The Project-And-Lift Algorithm for the Computation of Toric Gröbner Bases

An Implementation in Mathematica

Stephan Ritscher

Department of Computer Science
TU München

13th of April 2010
Outline

1 Motivation

2 Prerequisites

3 Problem Statement

4 Overview of Algorithms
   - Common Parts
   - Saturation Algorithm
   - Shadow Algorithm
   - Project-and-Lift Algorithm
   - Geometric Buchberger Algorithm

5 Experimental Results

6 Conclusion
Integer Linear Programming

Standard Form

\[
\min \{ cy : Ay = b, \ y \in \mathbb{N}^n \}
\]

Selected Applications

- Travelling Salesman
- Knapsack, Bin Packing
- Schedule Optimization
- Frequency Planning for Mobile Phone Networks
- Capacity Planning for Telecommunication Networks
Applying Gröbner-Bases

Definition (Test Set)

$T$ is a test set for $\min\{cy : Ay = b, y \in \mathbb{N}^n\}$ if

1. all $y \in T$ have negative cost ($cy < 0$),
2. all $y \in T$ solve $Ay = 0$ and
3. for each non-optimal solution $y_s \in \mathbb{N}^n$ of $Ay_s = b$ there is a $y \in T$ s.t. $y_s + y \in \mathbb{N}^n$. 
Applying Gröbner-Bases

Definition (Test Set)

$T$ is a test set for $\min\{cy : Ay = b, y \in \mathbb{N}^n\}$ if

1. all $y \in T$ have negative cost ($cy < 0$),
2. all $y \in T$ solve $Ay = 0$ and
3. for each non-optimal solution $y_s \in \mathbb{N}^n$ of $Ay_s = b$ there is a $y \in T$ s.t. $y_s + y \in \mathbb{N}^n$.

Optimization Algorithm

1. Find any solution $y_s \in \mathbb{N}^n$ of $Ay_s = b$.
2. While $\exists y \in T : y_s + y \in \mathbb{N}^n$, let $y_s \leftarrow y_s + y$.
3. Return $y_s$. 
Applying Gröbner-Bases

Definition (Test Set)

$T$ is a test set for $\min\{ cy : Ay = b, y \in \mathbb{N}^n \}$ if

1. all $y \in T$ have negative cost ($cy < 0$),
2. all $y \in T$ solve $Ay = 0$ and
3. for each non-optimal solution $y_s \in \mathbb{N}^n$ of $Ay_s = b$ there is a $y \in T$ s.t. $y_s + y \in \mathbb{N}^n$.

Optimization Algorithm

1. Find any solution $y_s \in \mathbb{N}^n$ of $Ay_s = b$.
2. While $\exists y \in T : y_s + y \in \mathbb{N}^n$, let $y_s \leftarrow y_s + y$.
3. Return $y_s$.

How can we find a finite (small) test set???
Polynomial Ideals

Notation

- \( \mathbb{K}[X] = \text{ring of polynomials over } \mathbb{K} \text{ in variables } X = \{x_1, \ldots, x_n\} \)
- \( I \subset \mathbb{K}[X] \) is an ideal iff for all \( a, b \in I \), \( r \in \mathbb{K}[X] \)
  1. \( a + b \in I \) and
  2. \( ar \in I \).
- \( \langle f_1, \ldots, f_s \rangle = \text{ideal generated by } f_1, \ldots, f_s \in \mathbb{K}[X] \)
Polynomial Ideals

Notation

- $\mathbb{K}[X]$ = ring of polynomials over $\mathbb{K}$ in variables $X = \{x_1, \ldots, x_n\}$
- $l \subset \mathbb{K}[X]$ is an ideal iff for all $a, b \in l, r \in \mathbb{K}[X]$
  1. $a + b \in l$ and
  2. $ar \in l$.
- $\langle f_1, \ldots, f_s \rangle$ = ideal generated by $f_1, \ldots, f_s \in \mathbb{K}[X]$

Example

- $\mathbb{Q}[x_1, x_2, x_3] \ni x_1^2 - x_2x_3, x_1x_2^2x_3 - 1, x_2^5x_3^3 - 1$
- $\langle x_1^2 - x_2x_3, x_1x_2^2x_3 - 1, x_2^5x_3^3 - 1 \rangle \ni x_1 - x_2^3x_3^2 = x_2^2x_3(x_1^2 - x_2x_3) - x_1(x_1x_2^2x_3 - 1)$
Monomial Orderings

Definition
A total ordering $\prec$ of the monomials $x^\alpha = x_1^{\alpha_1} \cdots x_n^{\alpha_n}$ is admissible iff for all $\alpha, \beta, \gamma \in \mathbb{N}^n$ ($0 \in \mathbb{N}$)

1. $x^\alpha \prec x^\beta \Rightarrow x^\alpha x^\gamma \prec x^\beta x^\gamma$ and
2. $1 \prec x^\alpha$ for $x^\alpha \neq 1$.

$\text{LM}(f) =$ largest monomial of $f \in \mathbb{K}[X]$ wrt. $\prec$
Monomial Orderings (2)

Example

- **Lexicographic Ordering:**
  \( x^\alpha \prec x^\beta \) iff first nonzero entry of \( \alpha - \beta \) is negative.

  \[ x_1 \succ x_2^3 x_3^2 \succ x_2^2 x_3^3 \]

- **Graded Reverse Lexicographic Ordering:**
  \( x^\alpha \prec x^\beta \) iff \( \deg(x^\alpha) < \deg(x^\beta) \) or \( \deg(x^\alpha) = \deg(x^\beta) \) and last nonzero entry of \( \alpha - \beta \) is positive.

  \[ x_2^3 x_3^2 \succ x_2^2 x_3^3 \succ x_1 \]
Monomial Orderings (3)

Definition (Matrix Orderings)

Given a matrix $C \in \mathbb{K}^{s,n}$, let $x^\alpha \prec x^\beta$ iff the first nonzero entry of $C\alpha - C\beta$ is positive.
Monomial Orderings (3)

Definition (Matrix Orderings)
Given a matrix $C \in \mathbb{K}^{s,n}$, let $x^\alpha \prec x^\beta$ iff the first nonzero entry of $C^\alpha - C^\beta$ is positive.

Notes
Any admissible monomial ordering can be
- represented by a matrix $C \in \mathbb{R}^{n,n}$.
- approximated up to an arbitrary, fixed degree by a matrix $C \in \mathbb{Z}^{n,n}$.
Common monomial orderings can be represented by a matrix $C \in \mathbb{Z}^{n,n}$.
Monomial Orderings (3)

Definition (Matrix Orderings)
Given a matrix $C \in \mathbb{K}^{s,n}$, let $x^\alpha \prec x^\beta$ iff the first nonzero entry of $C\alpha - C\beta$ is positive.

Notes
Any admissible monomial ordering can be
- represented by a matrix $C \in \mathbb{R}^{n,n}$.
- approximated up to an arbitrary, fixed degree by a matrix $C \in \mathbb{Z}^{n,n}$.

Common monomial orderings can be represented by a matrix $C \in \mathbb{Z}^{n,n}$

Example ($n = 3$)
Lexicographic Ordering: \[
\begin{pmatrix}
1 & 0 & 0 \\
0 & 1 & 0 \\
0 & 0 & 1 \\
\end{pmatrix}
\]
Graded Reverse-lexicographic Ordering: \[
\begin{pmatrix}
1 & 1 & 1 \\
0 & 0 & -1 \\
0 & -1 & 0 \\
\end{pmatrix}
\]
Gröbner Bases

Definition

$G$ is a Gröbner basis of $I$ wrt. a monomial ordering $\prec$ iff

1. $I = \langle G \rangle$ and
2. $\langle \text{LM}(I) \rangle = \langle \text{LM}(G) \rangle$. 

Example
Consider $I = \langle x_2^3 - x_2 x_3^2, x_1 x_2^2 x_3 - 1, x_5^2 x_3^3 - 1 \rangle$ and the lexicographic monomial ordering.

• $G_1 = \{ x_2^3 - x_2 x_3^2, x_1 x_2^2 x_3 - 1, x_5^2 x_3^3 - 1 \}$ is no Gröbner basis of $I$ since $x_1 \in \langle \text{LM}(I) \rangle$ but $x_1 \not\in \langle \text{LM}(G_1) \rangle$.

• $G_2 = \{ x_2^3 - x_2 x_3^2, x_1 x_2^2 x_3 - 1, x_5^2 x_3^3 - 1, x_1 - x_3^2 x_2^3 \}$ is a Gröbner basis of $I$. 

Gröbner Bases

Definition

$G$ is a **Gröbner basis** of $I$ wrt. a monomial ordering $\prec$ iff

1. $I = \langle G \rangle$ and
2. $\langle \text{LM}(I) \rangle = \langle \text{LM}(G) \rangle$.

Example

Consider $I = \langle x_1^2 - x_2 x_3, x_1 x_2^2 x_3 - 1, x_2^5 x_3^3 - 1 \rangle$ and the lexicographic monomial ordering.

- $G_1 = \{ x_1^2 - x_2 x_3, x_1 x_2^2 x_3 - 1, x_2^5 x_3^3 - 1 \}$ is no Gröbner basis of $I$ since $x_1 \in \langle \text{LM}(I) \rangle$ but $x_1 \notin \langle \text{LM}(G_1) \rangle$.
- $G_2 = \{ x_1^2 - x_2 x_3, x_1 x_2^2 x_3 - 1, x_2^5 x_3^3 - 1, x_1 - x_2^3 x_3^2 \}$ is a Gröbner basis of $I$. 
Problem Statement

Definition (Toric Ideals)

Given a matrix $A \in \mathbb{Z}^{k,n}$, the associated toric ideal is defined as

$$I(A) = \langle x^\alpha - x^\beta : \alpha, \beta \in \mathbb{N}^n : \alpha - \beta \in \ker(A) \rangle$$
Problem Statement

Definition (Toric Ideals)
Given a matrix $A \in \mathbb{Z}^{k,n}$, the associated toric ideal is defined as

$$I(A) = \langle x^\alpha - x^\beta : \alpha, \beta \in \mathbb{N}^n : \alpha - \beta \in \ker(A) \rangle$$

Example
$A = \begin{pmatrix} 1 & -3 & 5 \end{pmatrix}$

$\Rightarrow I(A) = \langle x_2^5x_3^3 - 1, x_1 - x_2^3x_3^2 \rangle$
Problem Statement

Definition (Toric Ideals)
Given a matrix $A \in \mathbb{Z}^{k,n}$, the associated toric ideal is defined as

$$I(A) = \langle x^\alpha - x^\beta : \alpha, \beta \in \mathbb{N}^n : \alpha - \beta \in \ker(A) \rangle$$

Example

$A = (1 \ -3 \ 5)$

$\Rightarrow I(A) = \langle x_2^5x_3^3 - 1, x_1 - x_2x_3^2 \rangle$

Task
Given

- matrix $A \in \mathbb{Z}^{k,n}$ and
- monomial ordering $\prec$ (as matrix)
compute a Gröbner basis of $I(A)$ wrt. $\prec$. 
”Blue Print” of an Toric Ideal Algorithm

**Input**: Matrix $A$, monomial ordering $\prec$ defined by matrix $C$

**Output**: Gröbner basis of $I(A)$

Calculate lattice basis $B$ of $\ker_\mathbb{Z}(A)$

Compute Markov basis $M$ of $\ker_\mathbb{Z}(A)$ resp. ideal basis $F$ of $I(A)$

Compute Gröbner basis of $I(A)$
Lattice Basis

Definition

A lattice $L$ is a set of the form

$$L = \mathbb{Z}v_1 + \mathbb{Z}v_2 + \ldots + \mathbb{Z}v_s \subset \mathbb{Z}^n.$$  

The set $\{v_1, \ldots, v_s\}$ is a lattice basis of $L$. 

Example

$$A = \begin{pmatrix} 1 & -3 & 5 \end{pmatrix} \Rightarrow \ker \mathbb{Z}(A) = \mathbb{Z}(2-1-1) + \mathbb{Z}(1 2 1)^T$$

Computation

1. Triangulate $A$ with unimodular operations (Hermite decomposition).
2. Lattice basis $B$ of $L = \ker \mathbb{Z}(A)$ can be read off the triangular form.
Lattice Basis

Definition
A lattice $L$ is a set of the form

$$L = \mathbb{Z}v_1 + \mathbb{Z}v_2 + \ldots + \mathbb{Z}v_s \subset \mathbb{Z}^n.$$  

The set $\{v_1, \ldots, v_s\}$ is a lattice basis of $L$.

Example
$A = \begin{pmatrix} 1 & -3 & 5 \end{pmatrix}$

$$\Rightarrow \ker_{\mathbb{Z}}(A) = \mathbb{Z} \begin{pmatrix} 2 & -1 & -1 \end{pmatrix}^T + \mathbb{Z} \begin{pmatrix} 1 & 2 & 1 \end{pmatrix}^T$$
Lattice Basis

Definition

A lattice $L$ is a set of the form

$$L = \mathbb{Z}v_1 + \mathbb{Z}v_2 + \ldots + \mathbb{Z}v_s \subset \mathbb{Z}^n.$$  

The set $\{v_1, \ldots, v_s\}$ is a lattice basis of $L$.

Example

$A = \begin{pmatrix} 1 & -3 & 5 \end{pmatrix}$

$$\Rightarrow \ker_{\mathbb{Z}}(A) = \mathbb{Z} (2 \quad -1 \quad -1)^T + \mathbb{Z} (1 \quad 2 \quad 1)^T$$

Computation

- Since $A \in \mathbb{Z}^{k,n}$, $L = \ker_{\mathbb{Z}}(A)$ is a lattice.
- 1. Triangulate $A$ with unimodular operations (Hermite decomposition).
- 2. Lattice basis $B$ of $L = \ker_{\mathbb{Z}}(A)$ can be read off the triangular form.
Markov Basis

Let $\alpha^+$ be defined by $\alpha^+_i = \max\{\alpha_i, 0\}$ and $\alpha^- = (-\alpha)^+$. 

Definition
Given a lattice $L = \ker_{\mathbb{Z}}(A)$, $B$ is a Markov basis of $L$ iff $J(B) = \{x^{\alpha^+} - x^{\alpha^-} : \alpha \in B\}$ is an ideal basis of $I(A)$. 
**Markov Basis**

Let $\alpha^+$ be defined by $\alpha_i^+ = \max\{\alpha_i, 0\}$ and $\alpha^- = (-\alpha)^+$.

**Definition**

Given a lattice $L = \ker_{\mathbb{Z}}(A)$, $B$ is a **Markov basis** of $L$ iff $J(B) = \{x^{\alpha^+} - x^{\alpha^-} : \alpha \in B\}$ is an ideal basis of $I(A)$.

**Example**

$J((2 \quad -1 \quad -1)^T, (1 \quad 2 \quad 1)^T) = \{x_1^2 - x_2x_3, x_1x_2^2x_3 - 1\}$ is a Markov basis of $A = (1 \quad -3 \quad 5)$.
Markov Basis

Let $\alpha^+$ be defined by $\alpha^+_i = \max\{\alpha_i, 0\}$ and $\alpha^- = (-\alpha)^+$.

**Definition**

Given a lattice $L = \ker \mathbb{Z}(A)$, $B$ is a **Markov basis** of $L$ iff $J(B) = \{x^{\alpha^+} - x^{\alpha^-} : \alpha \in B\}$ is an ideal basis of $I(A)$.

**Example**

$J((2 \quad -1 \quad -1)^T, (1 \quad 2 \quad 1)^T) = \{x_1^2 - x_2x_3, x_1x_2^2x_3 - 1\}$

is a Markov basis of $A = (1 \quad -3 \quad 5)$

**Computation**

1. Saturation algorithm (Sturmfels, Bigatti et. al)
2. Shadow algorithm (Kesh, Metha)
3. Project-and-Lift algorithm (Hemmecke, Malkin)
Saturation Algorithm

Definition
\[ I : f^\infty = \{ h \in \mathbb{K}[X] : f^k h \in I \text{ for some } k \in \mathbb{N} \} \] saturation of \( I \) wrt. \( f \).
Saturation Algorithm

Definition
\[ I : f^\infty = \{ h \in \mathbb{K}[X] : f^k h \in I \text{ for some } k \in \mathbb{N} \} \] saturation of \( I \) wrt. \( f \).

Fact
Given a basis \( B \) of the lattice \( L = \ker_{\mathbb{Z}}(A), \langle J(B) \rangle : (x_1 \cdots x_n)^\infty = I(A) \).
Saturation Algorithm

Definition

\[ I : f^\infty = \{ h \in \mathbb{K}[X] : f^k h \in I \text{ for some } k \in \mathbb{N} \} \] saturation of \( I \) wrt. \( f \).

Fact

Given a basis \( B \) of the lattice \( L = \ker_{\mathbb{Z}}(A), \langle J(B) \rangle : (x_1 \cdots x_n)^\infty = I(A) \).

Computation

- \( \langle g_1, \ldots, g_s \rangle : f^\infty = \langle g_1, \ldots, g_s, tf - 1 \rangle \cap \mathbb{K}[X] \)

\[ \implies \] Compute Gröbner basis of \( \langle g_1, \ldots, g_s, tx_1 \cdots x_n - 1 \rangle \) wrt. elimination ordering (e.g. lexicographic).

\[ \dagger \] Extra variable necessary.
Saturation Algorithm (2)

Optimization

$I$ $\omega$-graded iff for all $x^\alpha - x^\beta \in I : \omega \cdot \alpha = \omega \cdot \beta$, $\omega_i > 0$

- If $I$ is $\omega$-graded, then $I : x_1^\infty = \langle GB(I, \prec_\omega, k) : x_k^\infty \rangle$.
- $I : (x_1 \cdots x_n)^\infty = (\cdots ((I : x_1^\infty) : x_2^\infty) \cdots) : x_n^\infty$
Saturation Algorithm (2)

Optimization

\( I \) \( \omega \)-graded iff for all \( x^\alpha - x^\beta \in I : \omega \cdot \alpha = \omega \cdot \beta, \; \omega_i > 0 \)

- If \( I \) is \( \omega \)-graded, then \( I : x_k^\infty = \langle \text{GB}(I, \prec_\omega, k) : x_k^\infty \rangle \).
- \( I : (x_1 \cdots x_n)^\infty = (\cdots ((I : x_1^\infty) : x_2^\infty) \cdots) : x_n^\infty \)

Input: \( F = J(B) \) for basis \( B \) of lattice \( L \)
Output: \( G = \) ideal basis of \( I(L) \)

\( G \leftarrow F \)

for \( k \leftarrow 1 \) to \( n \) do
    \( G \leftarrow \text{GB}(G, \prec_\omega, k) : x_k^\infty \)
end
Saturation Algorithm (2)

Optimization

\( I \) \( \omega \)-graded iff for all \( x^\alpha - x^\beta \in I : \omega \cdot \alpha = \omega \cdot \beta, \omega_i > 0 \)

- If \( I \) is \( \omega \)-graded, then \( I : x^\infty_k = \langle \text{GB}(I, <_{\omega}, k) : x^\infty_k \rangle \).
- \( I : (x_1 \cdots x_n)^\infty = (\cdots ((I : x_1^\infty) : x_2^\infty) \cdots) : x_n^\infty \)

**Input**: \( F = J(B) \) for basis \( B \) of lattice \( L \)

**Output**: \( G = \) ideal basis of \( I(L) \)

\[
G \leftarrow F
\]

**for** \( k \leftarrow 1 \) **to** \( n \) **do**

\[
G \leftarrow \text{GB}(G, <_{\omega, k}) : x^\infty_k
\]

**end**

- Can be optimized to maximal \( \frac{n}{2} \) saturation steps.
- \( \frac{n}{2} \) Gröbner basis computations in \( n \) variables.
- Only for graded ideals, otherwise extra variable.
Shadow Algorithm

Projections

Let $\pi_k$ be the projection on the first $k$ variables $(x_{k+1}, \ldots, x_n \leftarrow 1)$.

- Further optimizes Saturation Algorithm
- $G = \text{GB}_k(I, \prec_\omega, k) \subset I$ such that $\pi_k(G) = \text{GB}(\pi_k(I), \prec_\omega, k)$
Shadow Algorithm

Projections

Let $\pi_k$ be the projection on the first $k$ variables ($x_{k+1}, \ldots, x_n \leftarrow 1$).

- Further optimizes Saturation Algorithm
- $G = GB_k(I, \prec_\omega, k) \subset I$ such that $\pi_k(G) = GB(\pi_k(I), \prec_\omega, k)$

**Input:** $F = J(B)$ for basis $B$ of lattice $L$

**Output:** $G =$ ideal basis of $I(L)$

$G \leftarrow F$

for $k \leftarrow 1$ to $n$ do

$G \leftarrow GB_k(G, \prec_\omega, k) : x_k^\infty$

end
Shadow Algorithm

Projections

Let $\pi_k$ be the projection on the first $k$ variables $(x_{k+1}, \ldots, x_n \leftarrow 1)$.

- Further optimizes Saturation Algorithm
- $G = \text{GB}_k(I, \prec_{\omega,k}) \subset I$ such that $\pi_k(G) = \text{GB}(\pi_k(I), \prec_{\omega,k})$

**Input:** $F = J(B)$ for basis $B$ of lattice $L$

**Output:** $G = \text{ideal basis of } I(L)$

$G \leftarrow F$

for $k \leftarrow 1$ to $n$ do
  $G \leftarrow \text{GB}_k(G, \prec_{\omega,k}) : x_k^\infty$
end

\[ \uparrow \text{ Only for graded ideals, otherwise extra variable.} \]
Project-and-Lift Algorithm

Markov Bases

- Also uses projections: $\pi_\sigma$ projects onto $\mathbb{K}[x_i : i \in \sigma]$.
- Instead of computing Gröbner bases, compute Markov bases.
- Sometimes we can simply add a polynomial to lift to larger projection.
- In remaining cases ideal is graded, so no extra variable.
Project-and-Lift Algorithm (2)

**Input:** Basis $B$ of lattice $L$

**Output:** $M =$ Markov basis of $L$

Find (maximal) $\sigma \subset \{1,\ldots,n\}$ s.t. $\ker(\pi_\sigma(L)) = \{0\}$.

Find $M \subset L$ s.t. $\pi_\sigma(M)$ is Markov basis of $\pi_\sigma(L)$.

while $\exists k \in \overline{\sigma} = \{1,\ldots,n\} \setminus \sigma$ do

   if $\exists u \in L : \pi_\sigma(u) \geq 0, u_k > 0$ then

      $M \leftarrow M \cup \{u\}$

   else

      Find $c \in \mathbb{Q}_+^n$ s.t. $\pi_{\overline{\sigma}}(c) = 0$ and $\forall u \in L : c \cdot u = -u_k$.

      $M \leftarrow \text{GB}_\sigma(M, \prec_c)$

   end

$\sigma \leftarrow \sigma \cup \{k\}$

end
Project-and-Lift Algorithm (3)

Optimizations

- Preprocess lattice $L$ with LatticeReduce $\Rightarrow$ short vectors.
- Use Minimize or extreme ray algorithm in order to find $u \in L : \pi_\sigma(u) \geq 0, u_k > 0 \Rightarrow$ short vectors.
- Lift variables $x_k$ last for which no Gröbner basis computation is necessary.
S-Polynomial

**Definition**

\[
S(f, g, \prec) = \frac{\text{LM}(g)}{\gcd(\text{LM}(f), \text{LM}(g))} \cdot f - \frac{\text{LM}(f)}{\gcd(\text{LM}(f), \text{LM}(g))} \cdot g
\]

**Example**

\[
S(x_1^2 - x_2 x_3, x_1 x_2^2 x_3 - 1) = x_2^2 x_3 (x_1^2 - x_2 x_3) - x_1 (x_1 x_2^2 x_3 - 1) = x_1 - x_2^3 x_3^2
\]
Overview of Algorithms

Geometric Buchberger Algorithm

Reduction

Reduce\((f, F, \prec)\)

**Input:** Polynomial \(f, F = \{f_1, \ldots, f_s\}\), monomial ordering \(\prec\)

**Output:** \(h \equiv f \mod \langle f_1, \ldots, f_s \rangle\) s.t. no monomial of \(h\) is in \(\langle \text{LM}(F) \rangle\)

\(h \rightarrow 0\)

**while** \(f \neq 0\) **do**

\(\text{if } \exists f_i : \text{LM}(f_i) \mid \text{LM}(f) \text{ then} \)

\(f \rightarrow f - \frac{\text{LM}(f)}{\text{LM}(f_i)} f_i\)

**else**

\(h \rightarrow h + \text{LM}(f)\)

\(f \rightarrow f - \text{LM}(f)\)

**end**

**end**

Example

\text{Reduce}(x_1 x_2^2 x_3, \{x_1 - x_2^3 x_3^2, x_2^5 x_3^3 - 1\}, \prec_{\text{lex}}): \ x_1 x_2^2 x_3 \xrightarrow{x_1 - x_2^3 x_3^2} x_2^5 x_3^3 \xrightarrow{x_2^5 x_3^3 - 1} 1
Buchberger Algorithm

**Input:** Basis $F$ of ideal $I$, monomial ordering $\prec$

**Output:** $G = \text{Gröbner basis of } I$

$G \leftarrow F$

$Pairs \leftarrow \{(i,j) : 1 \leq i < j \leq \#G\}$

**while** $\exists (i, j) \in Pairs$ **do**

$Pairs \leftarrow Pairs \setminus \{(i,j)\}$

$h \leftarrow S(g_i, g_j, \prec)$

$h \leftarrow \text{Reduce}(h, G, \prec)$

**if** $h \neq 0$ **then**

$G \leftarrow G \cup \{h\}$

$Pairs \leftarrow Pairs \cup \{(i, \#G) : 1 \leq i < \#G\}$

**end**

**end**
Geometric Buchberger Algorithm

Criteria

- Buchberger: Discard \((i, j)\) if \(\text{LM}(g_i), \text{LM}(g_j)\) relatively prime.
- Gebauer-Möller: Basis of syzygies is sufficient. \(\Rightarrow\) Discard superfluous pairs (only approximation, still expensive).
- Bigatti,Hemmecke-Malkin: Criterion-Tail (only for toric ideals) Discard \((i, j)\) if \(S(g_i, g_j)\) reduces to polynomial with smaller grading.
Geometric Buchberger Algorithm

Criteria

- **Buchberger**: Discard \((i, j)\) if \(\text{LM}(g_i), \text{LM}(g_j)\) relatively prime.
- **Gebauer-Möller**: Basis of syzygies is sufficient. \(\Rightarrow\) Discard superfluous pairs (only approximation, still expensive).
- **Bigatti,Hemmecke-Malkin**: Criterion-Tail (only for toric ideals)
  Discard \((i, j)\) if \(S(g_i, g_j)\) reduces to polynomial with smaller grading.

Vector Arithmetic

- Use vectors \(\alpha - \beta\) instead of binomials \(x^\alpha - x^\beta\).
- Automatically cancels common factors (i.e. partly saturates).
- Only works for toric ideals.

Example

\[
\begin{align*}
x_1^3x_2^2x_3 - x_1x_2^5x_3^3 & \quad \rightarrow \quad (2 \quad -3 \quad -2)^T \\
& \quad \rightarrow \quad x_1^2 - x_2^3x_3^2
\end{align*}
\]
Geometric Buchberger Algorithm (2)

Data Structures

- Originally, $G$ is a vector.
- Many searches in $G$ for $\text{LM}(g_i)|m$ for given monomial $m$.
- Vector notation: search $g_i^+ \leq m$.

$\Rightarrow$ Use multidimensional search data structure.
Geometric Buchberger Algorithm (2)

Data Structures

- Originally, $G$ is a vector.
- Many searches in $G$ for $\text{LM}(g_i)|m$ for given monomial $m$.
- Vector notation: search $g_i^+ \leq m$.

$\Rightarrow$ Use multidimensional search data structure.

Suggestions

- $kd$-Lists
- $kd$-Trees
**Geometric Buchberger Algorithm (3)**

**kd-Lists**

- First level: sorted array of first coordinates
- Each entry points to substructure representing all elements with this value.
- $i$-th level: sorted arrays of $i$-th coordinates of resp. substructure

![kd-List Diagram](image)

**Figure:** kd-List for $(2 \ -1 \ -1), \ (1 \ 2 \ 1), \ (0 \ 5 \ 3), \ (1 \ -3 \ -2)$
Geometric Buchberger Algorithm (4)

kd-Trees

- Each level: two subtrees
- Vectors stored at leaves
- Equally balances if possible

Figure: kd-Tree for $(2\ -1\ -1)$, $(1\ 2\ 1)$, $(0\ 5\ 3)$, $(1\ -3\ -2)$
### Experimental Results

<table>
<thead>
<tr>
<th>Input</th>
<th>P&amp;L (uses LR)</th>
<th>ToricGB</th>
<th>ToricGB + LR</th>
</tr>
</thead>
<tbody>
<tr>
<td>Random 1x5</td>
<td>0.06 0.12 0.11</td>
<td>0.00</td>
<td>0.00</td>
</tr>
<tr>
<td>Random 2x5</td>
<td>0.25 0.48 0.42</td>
<td>0.01</td>
<td>0.01</td>
</tr>
<tr>
<td>Random 3x5</td>
<td>2.45 3.42 3.54</td>
<td>0.35</td>
<td>0.31</td>
</tr>
<tr>
<td>Random 4x5</td>
<td>0.74 1.00 1.00</td>
<td>2.90</td>
<td>0.01</td>
</tr>
<tr>
<td>Random 5x5</td>
<td>0.11 0.22 0.22</td>
<td>0.00</td>
<td>0.00</td>
</tr>
<tr>
<td>Frobenius 6x10 dig.</td>
<td>7.56 26.86 14.33</td>
<td>429.76 0.49</td>
<td></td>
</tr>
<tr>
<td>Frobenius 7x10 dig.</td>
<td>168.33 714.25 293.49</td>
<td>383.29 56.80</td>
<td></td>
</tr>
<tr>
<td>Frobenius 8x5 dig.</td>
<td>22.87 64.75 44.31</td>
<td>5.55 1.25</td>
<td></td>
</tr>
<tr>
<td>Frobenius 10x4 dig.</td>
<td>24.98 50.32 43.51</td>
<td>3.38 1.56</td>
<td></td>
</tr>
</tbody>
</table>

**Attention:**

- ToricGB is implemented natively, P&L not.
- Statistic biased by memory leaks.
Conclusion

Data Structures

- Performance gain possible if implemented natively.
- kd-Lists can degenerate, thus kd-trees more promising.
Conclusion

Data Structures

- Performance gain possible if implemented natively.
- kd-Lists can degenerate, thus kd-trees more promising

Mathematica Implementation

- Add Criterion Tail to GroebnerBasis`ToricGroebnerBasis.
- Add capability of projections to GroebnerBasis`ToricGroebnerBasis.
- Use Project-and-Lift algorithm for saturation.
Thank you!