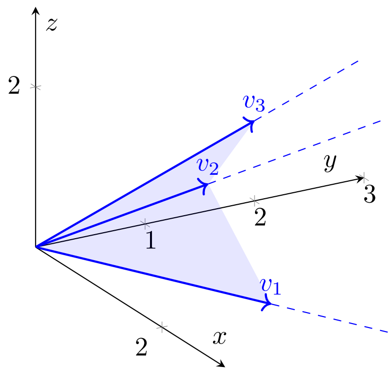


Figure 8: The three-dimensional cone over V

The face spanned by v_2 and v_3 is the intersection of $cone(V)$ with H_{u_3}.

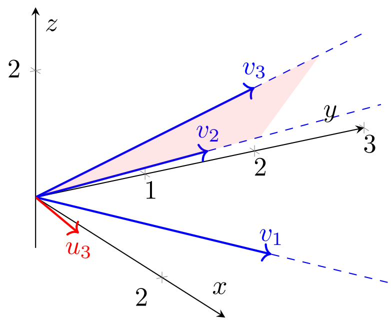


Figure 9: The facet highlighted in red has the normal vector u_3.

Remark C.7. Faces of cones are cones. This is a consequence of Theorem C.4.

Remark C.8. Faces of a face F of C are themselves faces of C. Because of this, the faces of C, together with the set itself and the empty set, form a *lattice* with respect to the inclusion relation (cf. [Zie95, pg. 55]).

The cone in example C.6 belongs to a special class: it is a *pointed cone*.

Definition C.9. Let $C \subset \mathbb{R}^n$ be a polyhedral cone.

- The **lineality space** $\mathcal{L}$ of C is the largest linear subspace of $\mathbb{R}^n$ such that $\mathcal{L} \subset C$.
- C is **pointed** if its lineality space is trivial (i.e. $\mathcal{L} = \{0\}$).

Example C.10. The two-dimensional cone

$$C = cone(\{ \begin{pmatrix} 1 \\ 0 \\ 0 \end{pmatrix} , \begin{pmatrix} -1 \\ 0 \\ 0 \end{pmatrix} , \begin{pmatrix} 0 \\ 1 \\ 0 \end{pmatrix} \}) \subset \mathbb{R}^3 \tag{68}$$

is not pointed, as $\mathcal{L} = span(e_1) \subset C$.

Definition C.11. A polyhedral **fan** in $\mathbb{R}^n$ is a finite collection of non-empty polyhedral cones

$$\mathcal{F} = \{C_1, ..., C_m\}.$$

with the following two properties:

(i) Every non-empty face of a cone in $\mathcal{F}$ is in $\mathcal{F}$.

(ii) The intersection of any two cones is a face of both.

We say that $\mathcal{F}$ is **complete** if

$$\mathbb{R}^n = \bigcup_{C \in \mathcal{F}} C,$$

whereas $\mathcal{F}$ is said to be **pure** if all of its maximal cones are of the same dimension. $\mathcal{F}$ is **pointed** if all of its maximal cones are pointed.

Remark C.12. In contrast to [Zie95], we do not assume polyhedral fans to be complete . This is because Gröbner fans generally do not have this property. However, Gröbner fans are pure: the dimension of any maximal cone is n (the number of variables of the polynomial ring).

Remark C.13. Any polyhedral fan $\mathcal{F}$ is uniquely determined by its set of maximal cones. All other cones in the fan are obtained by taking faces.

Example C.14. Let

$$v_1 = \begin{pmatrix} 2 \\ 1 \\ 0 \end{pmatrix} , v_2 = \begin{pmatrix} 1 \\ 1 \\ 1 \end{pmatrix} , v_3 = \begin{pmatrix} 0 \\ 2 \\ 1 \end{pmatrix} , v_4 = \begin{pmatrix} 0 \\ 1 \\ 2 \end{pmatrix} , v_5 = \begin{pmatrix} 2 \\ -1 \\ 0 \end{pmatrix} .$$

The cone $\mathcal{F}$ depicted in fig. 10 is uniquely determined by the maximal cones

$$C_1 = cone(\{v_1, v_2, v_3\} , C_2 = cone(\{v_3, v_3\}) , C_3 = cone(\{v_5\}).$$

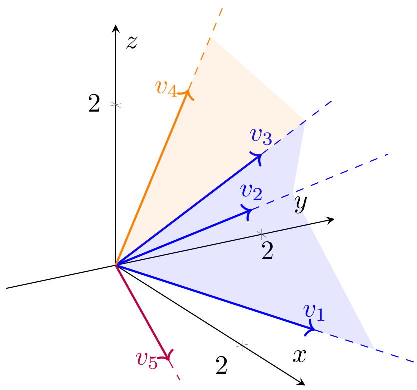


Figure 10: The fan $\mathcal{F}$ from example C.14.

It is a pointed, impure fan in $\mathbb{R}^3$.

D Overview of polynomial systems

We provide references and additional information for the polynomial ideals encountered in this text.

System	Characteristics						
	Reference	no. variables	max degree	$\lvert G_{\prec}\rvert$	$\lvert G_{\prec'}\rvert$	$\dim(I)$	misc.
`chap3`	(35)	3	2	3	4	0	$(1,1,1)$-homogeneous
`chap4`	(40)	3	5	2	4	1	poly's with up to 65 terms
`cyclic5`	[BF91]	5	5	5	11	0	-
`cyclic6`	[BF91]	6	6	6	17	0	-
`katsura6`	[SK89]	7	7	7	7	0	coeff's of order up to 10^{65}
`newellp1`	[Tra04]	5	6	3	39	2	poly's with up to 982 terms
`newellp4`	[Tra04]	5	6	3	13	2	poly's with up to 132 terms
`newellp25`	[Tra04]	5	6	3	56	2	poly's with up to 1762 terms
`agk4`	[AGK97]	5	4	3	29	2	poly's with up to 388 terms
`agk8`	[AGK97]	5	4	3	34	2	poly's with up to 539 terms
`randomknap`	[Now24]	6	789*	5	85*	0	toric, homogeneous
`toric`	[Now24]	7	643*	341*	297*	0	toric, coeff's are all 1

Table 9: Descriptions of the polynomial systems in this text.

The entry “max degree” is the maximal total degree of a polynomial in the starting basis, whereas $\dim(I)$ refers to the degree of the corresponding algebraic variety. The ideals `randomknap` and `toric` were obtained by running the scripts `random_knap.m2` and `random_toric.m2` from the `modifiedGWalk` repository once; the values marked by an asterisk will vary each time this script is run.

The ideals in the table are grouped by application. The first group correspond to lexicographic conversion. Here, the starting order is *grevlex* and the target order is *lex*. The second group correspond to implicitization of 2-dimensional surfaces in $\mathbb{R}^3$. In accordance with [Tra04], we choose to represent the starting and target monomial orders by the following two matrices:

$$A_{\prec} = \begin{pmatrix} 1 & 1 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 & 1 \\ 0 & 0 & 0 & 1 & 0 \\ 1 & 1 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 \end{pmatrix} \quad \text{and} \quad A_{\prec'} = \begin{pmatrix} 0 & 0 & 0 & 1 & 1 \\ 1 & 1 & 1 & 0 & 0 \\ 1 & 1 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 \end{pmatrix}, \tag{69}$$

where the variables are in the order x, y, z, u, v. This allows for the possibility of a computation of an elimination term order in fewer steps compared to, e.g. the monomial order *grevlex* refined by the vector $(0, 0, 0, 1, 1)$. We refer to [Tra04, pg.842] for more details on why this is the case.

Finally, the last two ideals are toric ideals, which were generated randomly according to Section 6.3. The starting and target orders of the conversions are *grevlex* and *glex* respectively.

References

[24] *OSCAR – Open Source Computer Algebra Research system, Version 1.0.2.* The OSCAR Team, 2024. URL: https://www.oscar-system.org.

[AGK97] Beatrice Amrhein, Oliver Gloor, and Wolfgang Küchlin. "On the walk". en. In: *Theoretical Computer Science* 187.1-2 (Nov. 1997), pp. 179–202. ISSN: 03043975. DOI: 10.1016/S0304-3975(97)00064-9. URL: https://linkinghub.elsevier.com/retrieve/pii/S0304397597000649 (visited on 09/01/2023).

[AL04] Karen Aardal and Arjen K. Lenstra. "Hard Equality Constrained Integer Knapsacks". en. In: *Mathematics of Operations Research* 29.3 (Aug. 2004), pp. 724–738. ISSN: 0364-765X, 1526-5471. DOI: 10.1287/moor.1040.0099. URL: https://pubsonline.informs.org/doi/10.1287/moor.1040.0099 (visited on 06/10/2024).

[BF91] Göran Björck and Ralf Fröberg. "A faster way to count the solutions of inhomogeneous systems of algebraic equations, with applications to cyclic n-roots". In: *Journal of Symbolic Computation* 12.3 (Sept. 1991), pp. 329–336. ISSN: 0747-7171. DOI: 10.1016/S0747-7171(08)80153-8. URL: https://www.sciencedirect.com/science/article/pii/S0747717108801538 (visited on 06/10/2024).

[BT18] Paul Breiding and Sascha Timme. *HomotopyContinuation.jl: A package for homotopy continuation in Julia.* arXiv:1711.10911 [cs, math]. May 2018. DOI: 10.48550/arXiv.1711.10911. URL: http://arxiv.org/abs/1711.10911 (visited on 06/10/2024).

[Buc06] Bruno Buchberger. "Bruno Buchberger's PhD thesis 1965: An algorithm for finding the basis elements of the residue class ring of a zero dimensional polynomial ideal". In: *Journal of Symbolic Computation.* Logic, Mathematics and Computer Science: Interactions in honor of Bruno Buchberger (60th birthday) 41.3 (Mar. 2006), pp. 475–511. ISSN: 0747-7171. DOI: 10.1016/j.jsc.2005.09.007. URL: https://www.sciencedirect.com/science/article/pii/S0747717105001483 (visited on 06/08/2024).

[CKM97] S. Collart, M. Kalkbrener, and D. Mall. "Converting Bases with the Gröbner Walk". en. In: *Journal of Symbolic Computation* 24.3-4 (Sept. 1997), pp. 465–469. ISSN: 07477171. DOI: 10.1006/jsco.1996.0145. URL: https://linkinghub.elsevier.com/retrieve/pii/S0747717196901455 (visited on 09/01/2023).

[CLO05] David A. Cox, John Little, and Donal O'Shea. *Using Algebraic Geometry.* en. Vol. 185. Graduate Texts in Mathematics. New York: Springer-Verlag, 2005. ISBN: 9780387207063. DOI: 10.1007/b138611. URL: http://link.springer.com/10.1007/b138611 (visited on 11/07/2023).

[CLO15] David A. Cox, John Little, and Donal O'Shea. *Ideals, Varieties, and Algorithms: An Introduction to Computational Algebraic Geometry and Commutative Algebra.* en. Undergraduate Texts in Mathematics. Cham: Springer International Publishing, 2015. ISBN: 9783319167206. DOI: 10.1007/978-3-319-16721-3. URL: https://link.springer.com/10.1007/978-3-319-16721-3 (visited on 12/11/2023).

[CT91] Pasqualina Conti and Carlo Traverso. "Buchberger algorithm and integer programming". en. In: *Applied Algebra, Algebraic Algorithms and Error-Correcting Codes.* Ed. by Harold F. Mattson, Teo Mora, and T. R. N. Rao. Berlin, Heidelberg: Springer, 1991, pp. 130–139. ISBN: 9783540384366. DOI: 10.1007/3-540-54522-0_102.

[Dub90] Thomas W. Dubé. "The Structure of Polynomial Ideals and Gröbner Bases". en. In: *SIAM Journal on Computing* 19.4 (Aug. 1990), pp. 750–773. ISSN: 0097-5397, 1095-7111. DOI: 10.1137/0219053. URL: http://epubs.siam.org/doi/10.1137/0219053 (visited on 01/29/2024).

[Fau+93] J.C. Faugère et al. "Efficient Computation of Zero-dimensional Gröbner Bases by Change of Ordering". en. In: *Journal of Symbolic Computation* 16.4 (Oct. 1993), pp. 329–344. ISSN: 07477171. DOI: 10.1006/jsco.1993.1051. URL: https://linkinghub.elsevier.com/retrieve/pii/S0747717183710515 (visited on 04/17/2024).

[Fau99] Jean-Charles Faugére. "A new efficient algorithm for computing Gröbner bases (F4)". en. In: *Journal of Pure and Applied Algebra* 139.1-3 (June 1999), pp. 61–88. ISSN: 00224049. DOI: 10.1016/S0022-4049(99)00005-5. URL: https://linkinghub.elsevier.com/retrieve/pii/S0022404999000055 (visited on 04/17/2024).

[FJT07] Komei Fukuda, Anders N. Jensen, and Rekha R. Thomas. "Computing Gröbner fans". In: *Mathematics of Computation* 76.260 (Oct. 2007), pp. 2189–2213. ISSN: 00255718. DOI: 10.1090/S0025-5718-07-01986-2. URL: http://www.ams.org/journal-getitem?pii=S0025-5718-07-01986-2 (visited on 01/29/2024).

[Fuk+07] K. Fukuda et al. "The generic Gröbner walk". In: *Journal of Symbolic Computation* 42.3 (Mar. 2007), pp. 298–312. ISSN: 0747-7171. DOI: 10.1016/j.jsc.2006.09.004. URL: https://www.sciencedirect.com/science/article/pii/S0747717106001003 (visited on 09/01/2023).

[HDB17] Amir Hashemi, Mahdi Dehghani Darmian, and Marzieh Barkhordar. "Gröbner Systems Conversion". en. In: *Mathematics in Computer Science* 11.1 (Mar. 2017), pp. 61–77. ISSN: 1661-8289. DOI: 10.1007/s11786-017-0295-3. URL: https://doi.org/10.1007/s11786-017-0295-3 (visited on 10/20/2023).

[Jen07] Anders Nedergaard Jensen. *Algorithmic Aspects of Gröbner Fans and Tropical Varieties.* Århus: Department of Mathematical Sciences , University of Aarhus, 2007.

[MR88] Teo Mora and Lorenzo Robbiano. "The Gröbner fan of an ideal". In: *Journal of Symbolic Computation* 6.2 (Oct. 1988), pp. 183–208. ISSN: 0747-7171. DOI: 10.1016/S0747-7171(88)80042-7. URL: https://www.sciencedirect.com/science/article/pii/S0747717188800427 (visited on 03/27/2024).

[Now24] Francesco Nowell. *modifiedGWalk*. July 2024. DOI: 10.5281/zenodo.12670965. URL: https://zenodo.org/records/12670965 (visited on 07/06/2024).

[Ovc02] Alexey Ovchinnikov. "Classification and Applications of Monomial Orderings and the Properties of Differential Term-Ordering". In: *Proceedings of CASC 2002* (2002), pp. 237–252.

[Rob85] Lorenzo Robbiano. "Term orderings on the polynomial ring". en. In: *EUROCAL '85*. Ed. by Bob F. Caviness. Lecture Notes in Computer Science. Berlin, Heidelberg: Springer, 1985, pp. 513–517. ISBN: 9783540396857. DOI: 10.1007/3-540-15984-3_321.

[RS12] Bjarke Hammersholt Roune and Michael Stillman. "Practical Gröbner basis computation". en. In: *Proceedings of the 37th International Symposium on Symbolic and Algebraic Computation* (July 2012), pp. 203–210. DOI: 10.1145/2442829.2442860. URL: https://dl.acm.org/doi/10.1145/2442829.2442860 (visited on 06/14/2024).

[Sim14] Bruno Simões. "An Hilbert-Driven Strategy for Signature-Based Gröbner Basis Algorithms". en. In: *Future Vision and Trends on Shapes, Geometry and Algebra*. Ed. by Raffaele De Amicis and Giuseppe Conti. Vol. 84. London: Springer London, 2014, pp. 13–37. DOI: 10.1007/978-1-4471-6461-6_2.

[SK89] Mitsuhiro Sasaki and Shigetoshi Katsura. "The asymmetric continuous distribution function of the effective field of the Ising model in the spin glass and the ferromagnetic states on the Bethe lattice". In: *Physica A: Statistical Mechanics and its Applications* 157.3 (June 1989), pp. 1195–1202. ISSN: 0378-4371. DOI: 10.1016/0378-4371(89)90039-3. URL: https://www.sciencedirect.com/science/article/pii/0378437189900393 (visited on 06/10/2024).

[Stu95] Bernd Sturmfels. "Gröbner Bases and Convex Polytopes". en. In: vol. 8. University Lecture Series. Providence, Rhode Island: American Mathematical Society, Dec. 1995. ISBN: 9780821804872. DOI: 10.1090/ulect/008. URL: http://www.ams.org/ulect/008 (visited on 09/22/2023).

[Tra00] Quoc-Nam Tran. "A Fast Algorithm for Gröbner Basis Conversion and its Applications". In: *Journal of Symbolic Computation* 30.4 (Oct. 2000), pp. 451–467. ISSN: 0747-7171. DOI: 10.1006/jsco.1999.0416. URL: https://www.sciencedirect.com/science/article/pii/S0747717199904169 (visited on 09/26/2023).

[Tra04] Quoc-Nam Tran. "Efficient Groebner walk conversion for implicitization of geometric objects". In: *Computer Aided Geometric Design* 21.9 (Nov. 2004), pp. 837–857. ISSN: 0167-8396. DOI: 10.1016/j.cagd.2004.07.001. URL: https://www.sciencedirect.com/science/article/pii/S0167839604000755 (visited on 06/10/2024).

[Zie95] Günter M. Ziegler. *Lectures on Polytopes*. Vol. 152. Graduate Texts in Mathematics. New York, NY: Springer New York, 1995. ISBN: 9780387943657. DOI: 10.1007/978-1-4613-8431-1. URL: http://link.springer.com/10.1007/978-1-4613-8431-1 (visited on 06/24/2024).