*J. Symbolic Computation* (2000) **30**, 451–467
doi:10.1006/jsco.1999.0416
Available online at http://www.idealibrary.com on IDEAL

# A Fast Algorithm for Gröbner Basis Conversion and its Applications

QUOC-NAM TRAN

*Department of Computer Science, Lamar University, TX 77710, U.S.A.*

The Gröbner walk method converts a Gröbner basis by partitioning the computation of the basis into several smaller computations following a path in the Gröbner fan of the ideal generated by the system of equations. The method works with ideals of zero-dimension as well as positive dimension. Typically, the target point of the walking path lies on the intersection of very many cones, which ends up with initial forms of a considerable number of terms. Therefore, it is crucial to the performance of the conversion to change the target point since we have to compute a Gröbner basis with respect to the elimination term order of such large initial forms.

In contrast to heuristic methods found in the literature, in this paper the author presents a deterministic method to vary the target point in order to ensure the generality of the position, i.e. we always have just a few terms in the initial forms. It turns out that this theoretical result brings a dramatic speed-up in practice. We have implemented the Gröbner walk method together with the deterministic method for varying the target point in the kernel of Mathematica. Our experiments show the superlative performance of our improved Gröbner walk method in comparison with other known methods. Our best performance is $3 \times 10^4$ times faster than the direct computation of the reduced Gröbner basis with respect to pure lexicographic term order (using the Buchberger algorithm and the sugar cube strategy). We also discuss the complexity of the conversion algorithm and prove a degree bound for polynomials in the target Gröbner basis.

In the second part of the paper, we present some applications of the conversion method for implicitization and geometric reasoning. We compare the efficiency of the improved Gröbner walk method with other methods for elimination such as multivariate resultant methods.



## 1. Introduction

The method of Gröbner bases (Buchberger, 1965, 1985) is one of the main tools for eliminating variables and solving systems of nonlinear algebraic equations. In order to eliminate variables or to solve such a system, one has to compute a Gröbner basis with respect to an elimination term order, for example pure lexicographic term order. However, it is time and memory consuming to do so directly.

One way to overcome these difficulties is to partition the computation of the Gröbner basis into several smaller computations following a path in the Gröbner fan of the ideal generated by the system of equations. This approach of Collart *et al.* (1997) is called the "Gröbner walk" method, and does not require any assumption about the dimension of the ideal.

A crucial parameter of the performance of the Gröbner walk method is the choice of the walking path since the number of conversion steps and the complexity of each step

0747–7171/00/100451 + 17 $35.00/0 

depend heavily on it. Ideally, the walking path should be free of the intersections of several cones since in this general position, the initial forms involve far fewer terms; therefore, the transformations can be computed cheaply. As suggested in Collart *et al.* (1997) and improved in Amrhein *et al.* (1996) and Amrhein and Gloor (1998), it is appropriate to vary the starting point in order to ensure the generality of the position.

However, the real difficulty comes from the target point, where one has to perform the last conversion with respect to the elimination term order but does not know how to vary the point. Typically, the target point lies on the intersection of very many cones, which ends up with initial forms of a considerable number of terms. (We have many examples whose initial forms have a few hundred terms.) Therefore, it increases the complexity of the conversion since we have to compute a Gröbner basis with respect to the elimination term order of such large initial forms. Amrhein and Gloor (1998) offer a heuristic way to guess a perturbed target point and check the validity after the conversion. But, if the heuristic guess fails then one has to compute the Gröbner basis with respect to the elimination term order of such large initial forms anyway.

In this paper, the author gives a deterministic method to vary the target point in order to ensure the generality of the position, i.e. we always have just a few terms in the initial forms. The main idea of the method is that, even though we do not know the Gröbner basis with respect to the target cone, we know in advance how large the polynomials in the Gröbner basis can be. More precisely, we use the upper degree bound for polynomials in the Gröbner basis (Bayer, 1982; Dubé, 1990) for the variation of the target point.

It turns out that this theoretical result brings a dramatic speed-up in practice. We have implemented the Gröbner walk method together with the deterministic method for varying the target point in the kernel of Mathematica. Our experiments show the superlative performance of our improved Gröbner walk method in comparison with other known methods. Our best performance is $3 \times 10^4$ times faster than the direct computation of the reduced Gröbner basis with respect to pure lexicographic term order (using the Buchberger algorithm and the sugar cube strategy). In Section 3.2 we discuss the complexity of the conversion algorithm. We prove that the degree of the polynomials in the target Gröbner basis is bounded by

$$2^{2^w-1}d^{2^w} + 2^{2^{w+1}}d(n+1)(d+1)^{2^w-2}(n+2)^{2^{w-1}-1},$$

where $w$ is the length of the Gröbner walk.

Other approaches for basis conversion are the FGLM method for zero-dimensional systems (Faugère *et al.*, 1993) and the method based on the Hilbert-Pointcaré series (Traverso, 1996). However, in this paper we concentrate on the Gröbner walk method.

In the second part of the paper, we present some applications of the conversion method for implicitization and geometric reasoning. We compare the efficiency of the improved Gröbner walk method with other very new methods for elimination such as multivariate resultant methods.

## 2. Preliminaries

### 2.1. A degree bound

Recall that we have the following upper degree bound for polynomials in a Gröbner basis with respect to any admissible term order.

Lemma 2.1. (Dubé, 1990) *Let $\mathbb{K}[x_1, \ldots, x_n]$ be a ring of multivariate polynomials with coefficients in a field $\mathbb{K}$, and let $F$ be a subset of this ring such that $d$ is the maximum total degree of any polynomial in $F$. Then for any admissible term order, the total degree of polynomials in a Gröbner basis for the ideal generated by $F$ is bounded by*

$$(d^2 + 2d)^{2^{n-1}}.$$

When the ideal is zero-dimensional, Caniglia *et al.* (1991) showed that we can even lower the degree bound to $d^{\mathrm{O}(n)}$.

## 2.2. Gröbner walk

In order to eliminate variables or to solve a system of nonlinear algebraic equations, one has to compute a Gröbner basis with respect to an elimination term order, for example pure lexicographic term order. However, it is time and memory consuming to do so directly. From the complexity point of view as well as the practical point of view, it is more efficient and requires much less memory to compute a Gröbner basis with respect to the degree reverse lexicographic term order in comparison with elimination term orders. Moreover, in some instances such as in implicitization of surfaces with polynomial parametric form, one knows in advance that the given set of polynomials is already a Gröbner basis with respect to some term orders. Therefore, it would be more efficient if one knew how to convert the known Gröbner basis to a Gröbner basis of the ideal with respect to an elimination term order.

The Gröbner walk method converts a Gröbner basis by partitioning the computation of the basis into several smaller computations following a path in the Gröbner fan of the ideal generated by the system of equations. The method works with ideals of zero-dimension as well as positive dimension.

We now give a short introduction to basic facts and analyze the performance of the traditional algorithms for the Gröbner walk method. We refer to Collart *et al.* (1997) for missing details.

Given the reduced Gröbner basis of an ideal $I \subset \mathbb{K}[x_1, \ldots, x_n]$ with respect to an admissible term order $\prec_1$, where $\mathbb{K}$ is a computable field, our goal is to compute the reduced Gröbner basis of $I$ with respect to another admissible term order $\prec_2$ without applying Buchberger's algorithm.

The set of terms in the variables $x_1, x_2, \ldots, x_n$ is denoted by $\mathcal{T}^n$. The set $\Omega = \{(\phi_1, \ldots, \phi_n) \in \mathbb{Q}^n : \phi_j \geq 0, \forall 1 \leq j \leq n\}$ is called the set of weight vectors. Let $\omega = (w_1, \ldots, w_n) \in \Omega$; for a monomial $t = cx_1^{i_1}x_2^{i_2}\cdots x_n^{i_n}$, we denote its $\omega$-degree by $\deg_\omega(t) = \sum_{j=1}^{n} i_j w_j$. The $\omega$-degree of a nonzero polynomial $f$, denoted $\deg_\omega(f)$, is the maximum of the $\omega$-degrees of the monomials which occur in $f$ with nonzero coefficients. The initial form of $f$ with respect to $\omega$, denoted $in_\omega(f)$, is the sum of all those monomials in $f$ with maximal $\omega$-degree. Furthermore, $\deg_\omega(0) = -1$ and $in_\omega(0) = 0$. We say that $\prec$ refines $\omega$ if $\deg_\omega(t_1) < \deg_\omega(t_2)$ implies $t_1 \prec t_2$ for all $t_1, t_2 \in \mathcal{T}^n$. We say that $\omega$ represents $\prec$ if $\langle I_\omega \rangle = \langle I_\prec \rangle$, where $I_\omega = \{in_\omega(f) : f \in I\}$ and $I_\prec = \{in_\prec(f) : f \in I\}$.

Lemma 2.2. (Eisenbud, 1995) *Let $I$ be an ideal, $\prec$ a term order and $G$ the reduced Gröbner basis of $I$ with respect to $\prec$. A weight vector $\omega$ represents $\prec$ for $I$ if and only if $in_\omega(g) = in_\prec(g), \forall g \in G$.*

For the term orders $\prec$ and the weight vector $\omega$, we define the term order $(\omega|\prec)$ by

$$t_1(\omega|\prec)t_2 \text{ if } deg_\omega(t_1) < deg_\omega(t_2) \text{ or } deg_\omega(t_1) = deg_\omega(t_2) \text{ and } t_1 \prec t_2.$$

The topological closure in $\mathbb{Q}^n$ of $\{w \in \Omega : \langle I_\prec \rangle = \langle I_\omega \rangle\}$, which is a convex, polyhedral cone in $\mathbb{Q}^n$ with a non-empty interior, is called the Gröbner cone of $I$ with respect to $\prec$.

The term orders $\prec_1$ and $\prec_2$ can be expressed by sequences of rational weight vectors $S_{\prec_1}$ and $S_{\prec_2}$ where their first elements, denoted by $\sigma$ and $\tau$, are weight vectors refined by $\prec_1$ and $\prec_2$, respectively. For $\sigma$ and $\tau$ in the set of weight vectors $\Omega$, we denote the line segment in $\Omega$ between $\sigma$ and $\tau$ by $\overline{\sigma\tau}$, i.e.

$$\overline{\sigma\tau} = \{(1-t)\sigma + t\tau : 0 \le t \le 1\}.$$

There exist finitely many weight vectors $\sigma = \omega_0, \omega_1, \ldots, \omega_m = \tau$ in $\overline{\sigma\tau}$ and pairwise different Gröbner cones $C_{\prec_1}(I) = C_0(I), C_1(I) = C_{(\omega_1|\prec_2)}(I), \ldots, C_m(I) = C_{\prec_2}(I)$ in the Gröbner fan of $I$ such that for every $k \in \{1, \ldots, m\}$, $\omega_k$ is the weight vector with

$$\overline{\omega_{k-1}\omega_k} = \overline{\omega_{k-1}\tau} \cap C_{k-1}(I).$$

We denote the reduced Gröbner basis of $I$ over the Gröbner cone $C_k(I)$ by $G_k$.

We perform the Gröbner walk method by moving on the line segment $\overline{\sigma\tau}$ from $\sigma$ to $\tau$, i.e. we compute $\omega_1, \ldots, \omega_{m-1}$ and $G_1, \ldots, G_m$ successively. The crucial point is that this conversion can be done efficiently without applying Buchberger's algorithm. We first check if $C_{k-1}(I)$ is equal to $C_{\prec_2}(I)$ for a given Gröbner basis $G_{k-1} = \{g_1, \ldots, g_r\}$. If so then $G_{k-1}$ is already the reduced Gröbner basis of $I$ with respect to $\prec_2$. Otherwise, we have to determine the next weight vector $\omega_k$, which is the point on the segment $\overline{\sigma\tau}$ where we leave the Gröbner cone $C_{k-1}(I)$. The weight $\omega_k$ can be easily computed from $\omega_{k-1}$, $\tau$ and $G_{k-1}$ as $\omega_k = \omega(\bar{t}) = \omega_{k-1} + \bar{t}(\tau - \omega_{k-1})$, where

$$\begin{aligned}\bar{t} = \min(\{t \in \mathbb{Q} \cap (0,1] : \deg_{\omega(t)} p_1 = \deg_{\omega(t)} p_i,\\ \text{for some } g = p_1 + \cdots + p_n \in G_{k-1}, 2 \le i \le n\}).\end{aligned} \tag{2.1}$$

After leaving $C_{k-1}(I)$ we enter $C_k = C_{(\omega_k|\prec_2)}(I)$. We now have to transform $G_{k-1}$ to $G_k$. Note that there exists a term order $\prec$ which refines $\omega_k$ such that $C_\prec(I) = C_{k-1}(I)$. Therefore $in_\prec(f) = in_\prec(in_{\omega_k}(f))$ for all $f \in I$ and

$$\langle\langle I_{\omega_k}\rangle_\prec\rangle = \langle I_\prec\rangle = \langle (G_{k-1})_\prec\rangle = \langle(((G_{k-1})_{\omega_k})_\prec\rangle$$

hence $(G_{k-1})_{\omega_k}$ is the reduced Gröbner basis of $I_{\omega_k}$ with respect to $\prec$. We now convert $(G_{k-1})_{\omega_k}$ to the the reduced Gröbner basis $M = \{m_1, \ldots, m_s\}$ of $\langle I_{\omega_k}\rangle$ with respect to $(\omega_k|\prec_2)$. Note that this conversion itself can be done with any basic conversion, for example by using the Hilbert–Poincaré series (Traverso, 1996) or by recursive use of the Gröbner walk method. However, we may want to use a specialized Buchberger's algorithm in this case since we can perturb the weight vectors such that most of the initials are monomial. Critical pairs of two monomials are unnecessary since its S-polynomial is always zero. As most of the S-polynomials reduce to zero in one step, there is no need of sophisticated selection strategies.

Since $m_1, \ldots, m_s$ are $\omega_k$-homogeneous, we can compute $\omega_k$-homogeneous polynomials $h_{i1}, \ldots, h_{ir}$ with

$$m_i = \sum_{j=1}^{r} h_{ij} in_{\omega_k}(g_j) \qquad \text{and} \qquad \deg_{\omega_k}(m_i) = \deg_{\omega_k}(h_{ij} in_{\omega_k}(g_j)),$$

for $j = 1 \dots r$ with $h_{ij} \neq 0$. Replacing $in_{\omega_k}(g_j)$ by $g_j$, we obtain

$$f_i = \sum_{j=1}^{r} h_{ij} g_j \qquad \text{and} \qquad G = \{f_1, \dots, f_s\}.$$

It immediately follows that $in_{\omega_k}(f_i) = m_i$ and therefore

$$\langle I_{(\omega_k|\prec_2)} \rangle = \langle \langle I_{\omega_k} \rangle_{(\omega_k|\prec_2)} \rangle = \langle M_{(\omega_k|\prec_2)} \rangle = \langle G_{(\omega_k|\prec_2)} \rangle.$$

Hence $G$ is a Gröbner basis of $I$ with respect to $(\omega_k| \prec_2)$ which we reduce to $G_k$.

## 3. Main Results

### 3.1. a fast algorithm

As we have seen in the previous section, it is crucial to the Gröbner walk method to keep the initial forms as small as possible since the complexity of the method depends heavily on them. We can partly achieve this goal by varying the starting and the intermediate weight vectors to ensure that the walking path is free of the intersections of several cones. In this general position, the initial forms involve far fewer terms; therefore, the transformations can be computed cheaply. This step can be done easily since we already know the Gröbner basis of the cone.

However, the real difficulty comes from the target weight vector, where one has to perform the last conversion with respect to the elimination term order but does not know how to vary the point. Typically, the target point lies on the intersection of very many cones, which ends up with the initial forms of a considerable number of terms as in the following examples.

Example 3.1. Given the system of equations

$$5x^4 + 13y^2z + 11x^4yz^3 + 12x^2z^4 + 2x^4z^4 + 5yz^4 + 13x^3yz^4,$$
$$11xy + 15y^3 + 4x^2y^4z + 2xz^2 + 18x^2z^2 + 19x^2yz^3,$$
$$3xy + 16xz^2 + 20x^3yz^2 + 3yz^3 + 4xy^2z^3 + 2x^4y^2z^3;$$

we convert from a Gröbner basis with respect to the degree reverse lexicographic term order of the ideal generated by the system to a Gröbner basis with respect to the pure lexicographic term order determined by $x \succ y \succ z$. The ideal is one-dimensional. The initial forms at the target weight vector have as many as 136 terms.

Therefore, it increases the complexity of the conversion since we have to compute a Gröbner basis with respect to the elimination term order of such large initial forms. Amrhein and Gloor (1998) offer a heuristic way to guess a perturbed target point and check the validity after the conversion. But, if the heuristic guess fails then one has to compute the Gröbner basis with respect to the elimination term order of such large initial forms anyway.

Since in many problems the target term order is lexicographic, we first state and prove the main result for this special case.

Lemma 3.1. *For every ideal* $I \subset \mathbb{K}[x_1, \dots, x_n]$*, the weight vector* $\omega = (d^{n-1}, d^{n-2},$

$\ldots, 1)$ *represents the Gröbner cone of $I$ with respect to the lexicographic term order $\prec$, where $d$ is an upper degree bound for polynomials in a Gröbner basis with respect to $\prec$.*

PROOF. Even though we do not know the Gröbner basis of $I$ with respect to the lexicographic term order $\prec$, the existence of the Gröbner basis and the upper degree bound is clear (Buchberger, 1965; Bayer, 1982). In order to prove that $\omega$ represents the cone, we need to show that $\forall g \in G, in_\omega(g) = in_\prec(g)$, where $G$ is the reduced Gröbner basis of $I$ with respect to $\prec$. The lexicographic term order can be expressed by the sequence $(1, 0, \ldots, 0)$, $(0, 1, \ldots, 0)$, $\ldots$, $(0, 0, \ldots, 1)$ of weight vectors. For every two monomial $t_1 = c_1 x_1^{e_{1_1}} x_2^{e_{1_2}} \cdots x_n^{e_{1_n}}$ and $t_2 = c_2 x_1^{e_{2_1}} x_2^{e_{2_2}} \cdots x_n^{e_{2_n}}$ of $g$, we can assume that $t_1 \succ t_2$; there exists a number $k$, $1 \le k \le n$ such that $e_{1_k} > e_{2_k}$ and $e_{1_j} = e_{2_j}$, $1 \le j < k$. If $k = n$ then it is obvious that $\deg_\omega(t_1) > \deg_\omega(t_2)$. Since $d^{n-k} e_{1_k} \ge d^{n-k}(e_{2_k} + 1)$ and

$$\begin{aligned} d^{n-k} &> d^{n-k-1}(d-1) + d^{n-k-2}(d-1) + \cdots + (d-1) \\ &\ge d^{n-k-1} e_{2_{k-1}} + d^{n-k-2} e_{2_{k-2}} + \cdots + e_{2_n}, \end{aligned}$$

we have $d^{n-k} e_{1_k} > \sum_{j=k}^{n} d^{n-j} e_{2_j}$. Therefore $\deg_\omega(t_1) > \deg_\omega(t_2)$, and hence $in_\omega(g) = in_\prec(g)$, $\forall g \in G$. □

We now state and prove the following main result for the general case:

THEOREM 3.1. *For every ideal $I \subset \mathbb{K}[x_1, \ldots, x_n]$ and for every admissible term order, there exists a deterministic and constructive method for finding weight vectors which represent the Gröbner cone of the term order.*

PROOF. Let $\prec$ be the term order which can be expressed by the sequence

$$\begin{aligned} w_1 &= (w_{1_1}, w_{1_2}, \ldots, w_{1_n}), \\ w_2 &= (w_{2_1}, w_{2_2}, \ldots, w_{2_n}), \\ &\cdots \quad \cdots \\ w_n &= (w_{n_1}, w_{n_2}, \ldots, w_{n_n}), \end{aligned}$$

of weight vectors. Let $\mathcal{M}$ be the maximum of the absolute value of all of the elements of $w_i$, $1 \le i \le n$. Let $d$ be the product of $\mathcal{M}$ and an upper degree bound for polynomials in the reduced Gröbner basis $G$ of $I$ with respect to $\prec$. We show that $\omega = (d^{n-1} w_1 + d^{n-2} w_2 + \cdots + d w_{n-1} + w_n)$ represents the Gröbner cone of $I$ with respect to $\prec$; i.e. $\forall g \in G$, $in_\omega(g) = in_\prec(g)$. Again, we do not know the Gröbner basis of $I$ with respect to the term order $\prec$ but the existence of the Gröbner basis and the upper degree bound is clear. For every two monomial $t_1 = c_1 x_1^{e_{1_1}} x_2^{e_{1_2}} \ldots x_n^{e_{1_n}}$ and $t_2 = c_2 x_1^{e_{2_1}} x_2^{e_{2_2}} \ldots x_n^{e_{2_n}}$ of $g$, we can assume that $t_1 \succ t_2$; there exists a number $k$, $1 \le k \le n$ such that $w_k \cdot e_1 > w_k \cdot e_2$ and $w_i \cdot e_1 = w_i \cdot e_2$, $1 \le i < k$, where $e_1 = (e_{1_1}, e_{1_2}, \ldots, e_{1_n})$ and $e_2 = (e_{2_1}, e_{2_2}, \ldots, e_{2_n})$. If $k = n$, it is obvious that $\deg_\omega(t_1) > \deg_\omega(t_2)$. Since

$$w_k \cdot e_1 \ge w_k \cdot e_2 + 1,$$

$$w_i \cdot e_2 = \sum_{j=1}^{n} w_{i_j} e_{2_j} \le \mathcal{M} \sum_{j=1}^{n} e_{2_j} \le d - 1, 1 \le i \le n,$$

and

$$\sum_{i=k+1}^{n} d^{n-i} w_i \cdot e_2 \leq \sum_{i=k+1}^{n} d^{n-i}(d-1) < d^{n-k},$$

we have

$$d^{n-k} w_k \cdot e_1 > \sum_{i=k}^{n} d^{n-i} w_i \cdot e_2.$$

Therefore $\deg_\omega(t_1) > \deg_\omega(t_2)$; hence $in_\omega(g) = in_\prec(g)$, $\forall g \in G$ and we are done. □

Remark. (1) The degree bound computation is based on the original system of equations.
(2) For the starting point and the intermediate weight vectors, the degree bound is the actual highest degree of the polynomials in the known reduced Gröbner basis.
(3) Theorem 3.1 is still correct for maximal Gröbner bases.

Corollary 3.1. *For every ideal $I \subset \mathbb{K}[x_1, \ldots, x_n]$ and for every pair of two admissible term orders, there exists a deterministic and constructive method based on the upper degree bound for finding an appropriate target point; i.e. we are able to find a weight vector which represents the target cone.*

Based on the previous theorem, we propose a fast algorithm for basic conversion as follows.

Algorithm 3.1.

**Inputs**: The reduced Gröbner basis $G_1$ of an ideal $I$ with respect to an arbitrary admissible term order $\prec_1$ and a second term order $\prec_2$; the unique weight vectors $\sigma$ and $\tau$ (up to a scalar factor) refined by $\prec_1$ and $\prec_2$ respectively.
**Output**: The reduced Gröbner basis $G_2$ of $I$ with respect to $\prec_2$.

(1) $G \leftarrow G_1$.
(2) Find a new starting point if $\sigma$ lies on the intersection of several cones.
(3) `While True`
  (a) Find the next weight vector $\omega$
  (b) `If` $\omega = \tau$
    (i) `If` $G$ is also the Gröbner basis of $I$ with respect to $\prec_2$, `return` $G$.
    (ii) `Else` find a new target point if $\tau$ lies on the intersection of several cones; `continue`.
  (c) Convert the $\omega$-initial Gröbner basis to the reduced Gröbner basis with respect to $(\omega| \prec_2)$.
  (d) Lift the $\omega$-initial Gröbner basis to the full Gröbner basis with respect to $(\omega| \prec_2)$.

The termination and correctness of the algorithm are obvious from the Gröbner walk method and Theorem 3.1. Note that we can even change the path during the walk, i.e. if the intermediate weight vector $\omega$ is on the intersection of several cones, we can find a new weight vector $\omega'$ such that $\omega'$ represents $(\omega'| \prec_2)$.

### 3.2. A DEGREE BOUND FOR THE CONVERSION ALGORITHM

In this section we assume that $I$ be a homogeneous ideal in $\mathbb{K}[x_0, x_1, \ldots, x_n]$, where $x_0$ is the homogenous variable.

Let $F$ and $G$ be two adjacent reduced Gröbner bases of $I$; i.e. the intersection of the Gröbner cone of $F$ and the Gröbner cone of $G$ generates an $n$-dimensional subspace in $\mathbb{Q}^{n+1}$. It has been shown in Kalkbrener (1999) that the degree of polynomials in $G$ is bounded by

$$2d^2 + (n+1)d, \tag{3.2}$$

where $d = \max(\{\deg(f) | f \in F\})$. Additionally, Theorem 3.1 gives us a mechanism to assure that we always walk between the adjacent Gröbner cones.

Let $w$ be the length of the Gröbner walk (Algorithm 3.1) from $C_{\prec_1}(I)$ to $C_{\prec_2}(I)$. We prove the following bound.

LEMMA 3.2. *The degree of polynomials in the reduced Gröbner basis $G_2$ of $I$ with respect to $\prec_2$ is bounded by*

$$B_w = 2^{2^w - 1} d^{2^w} + 2^{2^{w+1}} d(n+1)(d+1)^{2^w - 2}(n+2)^{2^{w-1}-1}.$$

PROOF. From the recursive function (3.2), it is easy to see that the bound $B_w$ can be written in the form

$$B_w = D_w + d(n+1)F_w,$$

where $D_w = 2^{2^w - 1} d^{2^w}$.

We now prove that the $F_w$ is bounded by $2^{2^{w+1}}(d+1)^{2^w-2}(n+2)^{2^{w-1}-1}$ by induction on the length $w$ of the Gröbner walk. It is obvious for $w = 1$. Assume that $F_{k-1}$ is bounded by $2^{2^k}(d+1)^{2^{k-1}-2}(n+2)^{2^{k-2}-1}$. Since

$$F_k = 2d(n+1)F_{k-1}^2 + 4D_{k-1}F_{k-1} + \frac{D_{k-1}}{d} + (n+1)F_{k-1},$$

we have

$$\begin{aligned} F_k \le\ & 2d(n+1)[2^{2^k}(d+1)^{2^{k-1}-2}(n+2)^{2^{k-2}-1}]^2 + \\ & 4 \times 2^{2^{k-1}-1} d^{2^{k-1}} 2^{2^k}(d+1)^{2^{k-1}-2}(n+2)^{2^{k-2}-1} + \\ & 2^{2^{k-1}-1} d^{2^{k-1}-1} + (n+1)2^{2^k}(d+1)^{2^{k-1}-2}(n+2)^{2^{k-2}-1}. \end{aligned}$$

Therefore

$$\begin{aligned} F_k \le\ & 2d(n+1)2^{2^{k+1}}(d+1)^{2^k-4}(n+2)^{2^{k-1}-2} + \\ & (d+1)^2 2^{2^{k+1}}(d+1)^{2^k-4}(n+2)^{2^{k-1}-2} + \\ & d^2(n+1)2^{2^{k+1}}(d+1)^{2^k-4}(n+2)^{2^{k-1}-2} + \\ & (n+1)2^{2^{k+1}}(d+1)^{2^k-4}(n+2)^{2^{k-1}-2} \\ =\ & 2^{2^{k+1}}(d+1)^{2^k-2}(n+2)^{2^{k-1}-1}. \quad \square \end{aligned}$$

### 3.3. Implementation

We have implemented Algorithm 3.1 in the kernel of Mathematica. The implemented function is designed so that the users can control several parameters (options) for the conversion. Among the parameters are:

- `ConvertOnly -> False/True`, which tells the function to compute or not to compute the Gröbner basis with respect to the first term order,
- `Perturbation -> True/False`, which tells the function to use or to skip the perturbation (varying) of the weight vectors.
- `MaximalGroebnerBasis -> False/True`, which tells the function to work with maximal or reduced Gröbner bases.

The following examples show the efficiency of the fast algorithm for basis conversion presented in the previous section. All the experiments were carried out on a laptop PC using Linux Redhat 5.0 on an Intel MMX-233 MHz processor with 144 MB RAM.

Example 3.2. (ISSAC system challenge 1997) Given the zero-dimensional system of equations

$$\begin{aligned}
&8w^2 + 5wx - 4wy + 2wz + 3w + 5x^2 + 2xy - 7xz - 7x + 7y^2 \\
&-8yz - 7y + 7z^2 - 8z + 8, \\
&3w^2 - 5wx - 3wy - 6wz + 9w + 4x^2 + 2xy - 2xz + 7x + 9y^2 \\
&+6yz + 5y + 7z^2 + 7z + 5, \\
&-2w^2 + 9wx + 9wy - 7wz - 4w + 8x^2 + 9xy - 3xz + 8x + 6y^2 \\
&-7yz + 4y - 6z^2 + 8z + 2, \\
&7w^2 + 5wx + 3wy - 5wz - 5w + 2x^2 + 9xy - 7xz + 4x - 4y^2 \\
&-5yz + 6y - 4z^2 - 9z + 2;
\end{aligned}$$

we convert from a Gröbner basis with respect to the degree reverse lexicographic term order of the ideal generated by the system to a Gröbner basis with respect to the pure lexicographic term order determined by $x \succ y \succ z \succ w$. The initial forms at the target weight vector have as many as 17 terms. Using the improved method in this paper, the conversion took 5.33 s while the traditional Gröbner walk method did not stop after 10 000 s, which is even worse than the direct computation of the Gröbner basis using Buchberger's algorithm and the sugar cube strategy. (The direct computation took 159 s.)

Example 3.3. Given the system of equations

$$\begin{aligned}
&16 + 3x^3 + 16x^2z + 14x^2y^3, \\
&6 + y^3z + 17x^2z^2 + 7xy^2z^2 + 13x^3z^2
\end{aligned}$$

we convert from a Gröbner basis with respect to the degree reverse lexicographic term order of the ideal generated by the system to a Gröbner basis with respect to the pure lexicographic term order determined by $x \succ y \succ z$. The ideal is one-dimensional. The initial forms at the target weight vector have as many as 44 terms. Using the improved method in this paper, the conversion took 91.07 s while both the traditional Gröbner walk method and the direct computation of the Gröbner basis using Buchberger's algorithm and the sugar cube strategy did not stop after 10 000 s.

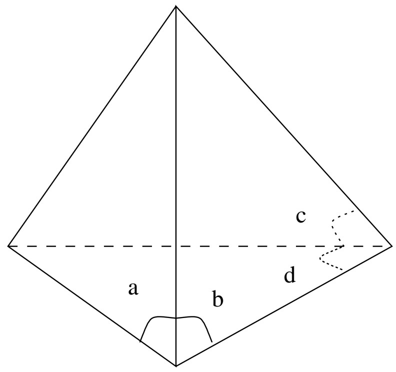


**Figure 1.** Tetrahedron.

The fast algorithm for basis conversion in this paper so far has the best performance of $3 \times 10^4$ times faster in comparison with the direct computation of the Gröbner basis using Buchberger's algorithm and the sugar cube strategy in the following example.

Example 3.4. Given the system of equations

$$15 + 10x^2y^2 + 13yz + 14xy^2z + 8x^2yz^2 + 11xy^3z^2,$$
$$5 + 4xy + 8y^2,$$
$$16x^3 + 19y + 4x^2y,$$

we convert from a Gröbner basis with respect to the degree reverse lexicographic term order of the ideal generated by the system to a Gröbner basis with respect to the pure lexicographic term order determined by $x \succ y \succ z$. The initial forms at the target weight vector have as many as 13 terms. Using the improved method in this paper, the conversion took 0.93 s, while the direct computation of the Gröbner basis using Buchberger's algorithm and the sugar cube strategy took 28 518 s.

## 4. Some Applications of the Method for Basis Conversion

### 4.1. Geometric Reasoning

Several problems in geometric reasoning can be transformed into variable elimination problems. However, the problems usually involve parameters, i.e. the ideal generated by the system of equations may be neither zero-dimensional nor a hypersurface. One of the strong points of the Gröbner walk method is that it is not restricted by any assumptions on the dimension of the ideal.

In contrast to multivariate resultant methods such as the Dixon resultant method, the method of Gröbner bases does not have any problem with extraneous solutions; in fact, it can be used for solving the problems in full generality.

Example 4.1. The problem is to compute the maximum volume that a tetrahedron can have in terms of the areas $a$, $b$, $c$ and $d$ of its faces. A straightforward approach using Lagrange multipliers, would lead to a system of 12 equations in 16 variables, which is too large even for computing a Gröbner basis with respect to the degree reverse lexicographic term order. Fortunately, Gerber (1975) showed that if there exist parameters $x$, $y$, $z$ and $w$ satisfying the equations

$$\begin{aligned}
&yz + zw + wy - a, \\
&zx + xw + wz - b, \\
&wx + xy + yw - c, \\
&xy + yz + zx - d, \\
&2(xyz + yzw + zwx + wxy) - 9T,
\end{aligned}$$

then the tetrahedron is orthocentric and therefore has the maximum volume with those face areas.

In Kapur (1998) the problem was deemed not easily solvable by any method but Dixon resultant. In that paper, the author reported that the problem can be solved in 76 s on a Sun workstation using the Dixon resultant method. Using the improved Gröbner walk method in this paper, we can solve the problem in 12 s on a laptop PC with an Intel MMX-233 MHz processor. Moreover, the solution is fully general; it is a quartic polynomial in $T^2$ of 434 terms and we are able to write down a radical formula for $T$ in terms of $a$, $b$, $c$ and $d$.

### 4.2. INTERSECTION OF SURFACES AND IMPLICITIZATION

Designing curves and surfaces plays an important role in the construction of many products such as airplane fuselages and wings, ship hulls, propeller blades, car bodies, shoe insoles and bottles. The subject is studied in several research areas such as computer aided geometric design (CAGD), visualization and solid modeling, where curves and surfaces are approximated, represented and processed by a computer. In these domains, finding the intersection of two surfaces is a fundamental and difficult problem.

Intersections are needed to build and interrogate models of complex shapes in the computer. They are primarily used to evaluate set operations on primitive volumes in creating boundary representations of complex objects or for subsequent manipulation of the objects.

Due to the importance of the problem, there have been persistent efforts at devising algorithms for this problem (see Barnhill *et al.*, 1987; Bajaj *et al.*, 1988; Hoffmann, 1989; Patrikalakis, 1993; Tran, 1995). The main issue in the problem is the efficient discovery and description of all features of the solution with high precision commensurate with the tasks required from the underlying geometric modeler. Reliability and efficiency of intersection algorithms are two basic prerequisites for their effective use in any geometric modeling system. They are closely associated with the way the algorithm handles such features as near singular cases, small loops, etc.

Since parametric form is the most common representation of surfaces in CAGD, solid modeling, etc. (because of its convenient manner for generating points along curves or surfaces), we first start with finding the intersection of two parametric surfaces.

Given two parametric surfaces $S_1$ and $S_2$ in $\mathbb{C}^3$ defined by the systems

$$\begin{cases} S_1 = \left(x = \frac{u_1(s_1,t_1)}{r_1(s_1,t_1)}, y = \frac{v_1(s_1,t_1)}{r_1(s_1,t_1)}, z = \frac{w_1(s_1,t_1)}{r_1(s_1,t_1)}\right), \\ S_2 = \left(x = \frac{u_2(s_2,t_2)}{r_2(s_2,t_2)}, y = \frac{v_2(s_2,t_2)}{r_2(s_2,t_2)}, z = \frac{w_2(s_2,t_2)}{r_2(s_2,t_2)}\right), \end{cases}$$

one needs to find a closed form expression for the intersection, which can be used for subsequent manipulation. But, whereas two (relatively prime) plane curves intersect in some finite number of isolated points, two surfaces meet in a space curve comprised of finitely many components.

It is well-known that the closed form expression of the intersection can be obtained by: first, implicitizing the first parametric surface $S_1$ getting an implicit equation, say $f_1(x, y, z)$; second, substituting the parametric expression of the second surface $S_2$ into the implicit equation $f_1(x, y, z)$. This results in the implicit representation $f_1(x(s_2, t_2)$, $y(s_2, t_2)$, $z(s_2, t_2))$ of the intersection curve in a parameter space, i.e. a projected image of the intersection.

Implicitization is an elimination problem, even though for some time it has been deemed as unsolvable in CAD literature. Sederberg and Anderson (1984) presented a solution of the implicitization problem using resultants. The solution is spelled out for surfaces in three dimensions and curves in two dimensions. However, in the general case, except for the traditional resultant system method (van der Waerden, 1940), which is very inefficient, other conditional multivariate resultants such as Dixon's resultant can yield only one implicit equation (e.g. implicitization of a space curve may have two or more equations) and may introduce nontrivial extraneous solutions. Arnon and Sederberg (1984) have shown how the method of Gröbner bases (Buchberger, 1965, 1985) can be used for the implicitization problem of $(n-1)$-dimensional hypersurfaces.

There were concerns (e.g. Buchberger, 1986; Hoffmann, 1993) about the efficiency of the method of Gröbner bases for elimination in terms of computation time as well as memory used for the computation. Fortunately, there have been extensive improvements of the method over the last 10 years. For example, Hoffmann (1989) has shown how to use the strategy of the FGLM-algorithm for the implicitization problem of $(n-1)$-dimensional hypersurfaces. Moreover, in some instances, such as in implicitization of surfaces with polynomial parametric form, one knows in advance that the given set of polynomials is already a Gröbner basis with respect to some term orders. Therefore, it would be more efficient if one knew how to convert the known Gröbner basis to a Gröbner basis of the ideal with respect to an elimination term order. By using a new algorithm for basis conversion for non-zero dimensional systems together with an elimination term order, we can implicitize the first surface of Example 4.2 (see below) in 0.7 s using 0.5 MB on a Sun Ultrasparc-5 machine. Meanwhile, the same surface cannot be parametrized in 85 500.0 s after consuming 512 MB memory using Buchberger's algorithm with lexicographic term order.

Let $K$ be an algebraically closed field. We consider the general implicitization problem for rational parametric surfaces.

Problem 4.1. (General implicitization)

Given: a surface in parametric form $S \equiv (x_i = \frac{p_i(s_1,\ldots,s_m)}{q(s_1,\ldots,s_m)}$, where $p_i, q \in K[s_1, \ldots, s_m]$, $i = 1, \ldots, n)$.

Find: $f_1, \ldots, f_k \in K[x_1, \ldots, x_n]$ such that the algebraic set $V(I) \equiv \{(\overline{x_1}, \ldots, \overline{x_n}) | f_i(\overline{x_1}, \ldots, \overline{x_n}) = 0,\ i = 1, \ldots, k\}$ is the smallest algebraic set in $K^n$ that contains the image $S$ of the parametrization.

In order to deal with the base points, one can either make use of an ingenious trick by adding one more equation $qt - 1$ into the system, where $t$ is a new variable, or embed the surface into the projective space. More theoretical details can be found in Cox *et al.* (1996).

The problem requires constructing $k$ polynomials implicitly defining hypersurfaces whose intersection is the smallest algebraic set containing the image described by the parametric representation.

It is worth mentioning that the algebraic set $V(I)$ is irreducible. In order to comply with the degree of the implicitization, one may have to embed the algebraic set into the projective space (Hoffmann, 1989). In this case, the corresponding algebraic set is still (projectively) irreducible.

Algorithm 4.1. (Implicitization)

$GB = GröbnerBasis(\{x_1q - p_1, \ldots, x_nq - p_n, qt - 1\})$ *a Gröbner basis with respect to an elimination term order determined by* $t \succ s_i \succ x_j\ \forall i = 1 \ldots m, j = 1 \ldots n$.
$\{f_1, \ldots, f_k\} = GB \cap K[x_1, \ldots, x_n]$.

Note that since we assume the base field is algebraically closed, the differences between the algebraic set $V(I)$ defined by the implicit equations and the image $S$ of the parametric surface are unions of some lower-dimensional algebraic sets, e.g. curves on a surface or points on a curve. Unfortunately, the property is not preserved for non-algebraically closed fields such as $\mathbb{R}$ as shown by the following counter-example. Let $S = (u^2, v^2, uv) \subset \mathbb{R}^3$ then $V(I) = V(z^2 - xy)$. However, $S$ covers only half of $V(I)$.

We can now state the problem of intersection of parametric surfaces and generalize it to higher dimensions as follows.

Problem 4.2. (Intersection of parametric surfaces)

Given: two parametric surfaces $S_1 = (x_i = \frac{p_{1i}(s_{11},\ldots,s_{1m})}{q_1(s_{11},\ldots,s_{1m})}$, where $p_{1i}, q_1 \in K[s_{11}, \ldots, s_{1m}]$, $i = 1, \ldots, n)$, and $S_2 = (x_i = \frac{p_{2i}(s_{21},\ldots,s_{2m})}{q_2(s_{21},\ldots,s_{2m})}$, where $p_{2i}, q_2 \in K[s_{21}, \ldots, s_{2m}]$, $i = 1, \ldots, n)$.
Find: a closed form expression for the intersection of $S_1$ and $S_2$.

The closed form expression for the intersection can be the projected image of the intersection onto the parameter space of either of the two given surfaces.

Algorithm 4.2. (Intersection of parametric surfaces)

$\underline{F_1} = \{f_1, \ldots, f_k\} \subset K[x_1, \ldots, x_n]$, *an implicitization of the surface* $S_1$.
$\overline{F_1} = F_1|_{x_i \leftarrow \frac{p_{2i}(s_{21},\ldots,s_{2m})}{q_2(s_{21},\ldots,s_{2m})}, \forall i=1,\ldots,n}$

Example 4.2. Given two tensor product surfaces, where the first one is the airplane wing-shape surface

$$\begin{aligned} q_1 &= (1-u)(1-v)^4 + 40(1-u)v(1-v)^3 + 120(1-u)v^2(1-v)^2 \\ &\quad +40(1-u)v^3(1-v) + (1-u)v^4 + u(1-v)^4 + 40uv(1-v)^3 \\ &\quad +120uv^2(1-v)^2 + 40uv^3(1-v) + uv^4; \\ x &= \frac{1}{q_1}(60u(1-v)^4 + 2400uv(1-v)^3 + 7200uv^2(1-v)^2 + 2400uv^3 \end{aligned}$$

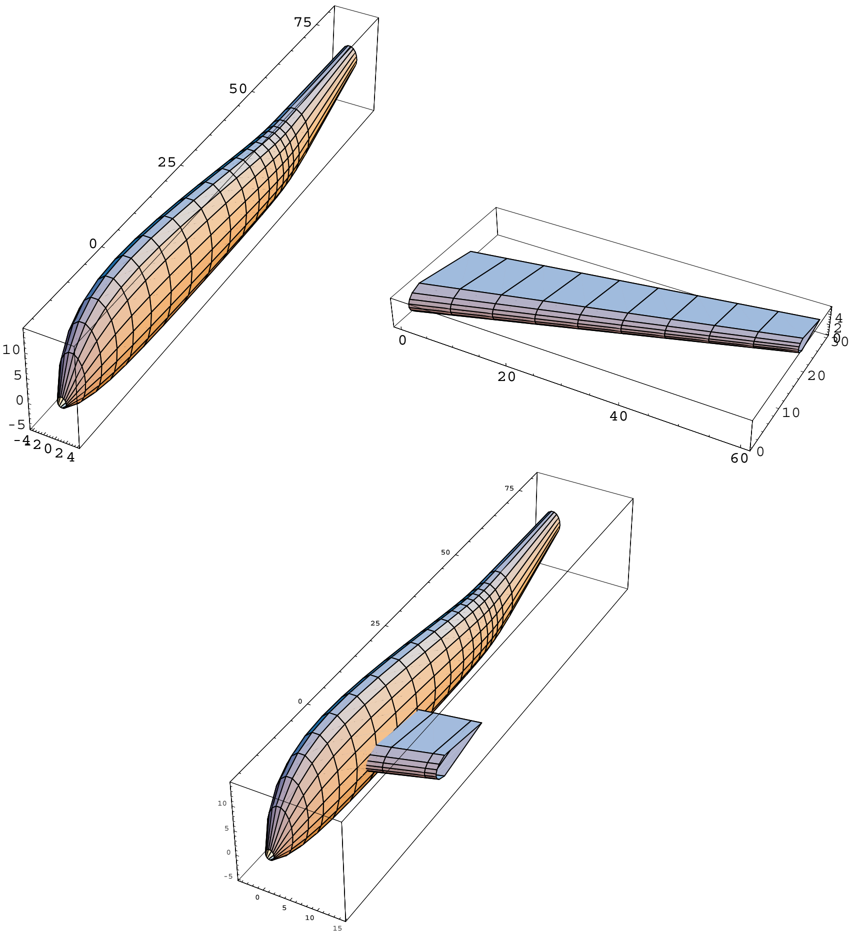


**Figure 2.** Intersections of parametric surfaces.

$$
\begin{aligned}
&(1-v) + 60uv^4);\\
y &= \frac{1}{q_1}(20(1-u)(1-v)^4 + 160(1-u)v(1-v)^3 + 160(1-u)v^3(1-v)\\
&\quad +20(1-u)v^4 + 30u(1-v)^4 + 880uv(1-v)^3 + 2400uv^2(1-v)^2\\
&\quad +880uv^3(1-v) + 30uv^4);
\end{aligned}
$$

$$
\begin{aligned}
z &= 3\frac{1}{q_1}((1-u)(1-v)^4 + 280(1-u)v(1-v)^3 + 360(1-u)v^2(1-v)^2\\
&\quad -40(1-u)v^3(1-v) + 3(1-u)v^4 + 3u(1-v)^4 + 200uv(1-v)^3\\
&\quad +360uv^2(1-v)^2 + 40uv^3(1-v) + 3uv^4);
\end{aligned}
$$

and the second one is the airplane fuselage-shape surface

$$
\begin{aligned}
q_2 &= (1-u)^3(1-v)^5 + 10(1-u)^3v(1-v)^4 + 30(1-u)^3v^2(1-v)^3\\
&\quad +30(1-u)^3v^3(1-v)^2 + 10(1-u)^3v^4(1-v) + (1-u)^3v^5
\end{aligned}
$$

$$+30u(1-u)^2(1-v)^5 + 300u(1-u)^2v(1-v)^4 + 900u(1-u)^2 v^2(1-v)^3 + 900u(1-u)^2v^3(1-v)^2 + 300u(1-u)^2v^4(1-v) +30u(1-u)^2v^5 + 30u^2(1-u)(1-v)^5 + 300u^2(1-u)v(1-v)^4 +900u^2(1-u)v^2(1-v)^3 + 900u^2(1-u)v^3(1-v)^2 + 300u^2 (1-u)v^4(1-v) + 30u^2(1-u)v^5 + u^3(1-v)^5 + 10u^3v(1-v)^4 +30u^3v^2(1-v)^3 + 30u^3v^3(1-v)^2 + 10u^3v^4(1-v) + u^3v^5;$$

$$x = \frac{1}{q_2}(-12.50(1-u)^3v(1-v)^4 - 37.50(1-u)^3v^2(1-v)^3 + 37.50 (1-u)^3v^3(1-v)^2 + 12.50(1-u)^3v^4(1-v) - 1650u(1-u)^2v (1-v)^4 - 4950u(1-u)^2v^2(1-v)^3 + 4950u(1-u)^2v^3 (1-v)^2 + 1650u(1-u)^2v^4(1-v) - 1650u^2(1-u)v(1-v)^4 -4950u^2(1-u)v^2(1-v)^3 + 4950u^2(1-u)v^3(1-v)^2 + 1650 u^2(1-u)v^4(1-v) - 25u^3v(1-v)^4 - 75u^3v^2(1-v)^3 + 75u^3v^3 (1-v)^2 + 25u^3v^4(1-v));$$

$$y = \frac{1}{q_2}(-20(1-u)^3(1-v)^5 - 200(1-u)^3v(1-v)^4 - 600(1-u)^3v^2 (1-v)^3 - 600(1-u)^3v^3(1-v)^2 - 200(1-u)^3v^4(1-v) - 20 (1-u)^3v^5 - 420u(1-u)^2(1-v)^5 - 4200u(1-u)^2v(1-v)^4 -12600u(1-u)^2v^2(1-v)^3 - 12600u(1-u)^2v^3(1-v)^2 - 4200 u(1-u)^2v^4(1-v) - 420u(1-u)^2v^5 + 1500u^2(1-u)(1-v)^5 +15000u^2(1-u)v(1-v)^4 + 45000u^2(1-u)v^2(1-v)^3 + 45000 u^2(1-u)v^3(1-v)^2 + 15000u^2(1-u)v^4(1-v) + 1500u^2(1-u) v^5 + 80u^3(1-v)^5 + 800u^3v(1-v)^4 + 2400u^3v^2(1-v)^3 + 2400 u^3v^3(1-v)^2 + 800u^3v^4(1-v) + 80u^3v^5);$$

$$z = \frac{1}{q_2}(-(1-u)^3(1-v)^5 - 10(1-u)^3v(1-v)^4 + 60(1-u)^3v^2(1-v)^3 +60(1-u)^3v^3(1-v)^2 - 10(1-u)^3v^4(1-v) - (1-u)^3v^5 -150u(1-u)^2(1-v)^5 - 1500u(1-u)^2v(1-v)^4 + 11700u (1-u)^2v^2(1-v)^3 + 11700u(1-u)^2v^3(1-v)^2 - 1500u(1-u)^2v^4 (1-v) - 150u(1-u)^2v^5 - 150u^2(1-u)(1-v)^5 - 1500u^2(1-u) v(1-v)^4 + 11700u^2(1-u)v^2(1-v)^3 + 11700u^2(1-u)v^3(1-v)^2 -1500u^2(1-u)v^4(1-v) - 150u^2(1-u)v^5 + 4u^3(1-v)^5 + 40u^3 v(1-v)^4 + 321u^3v^2(1-v)^3 + 321u^3v^3(1-v)^2 + 40u^3v^4(1-v) +4u^3v^5);$$

We first implicitize the wing-shape surface using a new algorithm for basis conversion for non-zero dimensional systems together with an elimination term order. The computation has been done in 0.7 s using 0.5 MB on a Sun Ultrasparc-5 machine. The implicitization representation of the wing-shape surface is an irreducible polynomial in $x$, $y$ and $z$, which has degree 8 and consists of 143 terms. Simply substituting the parametric expression of the fuselage-shape surface into the implicitization representation of the wing-shape surface we get a closed form expression for the intersection, which is a projected image of the intersection onto the parameter space of the wing-shape surface. The closed form expression is a rational function in $u$ and $v$ whose numerator has degree 88 and consists of 2009 terms. The denominator is of degree 11 and consists of 30 terms.

## 5. Conclusion

We presented a deterministic and constructive method for varying the target point in order to ensure the generality of its position, a crucial condition to the performance of the Gröbner walk method for basis conversion. We showed a fast algorithm for basis conversion and its implementation in the kernel of Mathematica. Our experiments show the superlative performance of our improved Gröbner walk method in comparison with other known methods. Our best performance is $3 \times 10^4$ times faster than the direct computation of the reduced Gröbner basis with respect to pure lexicographic term order (using the Buchberger algorithm and the sugar cube strategy). Finally, we reported the practical value of our algorithm for some problems in CAGD and geometric reasoning.

## References


Amrhein, B., Gloor, O. (1998). The fractal walk. In Buchberger, B., Winkler, F. eds, *Proceedings of the Conference "33 Years of Gröbner Bases", Linz.* Cambridge, UK, Cambridge University Press.

Amrhein, B., Gloor, O., Küchlin, W. (1996). Walking faster. In *Proceedings of the DISCO'96, Karlsruhe, Germany.* Berlin, Springer-Verlag.

Arnon, D., Sederberg, T. (1984). Implicit equation for a parametric surface by Gröbner basis. In *Proceedings of 1984 MACSYMA User's Conf.*, pp. 431–436. New York, USA, Schenectady.

Bajaj, C. L., Hoffmann, C. M., Lynch, R. E., Hopcroft, J. E. H. (1988). Tracing surface intersections. *Comput. Aided Geom. Des.*, **5**, 285–307.

Barnhill, R., Farin, G., Jordan, M., Piper, B. (1987). Surface/surface intersection. *Comput. Aided Geom. Des.*, **4**, 3–16.

Bayer, D. (1982). The division algorithm and the Hilbert scheme. Ph.D. Thesis, Harvard University, Cambridge.

Buchberger, B. (1965). An algorithm for finding a basis for the residue class ring of a zero-dimensional polynomial ideal (in German). Ph.D. Thesis, Institute of Mathematics, Univ. Innsbruck, Innsbruck, Austria.

Buchberger, B. (1985). Gröbner bases: An algorithmic method in polynomial ideal theory. In Bose, N. K. ed., *Multidimensional Systems Theory*, chapter 6, pp. 184–232. Dodrecht, Reidel Publishing Company.

Buchberger, B. (1986). What can Gröbner bases do for computational geometry and robotics. In *Proceedings of Workshop on Geometric Reasoning*, Oxford, England.

Caniglia, L., Galligo, A., Heintz, J. (1991). Equations for the projective closure and effective nullstellensatz. *Discrete Appl. Math.*, **33**, 11–23.

Collart, S., Kalkbrener, M., Mall, D. (1997). Converting bases with the Gröbner walk. *J. Symb. Comput.*, **24**, 465–469.

Cox, D., Little, J., O'Shea, D. (1996). In *Ideals, Varieties, and Algorithms: An Introduction to Computational Algebraic Geometry and Commutative Algebra, Undergraduate Texts in Mathematics*, 2$^{\text{nd}}$ edn. New York, Springer-Verlag.

Dubé, T. D. (1990). The structure of polynomial ideals and Gröbner bases. *SIAM J. Comput.*, **19**, 750–773.

Eisenbud, D. (1995). *Commutative Algebra with a View Towards Algebraic Geometry*, volume 150 of *Graduate Texts in Mathematics.* New York, Springer.

Faugère, F. C., Gianni, P., Lazard, D., Mora, T. (1993). Efficient computation of zero-dimensional Gröbner bases by change of ordering. *J. Symb. Comput.*, **16**, 329–344.

Gerber, L. (1975). The orthocentric simplex as an extreme simplex. *Pac. J. Math.*, **56**, 97–111.

Hoffmann, C. (1989). *Geometric and Solid Modeling—An Introduction.* San Mateo, California, Morgan Kaufmann Publishers, Inc..

Hoffmann, C. (1993). Implicit curves and surfaces in CAGD. *IEEE Comput. Graph. Appl.*, **13**, 79–88.

Kalkbrener, M. (1999). On the complexity of Gröbner bases conversion. *J. Symb. Comput.*, **28**, 265–274.

Kapur, D. (1998). Automated geometric reasoning: Dixon resultants, Gröbner bases and characteristic sets. In Wang, D. ed., *Automated Deduction in Geometry*, volume 1360 of *Lecture Notes in Artificial Intelligent.* pp. 1–36. Springer.

Patrikalakis, N. M. (1993). Surface-to-surface intersections. *IEEE Comput. Graph. Appl.*, **1**, 89–95.
Sederberg, T. W., Anderson, D. C. (1984). Implicit representation of parametric curves and surfaces. *Computer Vison, Graphics and Image Processing*, **28**, 72–84.
Tran, Q.-N. (1995). A hybrid symbolic-numerical method for tracing surface-to-surface intersections. In *The proceedings of ISSAC'95*, pp. 51–58. Montreal, Canada.
Traverso, C. (1996). Hilbert functions and the Buchberger algorithm. *J. Symb. Comput.*, **22**, 355–376.
van der Waerden, B. L. (1940). *Moderne Algebra*, volume 2, 2$^{nd}$ edn. Berlin, Springer-Verlag.

*Originally Received 31 January 1999*
*Accepted 22 June 1999*