

## Gröbner bases

In this lecture we introduce Buchberger's algorithm to compute a Gröbner basis for an ideal, following [2]. We sketch an application in filter design. Showing the termination of Buchberger's algorithm proves the Hilbert basis theorem. We also show that with a pure lexicographic term order, the Gröbner basis puts a given system into triangular form.

### 1 S-polynomials

We denote the leading term of a polynomial  $f$  by  $\text{LT}(f)$ . A set of polynomials  $g$  is a Gröbner basis for an ideal  $I$  if  $I = \langle g \rangle$  and the leading terms of  $g$  generate the ideal of leading terms of the polynomials in  $I$ , i.e.:  $\langle \text{LT}(g) \rangle = \langle \text{LT}(I) \rangle$ .

Given any Gröbner basis, the division algorithm decides the ideal membership problem.

A term is the product of a coefficient and a monomial. By  $\text{LM}(p)$  we denote the leading monomial of a polynomial  $p$ . The least common multiple of two monomials  $\mathbf{x}^{\mathbf{a}}$  and  $\mathbf{x}^{\mathbf{b}}$  is denoted by  $\text{LCM}(\mathbf{x}^{\mathbf{a}}, \mathbf{x}^{\mathbf{b}})$ .

To eliminate the leading term of two nonzero polynomials  $p$  and  $q$ , we construct its  $S$ -polynomial ( $S$  stands for Subtraction) as follows:

$$S(p, q) = \frac{\text{LCM}(\text{LM}(p), \text{LM}(q))}{\text{LT}(p)} \cdot p - \frac{\text{LCM}(\text{LM}(p), \text{LM}(q))}{\text{LT}(q)} \cdot q. \quad (1)$$

If  $p$  and  $q$  belong to the same ideal  $I$ , then  $S(p, q) \in I$ .

The use of  $S$ -polynomials to eliminate leading terms of multivariate polynomials generalizes the row reduction algorithm for systems of linear equations. If we take a system of homogeneous linear equations (i.e.: the constant coefficient equals zero), then it is not hard to see that bringing the system in triangular form yields a Gröbner basis for the system.

**Theorem 1.1 (Buchberger's criterion)** *A set  $g = \{g_1, g_2, \dots, g_s\}$  is a Gröbner basis if and only if for all pairs  $g_i$  and  $g_j$ ,  $i \neq j$ , the remainder of the division of  $S(g_i, g_j)$  by  $g$  equals zero.*

Buchberger's criterion leads to an algorithm to compute a Gröbner basis.

**Algorithm 1.1** The Buchberger algorithm to compute a Gröbner basis.

Input:  $f = \{f_1, f_2, \dots, f_N\}$ ,  $I = \langle f \rangle$ .

Output:  $g = \{g_1, g_2, \dots, g_s\}$ ,  $\langle \text{LT}(I) \rangle = \langle \text{LT}(g) \rangle$ .

```

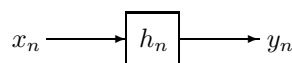
g := f;
repeat
  h := g;
  for each pair (p, q), p ≠ q, p, q ∈ g do
    S := S(p, q);
    r := remainder of S after division by g;
    if r ≠ 0
      then g := g ∪ {S};
    end if;
  end for;
until g = h.
```

A Gröbner basis  $g$  is called reduced if the leading coefficient of every polynomial in  $g$  is one and, moreover, for all  $p \in g$ , no monomial of  $p$  lies in  $\langle \text{LT}(g \setminus \{p\}) \rangle$ . A nice property is that, for any fixed monomial ordering, any nonzero ideal  $I$  has a unique reduced Gröbner basis. We may apply this property to decide whether two given sets of polynomials generate the same ideal.

## 2 Wavelet Design

Algorithms to compute a Gröbner basis provide effective methods to construct wavelet filters. Following [4], the design equations for the filters lead to polynomial systems.

The input of a filter is a discrete time signal  $\{x_n\}$ . The output of the filter  $\{y_n\}$  is completely determined by  $\{h_n\}$ , the impulse response. Schematically we draw



With the convolution operator  $*$ , we compute  $y = h * x = \sum_k x_k h_{n-k}$ . Notice that a convolution corresponds to the multiplication of two univariate polynomials (conv is the MATLAB command to multiply the coefficient vectors of two polynomials).

Via the  $Z$ -transform,  $Z(\{x_n\}) = \sum_n x_n z^{-n}$ , the convolution is transformed into a componentwise product:  $Y(z) = H(z)X(z)$ , with  $X(z) = Z(\{x_n\})$ ,  $H(z) = Z(\{h_n\})$ , and  $Y(z) = Z(\{y_n\})$ . The function  $H(z)$  is called the transfer function of the filter.

Following [4], we may wish to design a filter which realizes the following conditions:

1.  $h_2 = h_3, h_1 = h_4$ ;
2.  $(z + 1)^2$  divides  $H(z)$ ;
3.  $\sum_n h_n h_{n-2k} = \delta(k)$ , with  $\delta(k) = 1$  if  $k = 0$ ,  $\delta(k) = 0$  if  $k \neq 0$ .

This leads to the equations

$$\left\{ \begin{array}{l} h_0 + h_1 + h_2 + h_3 + h_4 + h_5 + h_6 + h_7 - 1 = 0 \\ h_2 h_0 + h_3 h_1 + h_4 h_2 + h_5 h_3 + h_6 h_4 + h_7 h_5 = 0 \\ h_6 h_2 + h_4 h_0 + h_5 h_1 + h_7 h_3 = 0 \\ h_6 h_0 + h_7 h_1 = 0 \\ h_0 - h_2 - 3h_4 - 5h_6 + 6h_7 + 4h_5 + 2h_3 = 0 \\ h_1 + 3h_3 + 5h_5 + 7h_7 - 6h_6 - 4h_4 - 2h_2 = 0 \\ h_2 - h_3 = 0 \\ h_1 - h_4 = 0 \end{array} \right. \quad (2)$$

Appending the equation

$$h_0 + h_1 + 2h_2 + 3h_3 + 4h_4 + 5h_5 + 6h_6 + 7h_7 - A = 0 \quad (3)$$

leads to a more compact Gröbner basis.

If we order the variables like  $h_0 > h_1 > h_2 > \dots > h_7 > A$  and use a pure lexicographic order, then the computed Gröbner basis is in triangular form.

More examples are given in [4], which also contains a short appendix on Gröbner bases.

### 3 Proof of the Buchberger criterion

We follow mainly [1], although the same outline is also found in [2]. The arguments hold for any ring of numbers, but we stick to our complex number field  $\mathbb{C}$ . Two lemmas are needed to prove the criterion. The first lemma, formalized in Lemma 3.1 below, links cancellations of leading terms to linear combinations of  $S$ -polynomials.

**Lemma 3.1** *Let  $f_1, f_2, \dots, f_N \in \mathbb{C}[\mathbf{x}]$  be such that  $\text{LM}(f_i) = \mathbf{x}^{\mathbf{a}}$ , for all  $i = 1, 2, \dots, N$ . Consider  $f = \sum_{i=1}^N \gamma_i f_i$ , for  $\gamma_i \in \mathbb{C}$ . If  $\text{LM}(f) < \mathbf{x}^{\mathbf{a}}$ , then  $f$  is a linear combination of the  $S$ -polynomials  $S(f_i, f_j)$ ,  $1 \leq i \neq j \leq N$ .*

*Proof.* Denote  $\text{LT}(f_i) = c_i \mathbf{x}^{\mathbf{a}}$  for  $c_i \in \mathbb{C} \setminus \{0\}$ . For all  $i, j$ :  $\text{LM}(f_i) = \text{LM}(f_j)$ :  $S(f_i, f_j) = \frac{1}{c_i} f_i - \frac{1}{c_j} f_j$ .

As  $\text{LM}(f) < \mathbf{x}^{\mathbf{a}}$ :  $\sum_{i=1}^N \gamma_i c_i = 0$ . Using telescoping sums in case  $N = 3$  (avoiding the dot, dot, dot):

$$f = \gamma_1 f_1 + \gamma_2 f_2 + \gamma_3 f_3 \quad (4)$$

$$= \gamma_1 c_1 \left( \frac{1}{c_1} f_1 \right) + \gamma_2 c_2 \left( \frac{1}{c_2} f_2 \right) + \gamma_3 c_3 \left( \frac{1}{c_3} f_3 \right) \quad (5)$$

$$= \gamma_1 c_1 \left( \frac{1}{c_1} f_1 - \frac{1}{c_2} f_2 \right) + (\gamma_1 c_1 + \gamma_2 c_2) \left( \frac{1}{c_2} f_2 - \frac{1}{c_3} f_3 \right) + (\gamma_1 c_1 + \gamma_2 c_2 + \gamma_3 c_3) f_3 \quad (6)$$

$$= \gamma_1 c_1 S(f_1, f_2) + (\gamma_1 c_1 + \gamma_2 c_2) S(f_2, f_3), \quad (7)$$

because  $\gamma_1 c_1 + \gamma_2 c_2 + \gamma_3 c_3 = 0$ . The extension for any  $N$  is clear.  $\square$

Our second lemma states that the division algorithm terminates with polynomials on output with leading monomials that are smaller than the leading monomial of the polynomial given on input for division. Lemma 3.2 formalizes this statement.

**Lemma 3.2** *For any  $p \in \mathbb{C}[\mathbf{x}]$  and  $f = (f_1, f_2, \dots, f_N)$ ,  $f_i \in \mathbb{C}[\mathbf{x}]$  for  $i = 1, 2, \dots, N$  on input, the division algorithm terminates yielding*

$$q_1, q_2, \dots, q_N, r \in \mathbb{C}[\mathbf{x}] : p = q_1 f_1 + q_2 f_2 + \dots + q_N f_N + r. \quad (8)$$

Moreover:

$$\text{LM}(p) = \max \left( \max_{k=1}^N (\text{LM}(q_k) \text{LM}(f_k)), \text{LM}(r) \right). \quad (9)$$

*Proof.* We first show that the division algorithm terminates. Referring to the definition of division algorithm of Lecture 4, we observe that at each stage of the algorithm we subtract from  $r$  (initialized with  $p$ ), producing a sequence of polynomials  $r_0 = p, r_1, r_2, \dots$ . To obtain  $r_{i+1}$  we subtract  $\frac{\text{LT}(r_i)}{\text{LT}(f_k)}$  from  $r_i$ , so we have:  $\text{LM}(r_{i+1}) < \text{LM}(r_i)$ , for all  $i$  in the sequence. This sequence must terminate for a monomial order  $<$  where every set of monomials has a smallest element.

To show (9), first recall that as the algorithm terminates, we have:  $\text{LM}(r) \leq \text{LM}(p)$ . To compute the  $q_i$ 's we collect terms  $\frac{\text{LT}(r)}{\text{LT}(f_k)}$  where  $\frac{\text{LT}(r)}{\text{LT}(f_k)} g_k$  cancels  $\text{LT}(r)$ . (We leave as an exercise to extend the division algorithm so that the  $q_i$ 's are in the output.) Therefore:  $\text{LM}(q_i) \text{LM}(f_i) \leq \text{LM}(p)$  and (9) follows.  $\square$

**Proof of Theorem 1.1.** The  $\Rightarrow$  of the theorem follows immediately from  $S(g_i, g_j) \in \langle g \rangle$ .

For the  $\Leftarrow$  direction, let  $f \in I$ . While we may write  $f$  in many ways, we choose this representation of  $f$ :

$$f = \sum_{i=1}^s h_i g_i, h_i \in \mathbb{C}[\mathbf{x}] \quad \text{for which} \quad \mathbf{x}^{\mathbf{a}} = \max_{i=1}^s \text{LM}(h_i g_i) \quad (10)$$

is least. If  $\text{LM}(f) = \mathbf{x}^{\mathbf{a}}$ , then  $\text{LT}(f) \in \langle \text{LT}(g) \rangle$  and we are done. Otherwise, we rewrite  $f$  into a representation

$$f = \sum_{i=1}^s \tilde{h}_i g_i, \tilde{h}_i \in \mathbb{C}[\mathbf{x}] \quad \text{where} \quad \max_{i=1}^s \text{LM}(\tilde{h}_i g_i) = \mathbf{x}^{\mathbf{a}}, \quad (11)$$

contradicting our first choice ( $\mathbf{x}^{\mathbf{a}}$  is least among all representations for  $f$ ), leaving only  $\text{LM}(f) = \mathbf{x}^{\mathbf{a}}$ .

To derive the contradiction, take  $H = \{ i \mid g_i \in g : \text{LM}(h_i g_i) = \mathbf{x}^{\mathbf{a}} \}$  and consider  $p = \sum_{i \in H} \text{LT}(h_i g_i)$ . We have  $\text{LM}(\text{LT}(h_i g_i)) = \mathbf{x}^{\mathbf{a}}$ , for all  $i \in H$  and  $\text{LM}(p) < \mathbf{x}^{\mathbf{a}}$ , because  $\text{LM}(f) < \mathbf{x}^{\mathbf{a}}$ . Application of Lemma 3.1 yields coefficients  $c_{ij} \in \mathbb{C}$ :  $p = \sum_{i,j \in H, i \neq j} c_{ij} S(\text{LT}(h_i g_i), \text{LT}(h_j g_j))$ . As  $\text{LCM}(\text{LM}(h_i g_i), \text{LM}(h_j g_j)) = \mathbf{x}^{\mathbf{a}}$ :

$$S(\text{LT}(h_i g_i), \text{LT}(h_j g_j)) = \frac{\mathbf{x}^{\mathbf{a}}}{\text{LT}(h_i g_i)} h_i g_i - \frac{\mathbf{x}^{\mathbf{a}}}{\text{LT}(h_j g_j)} h_j g_j = \frac{\mathbf{x}^{\mathbf{a}}}{\text{LT}(g_i)} g_i - \frac{\mathbf{x}^{\mathbf{a}}}{\text{LT}(g_j)} g_j = \frac{\mathbf{x}^{\mathbf{a}}}{\mathbf{x}^{\mathbf{b}}} S(g_i, g_j). \quad (12)$$

where  $\mathbf{x}^{\mathbf{b}} = \text{LCM}(\text{LT}(g_i), \text{LT}(g_j))$ . The expression for  $S(\text{LT}(h_i g_i), \text{LT}(h_j g_j))$  shows: if  $S(g_i, g_j)$  reduces to zero, then also  $S(\text{LT}(h_i g_i), \text{LT}(h_j g_j))$  reduces to zero. Collecting quotients from the division algorithm:

$$S(\text{LT}(h_i g_i), \text{LT}(h_j g_j)) = \sum_{k=1}^s \tilde{h}_{ijk} g_k \quad (13)$$

where by (9) of Lemma 3.2:

$$\max_{k=1}^s \text{LM}(\tilde{h}_{ijk} \text{LM}(g_k)) = \text{LM}(S(\text{LT}(h_i g_i), \text{LT}(h_j g_j))) < \max(\text{LM}(h_i g_i), \text{LM}(h_j g_j)) = \mathbf{x}^{\mathbf{a}}. \quad (14)$$

Substituting the expressions back into  $g$  we get the representation for  $f$  which gives the contradiction.  $\square$

## 4 Using Macaulay 2

Almost all computer algebra system provide functionality to compute Gröbner bases. Below is a small example with Macaulay 2, running from the command prompt \$.

```
$ M2
Macaulay2, version 1.3.1
with packages: ConwayPolynomials, Elimination, IntegralClosure, LLLBases,
               PrimaryDecomposition, ReesAlgebra, SchurRings, TangentCone
```

```
i1 : R = QQ[x,y,MonomialOrder => Lex];
```

```
i2 : I = ideal(x^2 + 1, x*y - 1);
```

```
o2 : Ideal of R
```

```
i3 : G = gens gb I
```

```
o3 = | y^2+1 x+y |
```

We see that  $\{y^2 + 1, x + y\}$  is a Gröbner basis for the ideal  $\langle x^2 + 1, xy - 1 \rangle$  with the lexicographical order.

## 5 Termination and Elimination

Showing that this algorithm terminates also shows the Hilbert basis theorem, i.e.: any ideal has a finite basis. The key observation is that as long as the repeat loop does not terminate, we augment  $g$  with a nonzero polynomial  $S = S(p, q)$  for which  $\text{LM}(S) < \text{LM}(p)$  and  $\text{LM}(S) < \text{LM}(q)$ , with respect to the term order  $<$ . Compared to  $h$ , we thus have that  $\langle \text{LT}(h) \rangle \subset \langle \text{LT}(g) \rangle$ . So as long as the loop runs, we create a chain of monomial ideals which cannot stretch for ever.

Consider again a system of homogeneous linear equations. Applying row reduction to bring such a system into triangular form can be written in terms of taking  $S$ -polynomials.

For an ideal  $I$  in  $\mathbb{C}[\mathbf{x}]$ ,  $\mathbf{x} = (x_1, x_2, \dots, x_n)$ , the  $k$ th elimination ideal is  $I_k = I \cap \mathbb{C}[x_{k+1}, \dots, x_n]$ . So  $I_k$  consists of all polynomials in  $I$  for which the first  $k$  variables have been eliminated.

**Theorem 5.1 (The Elimination Theorem)** *Let  $g$  be a Gröbner basis for an ideal  $I$  with respect to the pure lexicographical order  $x_1 > x_2 > \dots > x_n$ . Then the set  $g_k = g \cap \mathbb{C}[x_{k+1}, \dots, x_n]$  is a Gröbner basis of the  $k$ th elimination ideal  $I_k$ .*

*Proof.* To prove this theorem, we must show that  $\langle \text{LT}(I_k) \rangle = \langle \text{LT}(g_k) \rangle$ . By construction,  $\langle \text{LT}(g_k) \rangle \subset \langle \text{LT}(I_k) \rangle$ , so what remains to show is that  $\langle \text{LT}(I_k) \rangle \subset \langle \text{LT}(g_k) \rangle$ . For any  $f \in I_k$ , we must then show that  $\text{LT}(f)$  is divisible by  $\text{LT}(p)$  for some  $p \in g_k$ .

As  $f \in I$ , we have that  $\text{LT}(f)$  is divisible by  $\text{LT}(p)$  for some  $p \in g$ . Since  $f \in I_k$ , the only variables occurring in  $f$  are  $x_{k+1}, \dots, x_n$ . Because of the lexicographic order, we have that if  $\text{LT}(p) \in \mathbb{C}[x_{k+1}, \dots, x_n]$  then all other terms of  $p$  also belong to  $\mathbb{C}[x_{k+1}, \dots, x_n]$ . Thus the  $p$  for which  $\text{LT}(p)$  divides  $\text{LT}(f)$  belongs to  $g_k$ .  $\square$

While the elimination property of a Gröbner basis computed with lexicographic term order seems most desirable from a practical point of view, the size of the coefficients in such a Gröbner basis often grows so large that exact and extended multiprecision arithmetic must be used.

In [6, Chapter 13], complexity bounds for Gröbner bases are discussed. For polynomials in  $n$  variables with degrees bounded by  $d$ , the bound on the size of a Gröbner basis are essentially doubly exponential, of the form  $d^{2^\beta n}$ , where  $\beta \approx 0.5$ . Despite these pessimistic bounds, faster computers and better software have made Gröbner bases the driving engine for many algorithms in computational algebraic geometry. See for example [3] for a perspective.

## 6 Exercises

1. Solve the system (2). Use Maple or Sage to create a lexicographical Gröbner basis. Verify that by adding (3), the resulting Gröbner basis is more compact. How many real solutions do you find?
2. Apply Buchberger's algorithm by hand (you can use a Maple worksheet to compute all  $S$ -polynomials) to the ideal generated by the equations  $\{x_1^2 + x_2^2 - 1, x_1x_2 - 1\}$  using a pure lexicographical monomial order.
3. Show that for two systems  $f(\mathbf{x}) = \mathbf{0}$  and  $g(\mathbf{x}) = \mathbf{0}$ : if  $\langle f \rangle = \langle g \rangle$ , then their solutions are the same. Give an example of a case for which the opposite direction does hold.
4. Consider the example (taken from [5, page 323]):

$$f(x, y) = \begin{cases} x^2 + \epsilon xy + y^2 - 1 = 0 \\ y^3 - 3x^2y = 0 \end{cases} \quad \text{for } \epsilon \approx 0.$$

Although the solution set varies continuously with  $\epsilon$ , we will verify that a Gröbner basis cannot be a continuous function of  $\epsilon$ . Use Maple or Sage for the following calculations:

- (a) Make a plot of the two curves defined by the polynomials in the system. Justify why all intersection points are well conditioned roots.
  - (b) Compute Gröbner bases for various values of  $\epsilon$  and examine the growth of the coefficients as  $\epsilon$  gets smaller.
  - (c) Compute a Gröbner basis where  $\epsilon$  is a parameter. Interpret the results.
5. The twisted cubic is a curve in 3-space defined by  $(x_1 = t, x_2 = t^2, x_3 = t^3)$ , for a parameter  $t$ . Equivalently, the equations  $x_1^2 - x_2 = 0$  and  $x_1^3 - x_3 = 0$  defined the twisted cubic in implicit form. The surface of all lines tangent to points on the twisted cubic is

$$\begin{cases} x_1 = t + s \\ x_2 = t^2 + 2ts \\ x_3 = t^3 + 3t^2s, \end{cases} \quad (15)$$

for parameters  $s$  and  $t$ . Compute a lexicographical Gröbner basis using a monomial order that eliminates  $s$  and  $t$ . Find an equation for the surface that defines all tangent lines to the twisted cubic.

6. With a lexicographical Gröbner basis and a solver for polynomials in one variable we can solve zero dimensional polynomial systems, systems that have only isolated solutions. Write a procedure in a computer algebra system that takes on input a lexicographical Gröbner basis and computes all solutions by applying the univariate solver repeatedly and substituting the solutions into the remaining equations. For a numerical solver, show that the working precision must be sufficiently high enough as the solver progresses, considering the example of exercise 4.

## References

- [1] W.W. Adams and P. Loustaunau. *An Introduction to Gröbner Bases*, volume 3 of *Graduate Studies in Mathematics*. AMS, 1994.
- [2] D. Cox, J. Little, and D. O’Shea. *Ideals, Varieties and Algorithms. An Introduction to Computational Algebraic Geometry and Commutative Algebra*. Undergraduate Texts in Mathematics. Springer-Verlag, second edition, 1997.
- [3] D. Lazard. Thirty years of polynomial system solving, and now? *Journal of Symbolic Computation*, 44(3):222–231, 2009.
- [4] J. Lebrun and I. Selesnick. Gröbner bases and wavelet design. *Journal of Symbolic Computation*, 37(2):227–259, 2004. Special issue on computer algebra and signal processing, edited by J.R. Johnson, J.M.F. Moura, M. Püschel and D. Rockmore.
- [5] H.J. Stetter. *Numerical Polynomial Algebra*. SIAM, 2004.
- [6] C.K. Yap. *Fundamental Problems of Algorithmic Algebra*. Oxford University Press, 2000.