

Gröbner Basis Conversion

The book [2] is a recommended source for this lecture, leading into the FGLM Algorithm [3] to convert a Gröbner basis for any order into a triangular shape. As application, we mention the cyclic n -roots problem. In this lecture, ideals are generated by polynomials with finitely many common zeroes.

1 Normal Sets and the Quotient Algebra

A system of polynomials $f(\mathbf{x}) = \mathbf{0}$, $f = (f_1, f_2, \dots, f_N)$ defines the ideal $I = \langle f \rangle$. We assume $\#V(I)$ is finite. When a lexicographic order is used, the Gröbner basis for I has a triangular shape, allowing (almost) to read off the solutions. Unfortunately, it turns out that computing in a lexicographic term order can be very expensive and numerically undesirable because of excessive growth of the coefficients. While we write \mathbb{C} as our coefficient field, in computer algebra exact arithmetic is more common.

For Gröbner basis computations, the most favorable term order is proven to be the degree reverse lexicographic order, denoted by $>_{\text{grevlex}}$. Ordering the variables in \mathbf{x} by $x_1 > x_2 > \dots > x_n$, for two monomials $\mathbf{x}^{\mathbf{a}}$ and $\mathbf{x}^{\mathbf{b}}$, we define $\mathbf{x}^{\mathbf{a}} >_{\text{grevlex}} \mathbf{x}^{\mathbf{b}}$ if $\deg(\mathbf{x}^{\mathbf{a}}) > \deg(\mathbf{x}^{\mathbf{b}})$ or else if $\deg(\mathbf{x}^{\mathbf{a}}) = \deg(\mathbf{x}^{\mathbf{b}})$ and then the rightmost nonzero entry in $\mathbf{a} - \mathbf{b}$ is negative. For example: $x_1 x_2^4 x_3 >_{\text{grevlex}} x_1^2 x_2^2 x_3^2$. Although lexicographic and graded lexicographic orders of monomials seem most natural, the degree reverse lexicographic order is convenient in the study of singularities and local rings.

After fixing the term order $>$, once we have a Gröbner basis $g_{>}$ for the ideal I , the division algorithm yields a unique remainder r when dividing a polynomial h by $g_{>}$. We denote $h \rightarrow_{g_{>}} r$ and say that r is the normal form of h modulo $g_{>}$.

If the solution set V of the system $f(\mathbf{x}) = \mathbf{0}$ is finite in number (or zero dimensional), then the quotient ring $\mathbb{C}[\mathbf{x}]/I$ is a finite dimensional vector space. In particular: $\dim(\mathbb{C}[\mathbf{x}]/I) = \#V$, $V = V(I)$. To find a basis for $\mathbb{C}[\mathbf{x}]/I$, we compute a Gröbner basis $g_{>}$ and collect those monomials not in $\langle \text{LT}(g_{>}) \rangle$ in the so-called normal set $\mathcal{N}_{>}$. This normal set $\mathcal{N}_{>}$ contains all monomials which may appear in a remainder of all polynomials rewritten modulo $g_{>}$. The definition of the normal set

$$\mathcal{N}_{>} = \{ \mathbf{x}^{\mathbf{a}} \mid \mathbf{x}^{\mathbf{a}} \notin \text{LT}(g_{>}) \} \quad (1)$$

shows how to compute $\mathcal{N}_{>}$ using enumeration on a given basis $\text{LT}(g_{>})$ for the initial ideal. The number of monomials in any normal set for a zero dimensional ideal equals $\#V$, or $\#\mathcal{N}_{>} = \#V$. While the content of $\mathcal{N}_{>}$ depends on $>$, the cardinality $\#\mathcal{N}_{>}$ is independent from $>$. The finiteness of the solution set implies that for $i = 1, 2, \dots, n$: $x_i^{d_i} \in \mathcal{N}_{>}$, for some power d_i .

For any polynomial $h \in \mathbb{C}[\mathbf{x}]$, a Gröbner basis $g_{>}$ with corresponding normal set $\mathcal{N}_{>}$, we write

$$h \rightarrow_{g_{>}} r = \sum_{\mathbf{x}^{\mathbf{a}} \in \mathcal{N}_{>}} c_{\mathbf{a}} x^{\mathbf{a}}, \quad c_{\mathbf{a}} \in \mathbb{C}, \quad (2)$$

i.e.: the remainder r is a linear combination of the monomials in the normal set. The quotient ring $\mathbb{C}[\mathbf{x}]/I$ is thus a finite dimensional vector space with basis $\mathcal{N}_{>}$.

The quotient ring $\mathbb{C}[\mathbf{x}]/I$ turns into the quotient algebra, when considering the multiplication

$$\begin{aligned} * & : \mathcal{N}_{>} \times \mathcal{N}_{>} & \rightarrow & \mathbb{C}[\mathbf{x}]/I \\ & (\mathbf{x}^{\mathbf{a}}, \mathbf{x}^{\mathbf{b}}) & \mapsto & \mathbf{x}^{\mathbf{a}+\mathbf{b}} \rightarrow_{g_{>}} r. \end{aligned} \quad (3)$$

Applying this operation to all possible pairs of elements in $\mathbf{n}_{>}$ leads to a multiplication table. With a Gröbner basis $g_{>}$ for a zero dimensional ideal I we order the monomials in the normal set $\mathcal{N}_{>}$ into the vector $\mathbf{n}_{>}$. For any coordinate x_i , consider the multiplication map

$$\begin{aligned} m_{x_i} & : \mathbb{C}[\mathbf{x}]/I & \rightarrow & \mathbb{C}[\mathbf{x}]/I \\ & h & \mapsto & (x_i \cdot h) \rightarrow_{g_{>}} r. \end{aligned} \quad (4)$$

The map m_{x_i} is linear and leads to the eigenvalue problem $x_i \mathbf{n}_{>} = m_{x_i} \mathbf{n}_{>}$.

2 The cyclic n -roots problem

The origins of the so-called cyclic n -roots problems can be traced back to [1]. It is one of the most popular benchmarks in polynomial system solving, and also used in [3].

The cyclic 5-roots problem takes the following form:

$$f(\mathbf{x}) = \begin{cases} x_1 + x_2 + x_3 + x_4 + x_5 = 0 \\ x_1x_2 + x_2x_3 + x_3x_4 + x_4x_5 + x_5x_1 = 0 \\ x_1x_2x_3 + x_2x_3x_4 + x_3x_4x_5 + x_4x_5x_1 + x_5x_1x_2 = 0 \\ x_1x_2x_3x_4 + x_2x_3x_4x_5 + x_3x_4x_5x_1 + x_4x_5x_1x_2 + x_5x_1x_2x_3 = 0 \\ x_1x_2x_3x_4x_5 - 1 = 0. \end{cases} \quad (5)$$

Observe the symmetry in the system.

The table below shows the number of solution $\#V$ for increasing dimension n . The ∞ in the table marks cases for which there are positive dimensional solution sets.

n	3	4	5	6	7	8	9	10	11	12
$\#V$	6	∞	70	156	924	∞	∞	34,940	184,756	∞

3 Gröbner Conversion in Maple

Below is a session with Maple, on the cyclic 3-roots problem:

```
[> f3 := [ x[1] + x[2] + x[3],
          x[1]*x[2] + x[2]*x[3] + x[3]*x[1],
          x[1]*x[2]*x[3] - 1];
[> v := op(indets(f3));
          v := x[1], x[2], x[3]
[> g1 := Groebner[Basis](f3,tdeg(v));
          g1 := [x[1]+x[2]+x[3], x[2]^2+x[2]*x[3]+x[3]^2, -1+x[3]^3]
[> ns := Groebner[NormalSet](g1,tdeg(v))[1];
          ns := [1, x[3], x[2], x[3]^2, x[2]*x[3], x[2]*x[3]^2]
[> g2 := Groebner[FGLM](g1,tdeg(v),plex(x[3],x[2],x[1]));
          g2 := [-1+x[1]^3, x[1]*x[2]+x[1]^2+x[2]^2, x[1]+x[2]+x[3]]
```

Note that in the Gröbner basis conversion we changed the order of the variables in order to provoke some difference with the original Gröbner basis computed with the total degree monomial order.

The multiplication map M_{x_1} with the total degree order is

$$M_{x_1} = \begin{bmatrix} 0 & -1 & -1 & 0 & 0 & 0 \\ 0 & 0 & 0 & -1 & -1 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 \\ -1 & 0 & 0 & 0 & 0 & -1 \\ 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 \end{bmatrix} \quad \text{and} \quad L_{x_3} = \begin{bmatrix} 0 & -1 & 0 & -1 & 0 & 0 \\ 0 & 0 & -1 & 0 & -1 & 0 \\ -1 & 0 & 0 & 0 & 0 & -1 \\ 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 \end{bmatrix} \quad (6)$$

is the multiplication map with the pure lexicographic order $x_3 > x_2 > x_1$ and $\mathbf{n}_{>\text{lex}} = [1, x_1, x_1^2, x_2, x_1x_2, x_1^2x_2]$.

We can view the Gröbner basis conversion as the rewriting of $x_1\mathbf{n}_{>\text{tdeg}} = M_{x_1}\mathbf{n}_{>\text{tdeg}}$ into $x_3\mathbf{n}_{>\text{lex}} = L_{x_3}\mathbf{n}_{>\text{lex}}$.

In [3], the set of all monomials $x_i b$ with $b \in \mathcal{N}_{>}$ and $x_i b \notin \mathcal{N}_{>}$, for all i is called the bordering of the Gröbner basis.

4 The FGLM Algorithm

The FGLM Algorithm is named after its authors [3]. Below we describe the algorithm in pseudo code, assuming the ideal is zero dimensional.

The algorithm first considers all powers of x_1 and tries to write these powers as a linear combination of the previous powers of x_1 , modulo the ideal. The first power of x_1 we can write as a linear combination of previous powers belongs to the bordering of the Gröbner basis. That x_1 gets a special treatment follows from the Shape Lemma. Typically, the lexicographical Gröbner basis will have one high degree polynomial in x_1 , while the other variables x_i , $i > 1$, can be expressed as polynomials in x_1 .

Algorithm 4.1 The FGLM Algorithm

Input: $g_{>}$ a Gröbner basis for any monomial order $>$.

Output: $g_{>\text{lex}}$ a Gröbner basis for lexicographic order $>\text{lex}$.

```

 $\mathcal{N}_{>} := \text{Normal\_Set}(g_{>});$ 
 $g_{>\text{lex}} := \emptyset;$ 
 $\mathcal{N}_{>\text{lex}} := \emptyset;$ 
for  $i$  from 0 do  $\#\mathcal{N}_{>}$  do
  reduce  $x_1^i \rightarrow_{g_{>}} r$ ;
  if  $r$  is not a linear combination of the monomials in  $\mathcal{N}_{>\text{lex}}$ 
  then  $\mathcal{N}_{>\text{lex}} := \mathcal{N}_{>\text{lex}} \cup \{x_1^i\}$ ;
  else  $g_{>\text{lex}} := g_{>\text{lex}} \cup \{x_1^i - r\}$ ;
  end if;
end for;
for  $i$  from 2 to  $n$  do
  reduce  $x_i \rightarrow_{g_{>}} r$ ;
   $g_{>\text{lex}} := g_{>\text{lex}} \cup \{r\}$ ;
end for.
```

That the FGLM Algorithm is an algorithm in the proper sense follows from the following arguments.

The code in Algorithm 4.1 terminates because $\#V(I) < \infty$. The number of elements of any normal set for a zero dimensional ideal equals the number of solutions. Because there are finitely many solutions, the loop that rewrites the powers of x_1 must terminate.

Algorithm 4.1 is correct because $g_{>}$ is a Gröbner basis for I . Given a Gröbner basis $g_{>}$, every polynomial h has a unique normal form modulo the ideal. The FGLM Algorithm takes for h equal to powers of x_1 and the remaining variables x_i , $i > 1$.

The algorithm is described in greater detail in [3], accompanied with discussions about its cost. Examples can be found in [2]. The description of the FGLM Algorithm in [5, §29.2] makes the relation with Gaussian elimination more explicit.

5 The Complexity of Gröbner Bases

This section collects three theorems on the complexity of Gröbner bases and the cost of converting between different term orders. These complexity results have implications on polynomial system solving. To determine the number of solutions, we use a term order which leads to Gröbner bases of lower degrees.

In [4], we find the following theorem:

Theorem 5.1 *Let $I = \langle f_1, f_2, \dots, f_N \rangle$. In most cases, every Gröbner basis for the lexicographical ordering contains a polynomial of degree $\deg(f_1) \times \deg(f_2) \times \dots \times \deg(f_N)$.*

This theorem follows primarily from the theorem of Bézout. One very particular example is obtained via the Shape lemma. For the Shape Lemma we have as many Lagrange polynomials as the number of solutions, therefore the size will be $O(d^{n^2})$, where d is the maximal degree of the polynomials in f . Because of the *in most cases* of Theorem 5.1, the formulation of the lexicographic Gröbner basis is just as bad as the particular case of the Shape Lemma.

Restricting the set of polynomials a bit further and assuming conditions on the solutions at infinity, from [4], we deduce (simplify into a more specific version) the following theorem:

Theorem 5.2 *Let $I = \langle f_1, f_2, \dots, f_n \rangle$ where n is also the number of variables. Assume that both $V(I)$ and the solutions at infinity are finite in number. Then the polynomials of every minimal reduced Gröbner basis with respect to the degree reverse lexicographic order have degree bounded by $\deg(f_1) + \deg(f_2) + \dots + \deg(f_n) - n + 2$.*

The theorem applies to homogeneous ideals (i.e.: after a transformation to projective coordinates) and assumes that the linear equation for the hyperplane at infinity is sufficiently generic. Observe that the bound on the degrees has changed from a product into a sum. This leads to a complexity of $O(d^n)$ and the reductions in complexity are also noticed in practice [3].

The cost analysis of the FGLM Algorithm in [3] leads to the following result:

Theorem 5.3 *Denote by n the number of variables and D equals the number of common zeroes (counted with multiplicities). If the basis field operations need unit time, finding a new basis requires $O(nD^3)$ field operations.*

The theorem applies to the case where the growth of the coefficients is not taken into account.

The third power of D is not unexpected as we perform Gaussian elimination in the quotient ring. For D equal to the product of the degrees (the expected number of solutions according to the theorem of Bézout), we again arrive at the bad complexity of Theorem 5.1. However in most applications, the number of solutions D is much less than d^n . The number D is determined via a Gröbner basis of a lower complexity. Only to represent the solutions (via the Shape Lemma) we would use the Gröbner basis representation with a more expensive complexity.

Instead of the formal conversion to a lexicographic Gröbner basis, the alternative, more numerical approach, is to set up and solve the eigenvalue problems defined by the multiplication matrices. Also here, by Theorem 5.2, the dimension of the eigenvalue problems is computed via a Gröbner basis of lesser complexity.

6 Exercises

1. Consider as given a set of monomials representing $\text{LT}(g_{>})$, for $g_{>}$ a Gröbner basis with term order $>$. Write an algorithm to compute the normal set $\mathcal{N}_{>}$. Have the algorithm report an error message when it turns that the ideal $\langle g \rangle$ is not zero dimensional.
2. Describe the permutation symmetry of the system cyclic 5-roots. How many elements does the symmetry group have? What are the generators of the symmetry group? How many generators of the solution set do you expect?
3. Consider the cyclic n -roots problem for general n and compute the product D of the degrees. Compare D to the finite numbers $\#V$ in the table.
4. Consider the following modification (suggested in [3]) of the cyclic 5-roots problem:

$$f(\mathbf{x}) = \begin{cases} x_1 + x_2 + x_3 + x_4 + x_5 = 0 \\ x_1x_2 + x_2x_3 + x_3x_4 + x_4x_5 + x_5x_1 = 0 \\ x_1x_2x_3 + x_2x_3x_4 + x_3x_4x_5 + x_4x_5x_1 + x_5x_1x_2 = 0 \\ x_2x_3x_4 + x_2x_3x_4x_5 + x_3x_4x_5x_1 + x_4x_5x_1x_2 + x_5x_1x_2x_3 = 0 \\ x_1x_2x_3x_4x_5 - 1 = 0. \end{cases} \quad (7)$$

where the monomial $x_1x_2x_3x_4$ in the original cyclic 5-roots system is replaced by $x_2x_3x_4$. Compute a Gröbner basis with the graded lexicographical order to determine the number of roots of this modified cyclic 5-roots system. Use the FGLM method to convert the Gröbner basis to a pure lexicographical term order. Compare the size of the coefficients between the bases for the different term orders.

5. A Gröbner basis with respect to $>_{\text{grevlex}}$ is given in [2] by $g = \langle g_1, g_2, g_3, g_4 \rangle$, with

$$\begin{aligned} g_1 &= x_3^4 - 3x_3^3 - 4x_2x_3 + 2x_3^2 - x_2 + 2x_3 - 2, \\ g_2 &= x_2x_3^2 + 2x_2x_3 - 2x_3^2 + 1, \\ g_3 &= x_2^2 - 2x_2x_3 + x_3^2 - x_3, \\ g_4 &= x_1 + x_2 + x_3. \end{aligned} \quad (8)$$

which can be computed in Maple via $\mathbf{g} := \text{Groebner}[\text{Basis}](\mathbf{f}, \text{grevlex}(\mathbf{x}, \mathbf{y}, \mathbf{z}))$; where \mathbf{f} defines the ideal $I = \langle x_1x_2 + x_3 - x_1x_3, x_1^2 - x_3, 2x_1^3 - x_1^2x_2x_3 - 1 \rangle$.

- (a) What is $\mathcal{N}_{>_{\text{grevlex}}}$?
 - (b) Use the FGLM Algorithm to compute $g_{>_{\text{lex}}}$.
6. Use `Groebner[FGLM]` in Maple on the previous exercise. Alternatively, use the `fglm` command of Singular, as available in Sage. In both cases – whether you use Maple or Sage – verify whether the output of the software corresponds to the manual computations of the last exercise.
 7. Translate the pseudo code of our description of the FGLM algorithm into the language of a computer algebra system, Macaulay 2, Singular, or Maple.

References

- [1] G. Björck. Functions of modulus one on Z_p whose Fourier transforms have constant modulus. In *Proceedings of the Alfred Haar Memorial Conference, Budapest*, volume 49 of *Colloquia Mathematica Societatis János Bolyai*, pages 193–197, 1985.
- [2] D. Cox, J. Little, and D. O’Shea. *Using Algebraic Geometry*, volume 185 of *Graduate Texts in Mathematics*. Springer-Verlag, 1998.

- [3] J.C. Faugère, P. Gianni, D. Lazard, and T. Mora. Efficient computation of zero-dimensional Gröbner bases by change of ordering. *Journal of Symbolic Computation*, 16(4):329–344, 1993.
- [4] D. Lazard. Gröbner bases, Gaussian elimination and resolution of systems of algebraic equations. In J.A. van Hulzen, editor, *Computer Algebra. EUROCAL'83, European Computer Algebra Conference. London, England, March 1983*, volume 162 of *Lecture Notes in Computer Science*, pages 146–156. Springer-Verlag, 1983.
- [5] T. Mora. *Solving Polynomial Equation Systems II: Macaulay's Paradigm and Gröbner Technology*, volume 99 of *Encyclopedia of Mathematics and Its Applications*. Cambridge University Press, 2005.