Groebner Basis Conversion Using the FGLM Algorithm

Philip Benge, Valerie Burks, Nicholas Cobar

Louisiana State University

VIGRE REU, July, 2009

Outline

Ideal

Definition

Let k be a field. A subset $I \subset k[x_1, \ldots, x_n]$ is an **ideal** if it satisfies:

(i) $0 \in I$.

(ii) If $f, g \in I$, then $f + g \in I$.

(iii) If $f \in I$ and $h \in k[x_1, \ldots, x_n]$, then $h \cdot f \in I$.

Basis

Definition

If $f_1, \ldots, f_s \in k[x_1, ..., x_n]$, then $I = \langle f_1, ..., f_s \rangle$ is an ideal of $k[x_1, ..., x_n]$. We will call $\langle f_1, ..., f_s \rangle$ the **ideal generated by** $f_1, ..., f_s$, where the polynomials $f_1, ..., f_s$ form a **basis** of I.

By $\langle f_1, ..., f_s \rangle$ we mean that all of the elements in I can be written as $\sum_{i=1}^{n} a_i f_i$ where the a_i are elements in the ring and the f_i are polynomials in the basis.

Variety

When considering a collection of polynomials $(f_1, \ldots, f_s)$ in the field $k[x_1, \ldots, x_n]$ we call the set of common zeroes the **affine variety** of those polynomials.

Variety

When considering a collection of polynomials $(f_1, \ldots, f_s)$ in the field $k[x_1, \ldots, x_n]$ we call the set of common zeroes the **affine variety** of those polynomials.

Affine varieties that consist of a finite collection of points are described as being **zero-dimensional**

Example

Let $f_1 = x - z$, $f_2 = x + z - y$, and $f_3 = x + y + z^2 - 4$.

Example

Let $f_1 = x - z$, $f_2 = x + z - y$, and $f_3 = x + y + z^2 - 4$.

The ideal generated by these polynomials is written as $I =< x - z, x - y + z, x + y + z^2 - 4 >$.

Example

Let $f_1 = x - z$, $f_2 = x + z - y$, and $f_3 = x + y + z^2 - 4$.

The ideal generated by these polynomials is written as $I =< x - z, x - y + z, x + y + z^2 - 4 >$.

And the variety of the set is $\mathbf{V}(I) = \{(-4, -8, -4), (1, 2, 1)\}$.

Example

Let $f_1 = x - z$, $f_2 = x + z - y$, and $f_3 = x + y + z^2 - 4$.

The ideal generated by these polynomials is written as $I =< x - z, x - y + z, x + y + z^2 - 4 >$.

And the variety of the set is $\mathbf{V}(I) = \{(-4, -8, -4), (1, 2, 1)\}$.

Note that there are only two points where all three functions are simultaneously zero. This means that $I =< x - z, x - y + z, x + y + z^2 - 4 >$ is a zero dimensional ideal.

Monomial Ordering

Multivariable Notation is used throughout this paper to describe monomials such that $x^{\alpha} = x_1^{\alpha_1} x_2^{\alpha_2} \ldots x_n^{\alpha_n}$.

Monomial Ordering

Multivariable Notation is used throughout this paper to describe monomials such that $x^{\alpha} = x_1^{\alpha_1} x_2^{\alpha_2} \ldots x_n^{\alpha_n}$.

Characteristics

Monomial Ordering

Multivariable Notation is used throughout this paper to describe monomials such that $x^{\alpha} = x_1^{\alpha_1} x_2^{\alpha_2} \ldots x_n^{\alpha_n}$.

Characteristics

· Total linear ordering on $\mathbb{Z}_{\geq 0}^{n}$

Monomial Ordering

Multivariable Notation is used throughout this paper to describe monomials such that $x^{\alpha} = x_1^{\alpha_1} x_2^{\alpha_2} \ldots x_n^{\alpha_n}$.

Characteristics

· Total linear ordering on $\mathbb{Z}_{\geq 0}^n$

· For monomials x^{α} and x^{β}, if $\alpha > \beta$ and $\gamma \in \mathbb{Z}_{\geq 0}^n$ then $x^{\alpha} x^{\gamma} > x^{\beta} x^{\gamma}$.

Monomial Ordering

Multivariable Notation is used throughout this paper to describe monomials such that $x^{\alpha} = x_1^{\alpha_1} x_2^{\alpha_2} \ldots x_n^{\alpha_n}$.

Characteristics

· Total linear ordering on $\mathbb{Z}^n_{\geq 0}$

· For monomials x^{α} and x^{β}, if $\alpha > \beta$ and $\gamma \in \mathbb{Z}^n_{\geq 0}$ then $x^{\alpha} x^{\gamma} > x^{\beta} x^{\gamma}$.

· Well-ordering on $\mathbb{Z}^n_{\geq 0}$.

Lexicographic Order

In lexicographic, or lex, order, precedence between two monomials is decided by the vector difference between the n-tuples of their respective indices.

Lexicographic Order

In lexicographic, or lex, order, precedence between two monomials is decided by the vector difference between the n-tuples of their respective indices.

$x_1 x_2^2 >_{lex} x_2^3 x_3^4$

Lexicographic Order

In lexicographic, or lex, order, precedence between two monomials is decided by the vector difference between the n-tuples of their respective indices.

$x_1 x_2^2 >_{lex} x_2^3 x_3^4$

$\alpha = (1, 2, 0)$ and $\beta = (0, 3, 4)$

Lexicographic Order

In lexicographic, or lex, order, precedence between two monomials is decided by the vector difference between the n-tuples of their respective indices.

$x_1x_2^2 >_{lex} x_2^3x_3^4$

$\alpha = (1, 2, 0)$ and $\beta = (0, 3, 4)$

$\alpha - \beta = (1, -1, -4)$

Graded Lexicographic Order

Graded Lexicographic (grlex) order classifies one monomial as greater than another based on the total degree of each monomial. If the degrees are equal, order reverts back to lex.

Graded Lexicographic Order

Graded Lexicographic (grlex) order classifies one monomial as greater than another based on the total degree of each monomial. If the degrees are equal, order reverts back to lex.

$x_1x_2^2 <_{grlex} x_2^3x_3^4$ since $\sum \alpha_i = 3$ and $\sum \beta_i = 7$

Graded Reverse Lexicographic Order

Graded Reverse Lexicographic (grevlex) is similar to grlex, but in the event of a tie, grevlex also looks at the n-vector difference, but not like lex.

Graded Reverse Lexicographic Order

Graded Reverse Lexicographic (grevlex) is similar to grlex, but in the event of a tie, grevlex also looks at the n-vector difference, but not like lex.

$x_1^3 x_2^2 >_{grevlex} x_2 x_3^4$ since $\sum \alpha_i = \sum \beta_i = 5$ and $(3, 2, 0) - (0, 1, 4) = (3, 1, -4)$

Polynomial Terminology

Let $f = 3x^3y^2 - x^3y + x^2 - y$.

Polynomial Terminology

Let $f = 3x^3y^2 - x^3y + x^2 - y$.

The **multidegree** of f, or $multideg(f)$ is the maximum n-tuple of the monomials. $multideg(f) = (3, 2)$.

Polynomial Terminology

Let $f = 3x^3y^2 - x^3y + x^2 - y$.

The **multidegree** of f, or $multideg(f)$ is the maximum n-tuple of the monomials. $multideg(f) = (3, 2)$.

The **leading coefficient** of f is $LC(f) = a_{multideg(f)} \in k$.
$LC(f) = 3$

Polynomial Terminology

Let $f = 3x^3y^2 - x^3y + x^2 - y$.

The **multidegree** of f, or $multideg(f)$ is the maximum n-tuple of the monomials. $multideg(f) = (3, 2)$.

The **leading coefficient** of f is $LC(f) = a_{multideg(f)} \in k$.
$LC(f) = 3$

The **leading monomial** of f is $LM(f) = x^{multideg(f)}$ (with coefficient 1). $LM(f) = x^3y^2$

Polynomial Terminology

Let $f = 3x^3y^2 - x^3y + x^2 - y$.

The **multidegree** of f, or $multideg(f)$ is the maximum n-tuple of the monomials. $multideg(f) = (3, 2)$.

The **leading coefficient** of f is $LC(f) = a_{multideg(f)} \in k$.
$LC(f) = 3$

The **leading monomial** of f is $LM(f) = x^{multideg(f)}$ (with coefficient 1). $LM(f) = x^3y^2$

The **leading term** of f is $LT(f) = LC(f) \cdot LM(f)$. $LT(f) = 3x^3y^2$

Groebner Basis

Definition

Fix a monomial order. A finite subset $G = \{g_1, ..., g_t\}$ of an ideal, I, is said to be a **Groebner basis** if $< LT(g_1), ..., LT(g_t) >$ = $< LT(I) >$.

Groebner Basis

Definition

Fix a monomial order. A finite subset $G = \{g_1, ..., g_t\}$ of an ideal, I, is said to be a **Groebner basis** if $< LT(g_1), ..., LT(g_t) >$ = $< LT(I) >$.

Definition

A Groebner basis is call **reduced** if

1. $LC(p) = 1$ for all $p \in G$.
2. For all $p \in G$, no monomial of p lies in $\langle LT(G - \{p\})\rangle$.

Groebner Basis

Definition

Fix a monomial order. A finite subset $G = \{g_1, ..., g_t\}$ of an ideal, I, is said to be a **Groebner basis** if $< LT(g_1), ..., LT(g_t) >$ = $< LT(I) >$.

Definition

A Groebner basis is call **reduced** if

1. $LC(p) = 1$ for all $p \in G$.
2. For all $p \in G$, no monomial of p lies in $\langle LT(G - \{p\})\rangle$.

This means that the leading term of any element of I must be divisible by one of the $LT(g_i)$ for G to be a Groebner basis.

Division Algorithm

Definition

Fix a monomial order $>$ on $\mathbb{Z}^n_{\geq 0}$, and let $H = (h_1, ..., h_s)$ be an ordered s-tuple of polynomials in $k[x_1, ..., x_n]$.

Division Algorithm

Definition

Fix a monomial order $>$ on $\mathbb{Z}^n_{\geq 0}$, and let $H = (h_1, ..., h_s)$ be an ordered s-tuple of polynomials in $k[x_1, ..., x_n]$. Then every $f \in k[x_1, ..., x_n]$ can be written as

$$f = a_1 h_1 + ... + a_s h_s + r,$$

where $a_i, r \in k[x_1, ..., x_n]$,

Division Algorithm

Definition

Fix a monomial order $>$ on $\mathbb{Z}^n_{\geq 0}$, and let $H = (h_1, ..., h_s)$ be an ordered s-tuple of polynomials in $k[x_1, ..., x_n]$. Then every $f \in k[x_1, ..., x_n]$ can be written as

$$f = a_1 h_1 + ... + a_s h_s + r,$$

where $a_i, r \in k[x_1, ..., x_n]$, and either $r = 0$ or r is a linear combination, with coefficients in k, of monomials, none of which is divisible by any of $LT(h_1), ..., LT(h_s)$.

Division Algorithm

Definition

Fix a monomial order $>$ on $\mathbb{Z}^n_{\geq 0}$, and let $H = (h_1, ..., h_s)$ be an ordered s-tuple of polynomials in $k[x_1, ..., x_n]$. Then every $f \in k[x_1, ..., x_n]$ can be written as

$$f = a_1 h_1 + ... + a_s h_s + r,$$

where $a_i, r \in k[x_1, ..., x_n]$, and either $r = 0$ or r is a linear combination, with coefficients in k, of monomials, none of which is divisible by any of $LT(h_1), ..., LT(h_s)$. We will call r a **remainder** of f on division by H. Furthermore if $a_i, h_i \neq 0$, then we have

$$multideg(f) \geq multideg(a_i h_i).$$

Division Algorithm

EXAMPLE

Example

Consider the ideal
$I =< x^2 + y^2 + z^2 - 2x, x^3 - yz - x, x - y + 2z >$.
Then the Groebner basis is,
$G = \{x - y + 2z, 2y^2 - 4yz + 5z^2 - 2y + 4z, 3yz^2 + 4z^3 - 10yz + 11z^2, 375z^4 + 974z^3 - 1460yz + 144z^2\}$.

Example

Consider the ideal
$I =< x^2 + y^2 + z^2 - 2x, x^3 - yz - x, x - y + 2z >$.
Then the Groebner basis is,
$G = \{x - y + 2z, 2y^2 - 4yz + 5z^2 - 2y + 4z, 3yz^2 + 4z^3 - 10yz + 11z^2, 375z^4 + 974z^3 - 1460yz + 144z^2\}$.

A Groebner basis has many useful applications, one of which is the "ideal membership" problem.

Example

Consider the ideal
$I =< x^2 + y^2 + z^2 - 2x, x^3 - yz - x, x - y + 2z >$.
Then the Groebner basis is,
$G = \{x - y + 2z, 2y^2 - 4yz + 5z^2 - 2y + 4z, 3yz^2 + 4z^3 - 10yz + 11z^2, 375z^4 + 974z^3 - 1460yz + 144z^2\}$.

A Groebner basis has many useful applications, one of which is the "ideal membership" problem.

Let $f = 14xy - 2z + 3$, Then dividing f by $G = \{g_1, g_2, g_3, g_4\}$ using the division algorithm yields

Example

Consider the ideal
$I =< x^2 + y^2 + z^2 - 2x, x^3 - yz - x, x - y + 2z >$.
Then the Groebner basis is,
$G = \{x - y + 2z, 2y^2 - 4yz + 5z^2 - 2y + 4z, 3yz^2 + 4z^3 - 10yz + 11z^2, 375z^4 + 974z^3 - 1460yz + 144z^2\}$.

A Groebner basis has many useful applications, one of which is the "ideal membership" problem.

Let $f = 14xy - 2z + 3$, Then dividing f by $G = \{g_1, g_2, g_3, g_4\}$ using the division algorithm yields $f = 14y \cdot g_1 + 7 \cdot g_2 + r$, where $r = -35z^2 + 14y - 30z + 3$. Since $r \neq 0$, $f \notin I$.

Zero-Dimensional Ideals

A zero-dimensional ideal has a finite number of elements in its variety.

Zero-Dimensional Ideals

A zero-dimensional ideal has a finite number of elements in its variety. Also if we have a zero-dimensional ideal, then for each variable x_i, there is a polynomial in the Groebner basis for I, with a power of x_i as a leading monomial.

Zero-Dimensional Ideals

A zero-dimensional ideal has a finite number of elements in its variety. Also if we have a zero-dimensional ideal, then for each

variable x_i, there is a polynomial in the Groebner basis for I, with a power of x_i as a leading monomial.

Let $I =< xy^3 - x^2, x^3y^2 - y >$ in $\mathbb{R}[x, y]$. Using grlex the Groebner basis is $G = \{x^3y^2 - y, x^4 - y^2, xy^3 - x^2, y^4 - xy\}$ and $\langle LT(I)\rangle = \langle x^3y^2, x^4, xy^3, y^4\rangle$

Zero-Dimensional Ideals

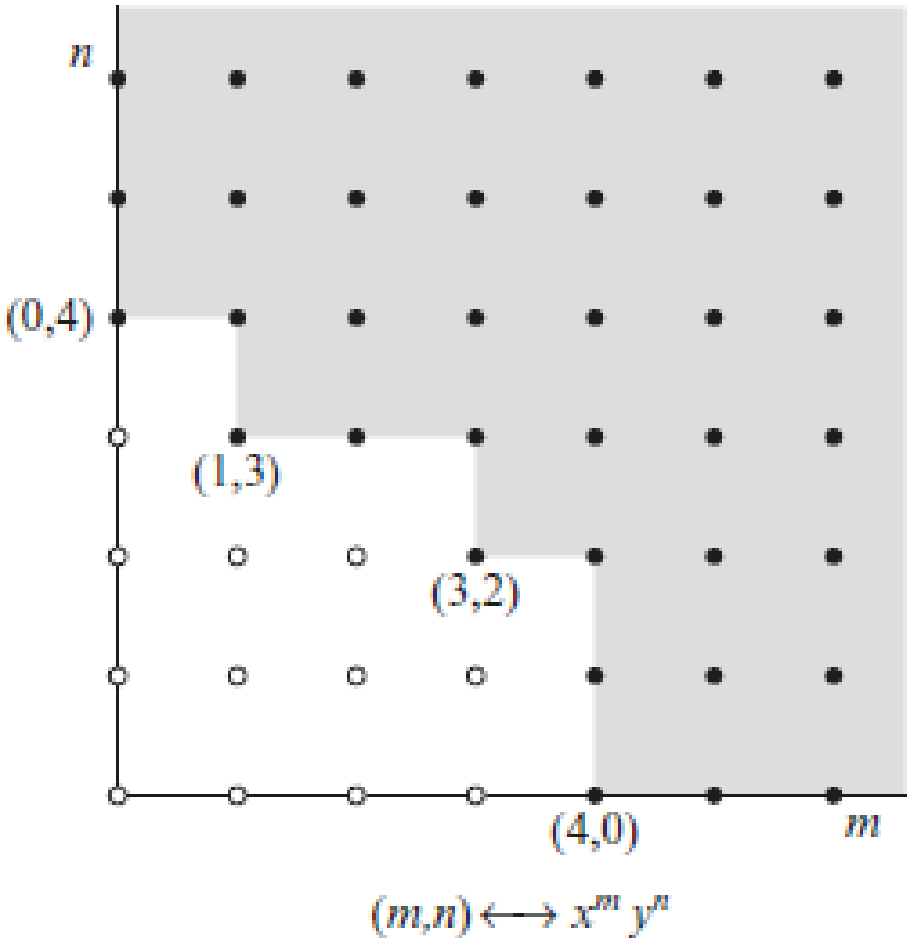

Figure: 1

FGLM Algorithm

The FGLM Algorithm - developed by J.C. Faugère, P. Gianni, D. Lazard, and T. Mora

FGLM Algorithm

The FGLM Algorithm - developed by J.C. Faugère, P. Gianni, D. Lazard, and T. Mora

FGLM converts a Groebner basis for an ideal relative to a certain monomial order to a Groebner basis for the same ideal relative to a different monomial order

FGLM Algorithm

The FGLM algorithm consists of three main parts.
(1) Main Loop: For this first step the user will take the current input monomial x^{α}, initially 1, and find $\overline{x^{\alpha}}^{G}$.

FGLM Algorithm

The FGLM algorithm consists of three main parts.
(1) Main Loop: For this first step the user will take the current input monomial x^α, initially 1, and find $\overline{x^\alpha}^G$.

In the first case, $\overline{x^\alpha}^G$ is *linearly independent* of the items in B_{lex}.
In this event, x^α is added to B_{lex}.

FGLM Algorithm

In the second case, if $\overline{x^{\alpha}}^{G}$ is *linearly dependent* on the remainders of the other members of B_{lex}, then we have a linear combination such that

$$\overline{x^{\alpha}}^{G} - \sum_{j} c_j \overline{x^{\alpha(j)}}^{G} = 0$$

where $x^{\alpha(j)} \in B_{lex}$ and $c_j \in k$.

FGLM Algorithm

In the second case, if $\overline{x^{\alpha}}^{G}$ is *linearly dependent* on the remainders of the other members of B_{lex}, then we have a linear combination such that

$$\overline{x^{\alpha}}^{G} - \sum_{j} c_j \overline{x^{\alpha(j)}}^{G} = 0$$

where $x^{\alpha(j)} \in B_{lex}$ and $c_j \in k$.

This implies that $g = x^{\alpha} - \sum_j c_j x^{\alpha(j)} \in I$. So we add g to the list G_{lex} as the last element,

FGLM Algorithm

In the second case, if $\overline{x^\alpha}^G$ is *linearly dependent* on the remainders of the other members of B_{lex}, then we have a linear combination such that

$$\overline{x^\alpha}^G - \sum_j c_j \overline{x^{\alpha(j)}}^G = 0$$

where $x^{\alpha(j)} \in B_{lex}$ and $c_j \in k$.

This implies that $g = x^\alpha - \sum_j c_j x^{\alpha(j)} \in I$. So we add g to the list G_{lex} as the last element, then G_{lex} must be tested to see if it is the desired Groebner basis.

FGLM Algorithm

In the second case, if $\overline{x^\alpha}^G$ is *linearly dependent* on the remainders of the other members of B_{lex}, then we have a linear combination such that

$$\overline{x^\alpha}^G - \sum_j c_j \overline{x^{\alpha(j)}}^G = 0$$

where $x^{\alpha(j)} \in B_{lex}$ and $c_j \in k$.

This implies that $g = x^\alpha - \sum_j c_j x^{\alpha(j)} \in I$. So we add g to the list G_{lex} as the last element, then G_{lex} must be tested to see if it is the desired Groebner basis.

To do this, we use the Termination Test, the second part of the FGLM algorithm.

FGLM Algorithm

(2) Termination Test: In the event that a new polynomial, g, was added to G_{lex}, the user must compute $LT(g)$. If the leading term of g is a power of x_i, where x_i is the greatest variable in the new monomial ordering, then the algorithm terminates.

FGLM Algorithm

(2) Termination Test: In the event that a new polynomial, g, was added to G_{lex}, the user must compute $LT(g)$. If the leading term of g is a power of x_i, where x_i is the greatest variable in the new monomial ordering, then the algorithm terminates.

Otherwise, proceed to the third part of the algorithm, the Next Monomial step.

FGLM Algorithm

(3) Next Monomial: Replace the x^{α} that has just been processed with the next monomial with respect to the new order which is not divisible by any of the leading terms of the polynomials in G_{lex}.

FGLM Algorithm

(3) Next Monomial: Replace the x^{α} that has just been processed with the next monomial with respect to the new order which is not divisible by any of the leading terms of the polynomials in G_{lex}.

The user repeats the steps of this algorithm until the conditions are met for the Termination Test.

FGLM Algorithm

Notice that whenever a polynomial g is added to G_{lex}, its leading term is $LT(g) = x^{\alpha}$ with coefficient 1, hence each basis element must be monic.

FGLM Algorithm

Notice that whenever a polynomial g is added to G_{lex}, its leading term is $LT(g) = x^{\alpha}$ with coefficient 1, hence each basis element must be monic.

Also, because the leading term of each basis element is linearly independent of the leading terms of all other elements, the Groebner basis obtained from this algorithm must be reduced.

Example of the FGLM Algorithm

Consider

$$I =< x^2 + 2y^2 - y - 2z, x^2 - 8y^2 + 10z - 1, x^2 - 7yz >$$

Example of the FGLM Algorithm

Consider

$$I =< x^2 + 2y^2 - y - 2z, x^2 - 8y^2 + 10z - 1, x^2 - 7yz >$$

which has a graded reverse lexicographic Groebner basis

$G = \{980z^2 - 18y - 201z + 13, 35yz - 4y + 2z - 1, 10y^2 - y - 12z + 1, 5x^2 - 4y + 2z - 1\}$.

Example of the FGLM Algorithm

Consider

$$I = < x^2 + 2y^2 - y - 2z, x^2 - 8y^2 + 10z - 1, x^2 - 7yz >$$

which has a graded reverse lexicographic Groebner basis

$G = \{980z^2 - 18y - 201z + 13, 35yz - 4y + 2z - 1, 10y^2 - y - 12z + 1, 5x^2 - 4y + 2z - 1\}$.

We will now convert G into a lexicographic Groebner basis using the FGLM algorithm.

Example of the FGLM Algorithm

We start with the least variable in the monomial order, z,

Example of the FGLM Algorithm

We start with the least variable in the monomial order, z,

Calculate the remainder of z^0 under division by G,

Example of the FGLM Algorithm

We start with the least variable in the monomial order, z,

Calculate the remainder of z^0 under division by G,

Then increase the degree of z and find remainders under division by G.

Example of the FGLM Algorithm

We start with the least variable in the monomial order, z,

Calculate the remainder of z^0 under division by G,

Then increase the degree of z and find remainders under division by G.

We stop once we find a remainder that is linearly dependent upon the other remainders.

Example of the FGLM Algorithm

$$\overline{z^0}^G = \overline{1}^G = 1$$

$$\overline{z}^G = z$$

$$\overline{z^2}^G = \frac{9}{490}y + \frac{201}{980}z - \frac{13}{980}$$

$$\overline{z^3}^G = \frac{2817}{480200}y + \frac{26653}{960400}z - \frac{2109}{960400}$$

Example of the FGLM Algorithm

$$\overline{z^0}^G = \overline{1}^G = 1$$

$$\overline{z}^G = z$$

$$\overline{z^2}^G = \frac{9}{490}y + \frac{201}{980}z - \frac{13}{980}$$

$$\overline{z^3}^G = \frac{2817}{480200}y + \frac{26653}{960400}z - \frac{2109}{960400}$$

Since $\overline{z^3}^G$ is a linear combination of $\overline{1}^G$, $\overline{z}^G$, and $\overline{z^2}^G$,

Example of the FGLM Algorithm

$$\overline{z^0}^G = \overline{1}^G = 1$$

$$\overline{z}^G = z$$

$$\overline{z^2}^G = \frac{9}{490}y + \frac{201}{980}z - \frac{13}{980}$$

$$\overline{z^3}^G = \frac{2817}{480200}y + \frac{26653}{960400}z - \frac{2109}{960400}$$

Since $\overline{z^3}^G$ is a linear combination of $\overline{1}^G$, $\overline{z}^G$, and $\overline{z^2}^G$,

$$g_1 = z^3 - \frac{313}{980}z^2 + \frac{37}{980}z + \frac{1}{490}$$

is the first polynomial added to G_{lex}.

Example of the FGLM Algorithm

Since we added a polynomial to G_{lex},

Example of the FGLM Algorithm

Since we added a polynomial to G_{lex},

the Next Monomial test tells us to consider the next monomial in the order, y.

Example of the FGLM Algorithm

We find y itself can be expressed as a linear combination of

$$y = \frac{490}{9}\overline{z^2}^G - \frac{67}{6}\overline{z}^G + \frac{13}{18}$$

Example of the FGLM Algorithm

We find y itself can be expressed as a linear combination of

$$y = \frac{490}{9}\overline{z^2}^G - \frac{67}{6}\overline{z}^G + \frac{13}{18}$$

So

$$g_2 = y - \frac{490}{9}z^2 + \frac{67}{6}z - \frac{13}{18}$$

is added to G_{lex}.

Example of the FGLM Algorithm

Now we move to the last monomial in the order, x.

$$\overline{x}^G = x$$

$$\overline{x^2}^G = \frac{4}{5}y - \frac{2}{5}z + \frac{1}{5}$$

Example of the FGLM Algorithm

Now we move to the last monomial in the order, x.

$\overline{x}^G = x$

$\overline{x^2}^G = \frac{4}{5}y - \frac{2}{5}z + \frac{1}{5}$

Notice $\overline{x^2}^G$ can be expressed as a linear combination of $\overline{y}^G$ and $\overline{z}^G$

Example of the FGLM Algorithm

Now we move to the last monomial in the order, x.

$\overline{x}^G = x$

$\overline{x^2}^G = \frac{4}{5}y - \frac{2}{5}z + \frac{1}{5}$

Notice $\overline{x^2}^G$ can be expressed as a linear combination of $\overline{y}^G$ and $\overline{z}^G$

So

$$g_3 = x^2 - \frac{392}{9}z^2 + \frac{28}{3}z - \frac{7}{9}$$

is the final function added to G_{lex}

Example of the FGLM Algorithm

Since x is the largest variable in our order, by the Termination Test,

we now have a lexicographic Groebner basis for I

$G_{lex} =$
$$\{z^3 - \frac{313}{980}z^2 + \frac{37}{980}z + \frac{1}{490}, y - \frac{490}{9}z^2 + \frac{67}{6}z - \frac{13}{18}, x^2 - \frac{392}{9}z^2 + \frac{28}{3}z - \frac{7}{9}\}$$

Lemma and Theorem

Lemma

(Dickson's Lemma) Given an infinite list $x^{\alpha(1)}, x^{\alpha(2)}, \ldots$ *of monomials in* $k[x_1, \ldots, x_n]$, *there is an* $N \in \mathbb{N}$ *such that every* $x^{\alpha(i)}$ *is divisible by one of* $x^{\alpha(1)}, \ldots, x^{\alpha(N)}$.

Lemma and Theorem

Lemma

(Dickson's Lemma) Given an infinite list $x^{\alpha(1)}, x^{\alpha(2)}, \ldots$ *of monomials in* $k[x_1, \ldots, x_n]$, *there is an* $N \in \mathbb{N}$ *such that every* $x^{\alpha(i)}$ *is divisible by one of* $x^{\alpha(1)}, \ldots, x^{\alpha(N)}$.

Theorem

The algorithm described above terminates on every input Groebner basis, G, that generates a zero-dimensional ideal I, and correctly computes a lex Groebner basis, G_{lex}, *for I and the lex monomial basis,* B_{lex}, *for the quotient ring* $A = k[x_1, \ldots, x_n]/I$.

Proof

We observe that monomials are added to the list B_{lex} in strictly increasing lex order

Proof

We observe that monomials are added to the list B_{lex} in strictly increasing lex order

so if $G_{lex} = \{g_1, \ldots, g_k\}$, then $LT(g_1) <_{lex} \ldots <_{lex} LT(g_k)$.

Proof

We observe that monomials are added to the list B_{lex} in strictly increasing lex order

so if $G_{lex} = \{g_1, \ldots, g_k\}$, then $LT(g_1) <_{lex} \ldots <_{lex} LT(g_k)$.

when the Main Loop adds a new polynomial g_{k+1} to $G_{lex} = \{g_1, \ldots, g_k\}$, the leading term $LT(g_{k+1})$ is the input monomial in the Main Loop

Proof

Since the input monomials are provided by the Next Monomial procedure, it follows that for all k,

$LT(g_{k+1})$ is divisible by none of $LT(g_1), \ldots, LT(g_k)$.

Proof

If the algorithm did not terminate for some input G, then the Main Loop would be executed infinitely many times.

Proof

If the algorithm did not terminate for some input G, then the Main Loop would be executed infinitely many times.

Cases:

1. G_{lex} would contain an infinite list $LT(g_1), LT(g_2), \ldots$ of monomials.

Proof

If the algorithm did not terminate for some input G, then the Main Loop would be executed infinitely many times.

Cases:

1. G_{lex} would contain an infinite list $LT(g_1), LT(g_2), \ldots$ of monomials.
2. B_{lex} would contain infinitely many monomials $x^{\alpha(j)}$ whose remainders on division by G were linearly independent in A.

Proof

If:

1. : When applied to $LT(g_1), LT(g_2), \ldots,$ Dickson's Lemma would contradict the fact that $LT(g_{k+1})$ is divisible by none of $LT(g_1), \ldots, LT(g_k)$.

Proof

If:

1. : When applied to $LT(g_1), LT(g_2), \ldots,$ Dickson's Lemma would contradict the fact that $LT(g_{k+1})$ is divisible by none of $LT(g_1), \ldots, LT(g_k)$.

2. : This would contradict the assumption that I is zero-dimensional.

As a result, the algorithm always terminates if G generates a zero-dimensional ideal I.

Proof

We now suppose for a contradiction that there were some $g \in I$ such that $LT(g)$ is not a multiple of any of the $LT(g_i)$, $i = 1, \ldots, k$.

Proof

We now suppose for a contradiction that there were some $g \in I$ such that $LT(g)$ is not a multiple of any of the $LT(g_i)$, $i = 1, \ldots, k$.

If $LT(g) > LT(g_k) = x_1^{a_1}$, then one easily sees that $LT(g)$ is a multiple of $LT(g_k)$.

Proof

We now suppose for a contradiction that there were some $g \in I$ such that $LT(g)$ is not a multiple of any of the $LT(g_i)$, $i = 1, \ldots, k$.

If $LT(g) > LT(g_k) = x_1^{a_1}$, then one easily sees that $LT(g)$ is a multiple of $LT(g_k)$.

But this case cannot occur, which means that $LT(g_i) < LT(g) \leq LT(g_{i+1})$ for some $i < k$.

Proof

We now suppose for a contradiction that there were some $g \in I$ such that $LT(g)$ is not a multiple of any of the $LT(g_i), i = 1, \ldots, k$.

If $LT(g) > LT(g_k) = x_1^{a_1}$, then one easily sees that $LT(g)$ is a multiple of $LT(g_k)$.

But this case cannot occur, which means that $LT(g_i) < LT(g) \leq LT(g_{i+1})$ for some $i < k$.

It is easy to show that the non-leading monomials that appear in g would have been included in B_{lex} by the time $LT(g)$ was reached by the Next Monomial procedure

Proof

Hence:

Proof

Hence:

G_{lex} is a lex Groebner basis for I.

Proof

To find a monomial basis for $A = k[x_1, \ldots, x_n]/I$, we need to find all monomials not in $\langle LT(g) \rangle$ for all $g \in G_{lex}$.

Proof

To find a monomial basis for $A = k[x_1, \ldots, x_n]/I$, we need to find all monomials not in $\langle LT(g) \rangle$ for all $g \in G_{lex}$.

But B_{lex} contains all such monomials, so B_{lex} forms a monomial basis for the quotient ring as a k-vector space.

Proof

To find a monomial basis for $A = k[x_1, \ldots, x_n]/I$, we need to find all monomials not in $\langle LT(g) \rangle$ for all $g \in G_{lex}$.

But B_{lex} contains all such monomials, so B_{lex} forms a monomial basis for the quotient ring as a k-vector space.

So B_{lex} consists of all monomials determined by the Groebner basis G_{lex}. $\square$

Why Zero-Dimensional?

What if I were not zero dimensional?

Why Zero-Dimensional?

What if I were not zero dimensional?

The Main Loop would never terminate.

Why Zero-Dimensional?

What if I were not zero dimensional?

The Main Loop would never terminate.

We would need to know an upperbound on the resulting Groebner basis.

Conclusion

We use lexicographic order to solve systems of equations.

Conclusion

We use lexicographic order to solve systems of equations.

It's computationally expensive.

Conclusion

We use lexicographic order to solve systems of equations.

It's computationally expensive.

$I =< x^5 + y^5 + z^5 - 1, x^3 + y^3 + z^2 - 1 >$

Conclusion

We use lexicographic order to solve systems of equations.

It's computationally expensive.

$I =< x^5 + y^5 + z^5 - 1, x^3 + y^3 + z^2 - 1 >$

415 terms, total degree of 37, with a largest coefficient of 141,592,532,029,352

Conclusion

We should find grevlex Groebner basis first.

Conclusion

We should find grevlex Groebner basis first.

38 terms, total degree of 11, and the largest coefficient is 7

Conclusion

We should find grevlex Groebner basis first.

38 terms, total degree of 11, and the largest coefficient is 7

Then use the FGLM algorithm.

The End

The End