UNIVERSITI
TEKNOLOGI
MARA

University Publication Centre (U

Academic Journal

UiTM Johor

Volume 10 2011 ISSN 1675-9141

ACADEMIC JOURNAL UiTM JOHOR
A Publication of
Universiti Teknologi MARA, Johor

The Gröbner Package in Maple and Computer Algebra System for Solving Multivariate Polynomial Equations

Shamsatun Nahar Ahmad
e-mail: shams551@johor.uitm.edu.my
Nor'aini Aris
e-mail: noraini@fs.utm.my
Universiti Teknologi Mara
Johor

ABSTRACT

This paper is a preliminary survey on the developments in the techniques of solving multivariate polynomial equations. Currently two main approaches originating from algebraic geometry have been used to compute the roots of a zero dimensional polynomial system. The first approach involves Gröbner bases computations. This method involved computing common roots by eliminating a set of variables from a system of polynomial equations and thereby reducing the problem to a sequence of univariate polynomials. The other approach is based on resultant formulations, which can eliminate many variables simultaneously and can also be performed in floating point arithmetic. The resultant techniques can also be viewed from linear algebra to reduce the root computations to a nonsingular eigenvector problem and to find approximate values of the solutions. In this paper, we present an overview of the stages and development in the Gröbner basis techniques and to discuss some basic implementations of the Gröbner package in Maple and the computer algebra system related to solving multivariate polynomial equations.

Keywords: *Gröbner bases, ideal basis, Maple, multivariate polynomial equations, variety.*

Introduction

A polynomial equation corresponds to a polynomial function $f(x)$ which is set to zero. It is so called a zero function which means that $f(x) = 0$ for all $x \in k$, k is any infinite field. The solutions to the equation are called the roots of the polynomial and they are the zeros of the function. If $x = a$, is a root of a polynomial, then $(x - a)$ is a factor of that polynomial or $(x-a) \in k[x]$ vanishes at a, which gives the zero function on the affine space k.

Solving a polynomial equation in one unknown is easily done on computer by some well known root-finding algorithms. Formulas for the roots of polynomials up to a degree of 2 have been known since ancient times and up to a degree of 4 since the 16th century. However, formulas for degree 5 eluded researchers. In 1824, Niels Henrik Abel proved that there is no general formula (involving only the arithmetical operations and radicals) for the roots of a polynomial of degree 5 or greater in terms of its coefficients (Abel-Ruffini theorem). This result initiates the foundation of Galois Theory, which engages in a detailed study of relationships among roots of polynomials (Stewart (2003)).

A system of polynomial equations is a set of *n* polynomials with coefficients over an arbitrary field. Solving systems of polynomial equations is a fundamental problem in geometric computations. In particular, it is a crucial or most challenging problem in algorithmic algebra and the computational complexity makes it very difficult to solve some problems in practice. Yun and Pohst (1981) studied the difficulties in computational methods for ideal bases which are useful for solving a system of polynomial equations. However, the problems of finding the common zeros of polynomial equations are far from satisfactorily solved. The investigations also reveal that polynomial systems often result in large number of solutions.

The theory on systems of multivariate polynomial equations also involves field and ideal theory. In recent years, the search for efficient algorithms for solving systems of polynomial equations has received renewed attention due to their importance to a variety of problems of both practical and theoretical interests. Greuel (2000) stressed about the need to count real solutions or to find exact or approximate solutions to such systems has arisen in a wide range of practical areas, including robotics and kinematics, computational number theory, solid modeling, quantifier elimination and geometric reasoning problems.

The currently known techniques for solving polynomial systems can be classified into symbolic, numeric and geometric. In the context of finding exact solution, symbolic method based on Gröbner bases and resultants originate from algebraic geometry and can be used for eliminating variables, which reduces the problem to finding roots of univariate polynomials. However, if the reduced equation is of degree 5 or more, there are no radical formulae that can solve the equation (Stewart (2003)). In such cases, combining the techniques of Gröbner bases with other numerical techniques, elimination ideal and extension theorem may be desirable.

New methods for Gröbner bases conversion such as the well known FGLM algorithm has been developed by Faugère, et al., (1993) and the Gröbner Walk algorithm can be applied to address the question of an efficient elimination methods using a suitable order by using basis conversion. While the main ideas of the

Gröbner walk is simple, the actual implementation of the algorithm is less understood.

Later improvement of Buchberger algorithm is the F_4 algorithm which was first described by Faugère (2002) and'is claimed to be the fastest routines for computing Gröbner bases in current implementation by Farr and Pearce (2005). They have tested the routines under Magma and the time results are presented on Steel's web page.

Another powerful method for polynomial system solving is a method based on resultants. Basically this method is only applied to generic polynomial systems. However later development in the multivariate resultant makes it possible to solve specific application problems efficiently, as revealed in Manocha (1994); Kapur, et al., (1994) or Canny, and Emiris (1993). In addition, algorithms for resultant computation deal with matrices and determinants. In this approach, combining properties from algebraic geometry and linear algebra leads to the construction of effective and efficient resultant computations and finding solutions to multivariate polynomial equations.

This paper gives an overview of the stages and development in the Gröbner basis techniques and to discuss some basic implementations of the Gröbner package in Maple related to solving multivariate polynomial equations.

The Algebra-Geometry in Solving Polynomial Equations

Algebraic geometry relates to the study of geometric objects defined by polynomial equations using algebraic tools and is also called a symbolic method. In general, the basic problem of algebraic geometry is to study the set of points in k^n satisfying a system of equations $f_1(x_1,\ldots,x_n)=0,\cdots,f_m(x_1,\ldots,x_n)=0$, where k is a field and $f_1,\ldots,f_m$ belongs to the polynomial ring $k[x_1,\ldots,x_n]$. The solution set of $f_1=\ldots=f_m=0$ is called the algebraic set, referred also as algebraic or affine variety of $f_1,\ldots,f_m$ and is denoted as $V=V(f_1,\ldots,f_m)$. The geometry of interest is the affine varieties of curves and surfaces defined by polynomial equations as described by Greuel (2000). In finding the most effective solution of the affine varieties, the relation between algebra and geometry can be investigated by considering the algebraic and geometric objects such as fields, ideals, affine varieties and propositions, theorems or corollaries involving them.

Definition 1 (Properties of multivariate polynomials)

Let $f(x_1,\ldots,x_n)=\sum_\alpha a_\alpha x^\alpha$ be a multivariate polynomial in $k[x_1,\ldots,x_n]$.

(i) a_α refer to the coefficient of the monomial x^α , $a_\alpha \in k$.

(ii) If $a_\alpha \neq 0$, then $a_\alpha\ x^\alpha$ is a term of *f*.

(iii) The total degree of *f*, $\deg(f)$ is the maximal $|\alpha|$ such that the coefficient a_α is nonzero.

Here, the notation x^α represents the power products $x_1^{\alpha_1}x_2^{\alpha_2}...x_n^{\alpha_n}$ where each α_i is a nonnegative integer. So, $\alpha=(\alpha_1,\alpha_2,...\alpha_n)\in \mathbf{N}^n$. Several term order can be defined on the set of all power products of multivariate polynomials as given below:

Definition 2 (Becker, and Weispfenning (1993))

Let $\alpha=(\alpha_1,\ldots,\alpha_n),\quad \beta=(\beta_1,\ldots,\beta_n)\in \mathrm{N}^n$ with $x_1 > x_2 > \cdots > x_n$

(1) Lexicographical order (or Lex-order), symbolically $>_{\mathrm{Lex}}$ is defined by $\alpha >_{\mathrm{Lex}} \beta$ if and only if the left-most nonzero entry in $\alpha-\beta$ is positive.

(2) Degree-lexicographical order (or DegLex order), symbolically $>_{\mathrm{DegLex}}$ is defined by $\alpha >_{\mathrm{DegLex}} \beta$ if and only if $\deg(\alpha)>\deg(\beta)$ or $\deg(\alpha)=\deg(\beta)$ and $\alpha >_{\mathrm{Lex}} \beta$.

(3) Degree-reverse-lexicographical order (or DegRevLex order), symbolically $>_{\mathrm{DegRevLex}}$ is defined by $\alpha >_{\mathrm{DegRevLex}} \beta$ if and only if $\deg(\alpha)>\deg(\beta)$ or $\deg(\alpha)=\deg(\beta)$ and the right-most nonzero entry in $\alpha-\beta$ is negative.

Example 1

(1) $x_1x_2^3 >_{\mathrm{Lex}} x_2^3x_3^5$.

(2) $x_1x_2^2x_3^3 >_{\mathrm{DegLex}} x_1^2x_2^3$ and $x_1^3x_2^2x_3^3 >_{\mathrm{DegLex}} x_1^2x_2^3x_3^3$.

(3) $x_1^4x_2^7x_3 >_{\mathrm{DegRevLex}} x_1^4x_2^2x_3^5$ and $x_1^3x_2^3x_3^2 <_{\mathrm{DegRevLex}} x_1^4x_2^2x_3^2$

Definition 3 (Becker, and Weispfenning (1993))

Let $f(x_1,\ldots,x_n)=\sum_\alpha a_\alpha x^\alpha$ be a nonzero polynomial in $k[x_1,\ldots,x_n]$ and let > be a monomial order.

(i) The multidegree of f is multigraded$(f)=\max((\alpha\in \mathrm{Z}^n_{\geq 0}):a_\alpha\neq 0)$.

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

(iii) The leading monomial of f is $\mathrm{LM}(f) = x^{\mathrm{multideg}(f)}$ (with coefficient 1).

(iv) The leading term of f is $\mathrm{LT}(f) = \mathrm{LC}(f)\mathrm{LM}(f)$.

Example 2
The polynomial $f = 3xyz + 4z^2 - 3x^2 + 7x^2y^2$ in the lex order is $7x^2y^2 - 3x^2 + 3xyz + 4z^2$. Then $\mathrm{multideg}(f) = (2,2,0)$, $\mathrm{LC}(f) = 7,\ \mathrm{LM}(f) = x^2y^2,\ \mathrm{LT}(f) = 7x^2y^2$.

Definition 4 (Affine *n*-space) (Cox, et al., (1996))
Given a field *k* and a positive integer *n*, the *n*-dimensional affine space over *k* is the set $k^n = \{(a_1,\ldots,a_n) : a_1,\ldots,a_n \in k\}$.

From the above definitions, we can relate polynomials to affine *n*-space over *k*. Any polynomial $f = \sum_\alpha a_\alpha x^\alpha \in k[x_1,\ldots,x_n]$ gives a function $f : k^n \to k$ defined by the map $(a_1,\ldots,a_n) \to f(a_1,\ldots,a_n)$. The value $f(a_1,\ldots,a_n)$ is obtained by substituting a_α for x^α in *f*. Subsequently, this gives the affine variety in k^n, namely $V(f)$.

Proposition 1 (Cox, et al., (1998))
Let *k* be an infinite field and $f \in k[x_1,\ldots,x_n]$. Then $f = 0$ in $k[x_1,\ldots,x_n]$ if and only if $f : k^n \to k$ is the zero function.

Definition 5 (Affine Variety) (Cox, et al., (1996))
Let *k* be a field, and let $f_1,\ldots,f_m \in k[x_1,\ldots,x_n]$. Then the set

$$V(f_1,\ldots,f_m) = \{(a_1,\ldots,a_n) \in k^n : f_i(a_1,\ldots,a_n) = 0\ \forall 1 \le i \le m\} \qquad (1)$$

is an affine variety defined by $f_1,\ldots,f_m$. It is also called the vanishing locus of *f* in k^n. If f_i nonconstant, $V(f_{1,\ldots,}f_m)$ is called a hypersurface in k^n.

For simplicity, the affine variety defined by $f_1,\ldots,f_m$ is often denoted by the symbol *V*, and the polynomial ring $k[x_1,\ldots,x_n]$ is denoted as $k[x]$. Moreover, this algebraic set depends only on the ideal generated by $f_1,\dots,f_m$ in $k[x]$, which is given by

$$I=\langle f_1,\dots,f_m\rangle=\left\{f\in k[x]\mid f=\sum_{i=1}^{m}g_if_i,\ g_i\in k[x]\right\}. \tag{2}$$

By the definition of I, $V(I)=\{x\in k^n \mid f(x)=0 \text{ for all } f\in I\}$, and hence $V=V(I)$.

The ideal $I=\langle f_1,\dots,f_m\rangle$ is generated by $f_1,\dots,f_m$ and is the smallest ideal in $k[x_1,\dots,x_n]$ containing $f_1,\dots,f_m$. The polynomials $f_1,\dots,f_m$ form a basis of I. Hilbert Bases Theorem proves that every ideal is finitely generated.

The radical of an ideal I is given by $\sqrt{I}=\{f\in k[x]: f^r\in I \text{ for some } r\geq 1\}$. An ideal I in $k[x]$ is a radical ideal if $I=\sqrt{I}$. There exist some polynomials in $k[x]$ such that $V=V(I)$ depends only on the radical of I. Let $A\subseteq k$ such that the biggest ideal determined by A is given by $I(A)=\{f\in k[x]: f(x)=0 \text{ for all } x\in A\}$. Thus, $I\subset\sqrt{I}\subset I(A)$ and $V(I(A))=V(\sqrt{I})=V(I)=V$. Here, A is the algebraic set in k^n and $I(A)$ is the vanishing ideal. Vanishing ideals have a property not shared by all ideals; they are radical ideals.

The important Hilbert Nullstellensatz states that, for k an algebraically closed field, we have for any variety $V\subset k^n$ and any ideal $J\subset k[x]$,

$$V=V(J)\Rightarrow I(V)=\sqrt{J}. \tag{3}$$

Therefore we can recover the ideal J, up to radical from its zero set. Farr and Pearce, 2005 assert that for a field such as $\mathbf{C}$ (but not for $\mathbf{R}$), geometry and algebra are almost equal and that we shall have occasions to see that the difference between I and $\sqrt{I}$ has very visible geometric consequences (Farr, and Pearce, 2005).

Solving Polynomial Equations

Let

$$
\begin{aligned}
&f_1(x_1,\ldots,x_n)=0\\
&f_2(x_1,\ldots,x_n)=0\\
&\vdots\\
&f_m(x_1,\ldots,x_n)=0
\end{aligned}
$$

(4)

be a system of m polynomial equations in n variables over a field k. "Solving" the system can also mean to determine whether there are many finitely solutions and if it is the case, to find the solutions over the given coefficient field k or over some extension field of k. If k is a subfield of $\mathbf{C}$, we may be interested to represent the solutions symbolically, or to compute floating point approximations of the solutions up to a given precision.

Currently, the techniques of Gröbner bases and resultants have received much attention as algorithmic methods for symbolic and numeric applications such as solving multivariate polynomial equations. Gröbner basis algorithm has been intensively studied and more applications have been exploited. In particular, solving polynomial equations and answering questions about the solvability of such systems are the most important applications of Gröbner bases that can be found in numerous research papers, proceedings or books such as Adams and Loustaunon (1994); Ajwa et.al (1995); Aubry et.al (1997); Becker and Weisfenning (1993); Cox et al., (1996, 1998); and Lebrun and Selesnick (2002).

Combining Gröbner Basis and Numerical Root Finding Algorithms

The algorithms for Gröbner bases computation have been implemented quite efficiently in several computer algebra systems, such as Maple, CoCoA, Magma, Mathematica, Matlab or Singular. A Gröbner basis has the property that the leading monomial of every polynomial in the ideal is divisible by the leading monomial of some polynomials in the Gröbner basis. The Elimination and Extension Theorem below can be used to find solutions of multivariate polynomial equations via elimination, (Cox, et al., (1998)).

Theorem 1 (Elimination Theorem)

If G is a Gröbner basis for an ideal I with a monomial order, then $G_l = G \cap k(x_{l+1},\ldots,x_n)$ is a Gröbner basis for the lth elimination ideal I_l such that $I_l = I \cap k[x_{l+1,\ldots,}x_n]$.

I_l is called an elimination ideal since the variables $x_1,...,x_l$ have been eliminated. A variation of Theorem 1 with its proof is given in (Decker, and Lossen, 2006), considering two sets of variables $\{x_1,...,x_n\}$ and $\{y_1,...,y_m\}$ ordered by term orders $<_x, <_y$ respectively, termed as an elimination order. An application of the elimination order to constructing a method of finding the generators for the intersection of two ideals is also presented.

Theorem 2 (Extension Theorem)
If k is algebraically closed, then a partial solution $(a_{l+1},...,a_n)$ in $V(I_l)$ extends to $(a_l, a_{l+1},...,a_n)$ in $V(I_{l-1})$ provided that the leading coefficient polynomials of a lex Gröbner basis for I_{l-1} do not all vanish at $(a_{l+1},...,a_n)$.

If the Gröbner basis G and $\{f_1,...f_m\}$ generates the same ideal I then $V(f_1,...,f_m) = V = V(I) = V(G)$. Therefore, finding the roots of G give the variety of f, V. The Elimination Theorem implies that a lex Gröbner basis reduces the problem of solving the system in (4) to the problem of solving a sequence of univariate polynomials. Since the elimination keep on eliminating the variables, reducing the system to an equivalent triangular form, the Gröbner basis polynomial with the least number of variables is univariate, say in x_n. From this polynomial we can find the partial solution $(a_n) \in V(I_{n-1})$, which can be extended one variable at a time using backward substitution, that is, solving a univariate polynomial equation in the lth variable for $l = n-1, n-2,...,2,1$, successively until all the solutions are obtained. A simple example illustrates this procedure:

Example 3
Let $f_1 = x^3 - 2xy + y^3$ and $f_2 = x^5 - 2x^2y^2 + y^5$ be a system of two bivariate polynomials in $\mathbf{Q}[x,y]$. Applying the Gröbner package in Maple gives:

```
>with(grobner):
>PList1:=[x^3-2*x*y+y^3,x^5-2*x^2*y^2+y^5]:
>VList1:=[x,y]:
>G1=gbasis(PList1, VList1, plex);
```

$$G1 := [x^3-2xy+y^3, 200xy^2+193y^9+158y^8-45y^7-456y^6+50y^5-100, y^{10}+2y^6-y^8-2y^7]$$

```
>factor(y^10+2*y^6-y^8-2*y^7);
```

$$y^6\left(y^2+2y+2\right)\left(y-1\right)^2$$

```
>solve(convert(G1,set),{x,y});
```

$$\{x=0, y=0\}, \{x=0, y=0\}, \{x=0, y=0\}, \{y=1, x=1\},$$
$$\{y=\mathrm{RootOf}\ (\_Z^2+2\ \_Z+2), x=-\mathrm{RootOf}\ (\_Z^2+2\ \_Z+2)-2\}$$

In this example, the Gröbner basis which generates the ideal $I=\left\langle f_{1,}f_2\right\rangle$ is given by
$G=\{x^3-2xy+y^3, 200xy^2+193y^9+158y^8-45y^7-456y^6+50y^5-100y^4, y^{10}+2y^6-y^8-2y^7\}$.
$G=G_0\cup G_1$ such that

$G_0=\{x^3-2xy+y^3, 200xy^2+193y^9+158y^8-45y^7-456y^6+50y^5-100y^4\}$.

$G_1=\{y^{10}+2y^6-y^8-2y^7\}$

Factorizing the univariate polynomial in G_1 gives

$$y^{10}+2y^6-y^8-2y^7=y^6(y^2+2y+2)(y-1)^2 \qquad (5)$$

and solving for y in (5) gives $V(I_1)=\{0,1,-1-i,1+i\}$.

Each $y_i\in V(I_1)$, is substituted into each polynomial $g\in G_0$.

For $y=0$, $G_0=\{x^3,0\}$. Therefore, $(0,0)\in V(I_0)$.

For $y=1$, $G_0=\{x^3-2x+1, 200x-200\}$ which gives $(1,1)\in V(I_0)$.

For $y=-1-i$, we obtain $x=-1+i$ which implies that $(-1+i,-1-i)\in V(I_0)$.

For $y=-1+i$, $x=-1-i$, which gives $(-1-i,-1+i)\in V(I_0)$.

Therefore, there are four zeroes of the system which is given by

$$V=\{(0,0),(1,1),(-1+i,-1-i),(-1-i,-1+i)\}.$$

If the Gröbner basis polynomial with the least number of variables is univariate and irreducible over **Q**[, Maple procedure gives a purely algebraic structural description of the solutions. If this solution is inexact, a numerical approach can be executed. However, substituting an approximate root into a polynomial in order to find the coordinates of the other variables will also give a polynomial, which is also an approximation. For systems with a large number of variables, accumulated errors after several approximation and extension steps can build up quite rapidly and the effect can be particularly severe if equations of high degree are present as stated in a book of Cox, et al., (1998). The effect on the roots of polynomials with approximate coefficients can be illustrated in the following example.

Example 4

Let $f = (x+2)^2 (x+4)^3$. If we perturb the coefficient of x^3 in f, we obtain a polynomial $g = f + 0.0001x^3 = x^5 + 16x^4 + 100.0001x^3 + 304x^2 + 448x + 256$ with approximate coefficients so that g is a polynomial that approximates f. Applying the function *fsolve* in Maple to approximate the solutions to g we obtain the approximate solutions $-4.059 \pm 0.1I, -3.881, -2.010, -1.990$ with a root which belongs to $\mathbf{C}$. It is difficult to determine the nature of the solutions of g. On the other hand, computing the exact solutions of f gives a double root $x = 2$ and a root $x = -4$ of multiplicity three.

Numerical methods are considered as unstable in an unpredictable way as proved by Decker, and Lossen (2006) which may have problems with over determined systems, as well as systems with multiple or equal spaced roots. In Cox et al., (1998) some examples of these problems are given and suggestions on the approach to circumvent these problems are presented. In Decker and Lossen (2006) the possibility of using Gröbner bases in a symbolic-numerical approach to solving is explored. Such an approach makes use of the symbolic methods to find additional information on the algebraic structure of the solution set which allows some control of the numerical methods and to preprocess the given system of equations so that it is expected to be better suited for numerical methods.

In the following, we present the developments of the Gröbner bases algorithms, which originate from the Buchberger's algorithm, and improvised through several stages with the aim of producing more efficient algorithms.

Developments of Gröbner Basis Algorithms

The first algorithm for computing Gröbner bases is based on polynomial ideal theory, which generates special bases for polynomial ideals with respect to a term ordering, due to Buchberger as surveyed in Buchberger (1985, 1989). Its applications include ideal membership testing and performing algebraic operations like union, intersection on ideals, in addition to eliminating a set of variables or computing the numerical solutions of a system of polynomial equations. In fact, it is a technique that provides algorithmic solutions to a variety of problems in computing algebra and algebraic geometry (Gianni, et.al, (1988)). It is also known as a computational tool for testing solvability of a system of polynomial equations, for counting the number of solutions (with multiplicities) and computations involving the quotient ring modulo the given polynomials. In general, Gröbner basis is a set of multivariate polynomials that has desirable algorithmic properties that can be reduced to triangular form and is regarded as an analogue of Gaussian elimination for multivariate polynomial reduction.

The basic idea to the theory of Gröbner bases is about polynomial reduction to compute a normal form from an S-polynomial of a given polynomial as stated in the Definition below.

Definition 6 (Ajwa, et al., (1995))
$G=\{g_1,\ldots,g_s\}$ of an ideal is a Gröbner basis if for all $f\in I$, there is g_i such that leading monomial of g_i, (LM(g_i)) divides the leading monomial of *f*, (LM(*f*)). $G=\{g_1,\ldots,g_s\}$ of I is a Gröbner basis if and only if for all *i*, *j*, the S-polynomial $(g_i,g_j)\xrightarrow[G]{}0$.

The first algorithm to compute Gröbner bases seemed to be very slow since the computed Gröbner bases is not reduced and there are some reductions S-polynomial that can be avoided. As a result, Buchberger (1989) developed a criterion to detect unnecessary reductions and allows detecting the S-polynomials that will reduce to zero without carrying out the reduction. Consider the following:

Lemma 1 (Buchberger's criterion)
Let *I* be a polynomial ideal and fix a monomial order $\succ$ on $k[x]$. Then a basis $\{g_1,\ldots,g_r\}$ for *I* is a Gröbner bases for *I* if and only if for all pairs (*i, j*) *with* $i\neq j$, there exist $h_1^{ij},\ldots,h_r^{ij}\in k[x]$ such that

$$S(g_i,g_j)=\sum_{t=1}^{r}h_t^{ij}g_t$$

and

$$\text{LCM}(\text{LM}(g_i),\text{LM}(g_j))\succ\text{LM}\left(h_t^{ij}g_t\right)$$

for any $1\le t\le r$, *where* $S(g_i,g_j)$ denotes the *S*-polynomial *of* g_i and g_j, LM denotes leading monomial, and LCM stands for the least common multiple.

Buchberger Criterion (Ajwa, et al., (1995))

1. In the process of picking a pair $\{f_i,f_j\}$, choose a pair such that LCM (LM(f_i), LM(f_j)) is minimal among all the pairs.

2. If the LM$(f_i)$and LM$(f_j)$are relatively prime, then S-poly(f_i, f_j)reduces to zero and can be ignored. Thus pick a pair $\{f_i, f_j\}$such that LM$(f_i)$and LM$(f_j)$are not relatively prime.
3. If there is an element$f_k$of the basis such that the LM(f_k)divides LCM (LM(f_i), LM(f_j)) and if the S-poly$(f_i, f_k)$and the S-poly(f_j, f_k)have already been considered, then S-poly(f_i, f_j)reduces to zero and could be ignored.

Buchberger criterion states that a set G is a Grobner basis if and only if all its S-polynomials have normal form zero. In particular, the modified Buchberger's algorithm after refinement is more efficient than the original Buchberger's algorithm.

FGLM and Gröbner Walk

The computation of Gröbner bases varies substantially when we use different monomial orderings. The time required for computing a Gröbner basis can grow drastically with the degree, size, and number of input polynomials, as well as the number of variables. Pure lexicographic order leads to a triangular system for solving the original system. However pure lexicographic order frequently requires a large amount of computation. On the other hand, it is possible to compute a grevlex Gröbner basis first and then converting it to a lex basis using the FGLM basis conversion algorithm which was constructed by Faugère et al., (1993). This change of basis algorithm can be utilized for solving zero-dimensional systems of equations (Decker and Lossen (2006)) and is considered to be more efficient than the first and modified Buchberger algorithms. Another method for Gröbner bases conversion is the Gröbner Walk. Both algorithms are implemented in Maple, and can be applied to address the question of an efficient elimination method using a suitable order by using basis conversion.

The Gröbner Walk and FGLM algorithms convert a Gröbner basis of commutative polynomials from one monomial order to another. They are frequently applied when a Gröbner basis is too difficult to compute directly. The Walk command (applied in Maple) takes a Gröbner basis G with respect to a monomial order T, and returns the reduced Gröbner basis for G with respect to T2 (command sequence: Walk(G, T, T2)). Walk supports the following types of monomial orders: 'plex', 'grlex', 'tdeg', 'wdeg', 'lexdeg', 'matrix', and products of these orders formed using 'prod'. Unlike FGLM, the ideal defined by G can be of any dimension. The Gröbner walk is typically not as fast as FGLM on zero-dimensional ideals. The Walk command does not check that G is indeed a Gröbner basis with respect to T. One example of code using Walk command in Maple 10 is illustrated below:

Example 5

```
>F:=[10*x*z-6*x^3-8*y^2*z^2,'-6*z+5*y^3]:
>G:=Basis(F1,tdegree(x,y,z)):
>Walk(G,tdeg(x,y,z),plex(x,y,z));
```

$$-6z+5y^3,-5xz+3x^3+4y^2z^2$$

Unfortunately, although FGLM command also takes a Gröbner basis G with respect to a monomial order T, and returns the reduced Gröbner basis for G with respect to T2 (command sequence: FGLM(G, T, T2)), but the ideal defined by G must be zero-dimensional. Otherwise, the algorithm will not terminate. FGLM supports the same monomial orders as Walk but cannot convert to 'wdeg' or 'matrix' orders, or to any product order that makes use of one of these orders. These orders are perfectly valid as a starting point. The FGLM command does not check that G is indeed a Gröbner basis with respect to T. If no truncation order is specified, then it checks whether G is a zero-dimensional system. This check may fail if G is not a Gröbner basis. An example of using FGLM command in Maple 10 is given below.

Example 6

```
>F:=[x^3+x*y-y^2+1,y^3-x*y+x] ;
>G:=Basis(F,tdeg(x,y));
```

$$G:=[y^3-xy+x,x^3+xy-y^2+1]$$

```
>FGLM(G, tdeg(x,y), plex(x,y));
```

$$[\,y^9+y^6-3y^5+4y^4-2y^3-2y^2+3y-1,x+y^8+y^6+2y^5-y^4+3y^3+y^2-2y+1]$$

For some large systems with big coefficients, a lexicographic Gröbner basis cannot be computed, even with FGLM (Faugère et al., (1993)). Nevertheless, such systems can be solved in Maple on the Katsura-4 system applying a rational univariate representation (Monagan et al.,(2005).

Example 7

```
> katsura4 := [2*x*t+2*u*y+2*z*t-y, 2*t+u+2*x+2*y+2*z-1,
2*t*u+x*y+2*z*t+2*y*z-t, t^2+2*y*t+2*z*u+2*x*z-z,
2*t^2+u^2+2*x^2+2*y^2+2*z^2-u]:
G := Basis(katsura4, tdeg(x,y,z,t,u)):map(length@maxnorm, G);
      [1, 1, 2, 2, 2, 3, 4, 4, 4, 12, 12, 12, 22]
```

Here, the largest coefficients appearing in the total degree Gröbner basis have 22 digits. The coefficients in the rational univariate representation will be of a similar order of magnitude, so we can run FGLM modulo a sufficiently large prime, and then apply rational reconstruction to compute the result. The algorithm described in the following section is another iriprovement of Buchberger algorithm.

F_4 Algorithm

Gröbner bases provide a general tool for studying arbitrary polynomial ideals, eliminating variables as well as finding the common roots of a system. However, the large coefficient size and the degree of the basis polynomials to be solved are the main limitations for Gröbner bases. In fact, the time complexity of the algorithm makes it impractical. This is due to the repetitions of the process during the elimination. Moreover, in Knuth (1981) and Moller (1993), it is stressed that the worst case of Buchberger's algorithm is known to run in double exponential time, and on average its running time seems to be single exponential. In particular, for solving sparse system, the complexity of Gröbner bases techniques is still impractical and hard to determine.

Later, in 1991, Faugère proposed a new efficient algorithm for computing Gröbner bases, which is known as F_4 algorithm, and Farr, and Pearce (2005) discussed its various strategies. The F_4 algorithm consists of a very simple improvement; one runs the Buchberger algorithm but at each step selects multiple syzygies. They are placed into a common matrix along with any rows that are needed for the reduction process, and this matrix is triangularized. The rows with new pivots correspond to new polynomials, which are then added to the basis. In particular, one should select all of the syzygies of smallest degree at each step of the algorithm, and reuse rows from previously reduced matrices where possible. This detail description can be summarized that F_4 algorithm is simultaneous reduction of all polynomials and a combination of Buchberger criteria and very efficient linear algebra. Continuation from F_4 algorithm, Faugère comes out with another algorithm named F_5 algorithm that construct matrices iteratively on the degree and on the number of equations, and replace Buchberger criteria with new criteria to avoid reduction to zero, as described in Decker and Lossen (2006) and Faugère (2002).

One way to visualize the F_4 algorithm is to consider the reduction of a single S-polynomial in the Buchberger algorithm. Consider the example in Faugère (2002) as shown below:

Example 8

Let $G = [x^2 + y, xy^2 - xy, y^3 - 1]$ where $G_1 = x^2 + y$, $G_2 = xy^2 - xy$, $G_3 = y^3 - 1$, and consider the syzygy $S_{1,2} = x^2y + y^3$ under graded lex order. In the division algorithm, $S_{1,2}$ is reduced first by subtracting yG_1 and then by subtracting G_3, as shown below.

$$x^2y + y^3 \rightarrow x^2y + y^3 - y\,(x^2 + y) = y^3 - y^2 \rightarrow y^3 - y^2 - (y^3 - 1) \quad = -y^2 + 1.$$

The key observation is that this reduction process is equivalent to a matrix triangularization. In the example below, the columns of the matrix correspond to the monomials $[x^2y, y^3, y^2, 1]$, while the rows contain $S_{1,2}$, y G_1, and G_3, respectively. Examining the reduced matrix on the right, we find one new pivot belonging to $y^2 - 1$.

$$\begin{bmatrix} 1 & 1 & 0 & 0 \\ 1 & 0 & 1 & 0 \\ 0 & 1 & 0 & -1 \end{bmatrix} \rightarrow \begin{bmatrix} 1 & 1 & 0 & 0 \\ 0 & 1 & -1 & 0 \\ 0 & 0 & 1 & -1 \end{bmatrix}$$

From this perspective we can detect the weakness of the Buchberger algorithm. It selects syzygies one by one, and for each one it triangularizes an entire matrix.

However, this potential improvement is not fully realized because the next improvement, computing modulo a number of primes, offsets some of the advantages. In such algorithm, the matrix will be reduced modulo a number of primes until the desired rows can be recovered using Chinese remaindering and rational reconstruction. Over algebraic function fields, sparse rational function interpolation can also be used so that the cost of recovering each row becomes significantly higher. Unfortunately, in any case the best strategy seems to be a hybrid approach. That is, after the initial reductions modulo a prime, one can identify rows with new pivots and further reduce them using the rest of the matrix. These sparse rows are easier to reconstruct, and as a side effect one computes the reduced Gröbner bases automatically. In 2005, Far and Pearce have been working on a more robust implementation of F_4 for Maple 10 (Faugère, 2002).

Conclusion and Future Work

Computational methods for manipulating sets of polynomial equations are becoming of greater importance due to the use of polynomial equations in various applications. In some cases we need to eliminate variables from a given system of polynomial equations to obtain a symbolically smaller system, while in others we desire to compute the numerical solutions of non-linear polynomial equations.

The method of Gröbner bases deals with ideals and varieties. Apart from that, the method depends on the monomial order in the polynomial representation and the computational complexity of the method is still impractical and hard to determine for large sized systems. When it comes to practice, Gröbner bases method is slow and not effective for a variety of reasons compared to resultants as in Kapur and Saxena (1995).. However, Gröbner bases method can be applied to arbitrary systems of polynomial equations as compared to resultant based techniques, which basically deal with generic systems. The method of resultant involves computing determinants of large matrices besides having to reduce or eliminate the presence of extraneous factors in its computation. Numerous research articles concerning the computational problems in solving polynomial systems can be found.

Combining elimination with numerical root finding can have potentially severe difficulties when the approach is implemented on computers using finite precision arithmetic. To handle this problem, it is possible to apply algebraic tools based on the algebraic structure of the quotient rings $k[x_1,...,x_n]/I$. By applying these tools from algebraic geometry, alternative numerical methods for approximating solutions of polynomial systems can be developed considering real root counting and root isolation to get a better approximation of the desired solutions.

References

Adams, W.W and Loustaunau, P. (1994). *An Introduction to Gröbner Bases.* In Volume 3 of Graduate Studies in Mathematics. American Mathematical Society.

Ajwa, I.A., Liu, Z., and Wang, P.S. (1995). *Gröbner Bases Algorithm.* In Proceedings of ICM, 1-15.

Aubry, P., Moreno Maza, M. (1997). Triangular sets for solving polynomial systems: a comparative implementation of four methods. J. Symb. Comput., 28, 125 – 154.

Becker, T., Weispfenning, V. (1993). *Gröbner Bases. A Computational Approach to Commutative Algebra.* Undergraduate Texts in Mathematics. 141 Springer-Verlag. New York.

Buchberger, B. (1985). *Gröbner Bases: An Algorithmic Method in Polynomial Ideal Theory.* Recent Trends in Multidimensional Systems Theory, Reider ed. Bose, 1985.

Buchberger, B. (1989). *Applications of Gröbner Bases in Nonlinear Computational Geometry.* In Geometric Reasoning, Editors: Kapur, D., and Mundy, J., 415-447. MIT Press.

Canny, J., and Emiris, I. (1993). *An Efficient Algorithm for the Sparse Mixed Resultant.* In Proceeding of AAECC.

Cox, D., Little, J., and O'Shea, D. (1996). *Ideal, Varieties, and Algorithms. An Introduction to Computational Algebraic Geometry and Commutative Algebra.* Undergraduate Texts in Mathematics. Springer-Verlag, New York, Second edition.

Cox, D., Little, J., and O'Shea, D. (1998). *Using Algebraic Geometry*, Volume 185 of Graduate Texts in Mathematics. Springer-Verlag, New York.

Decker,W., and Lossen, C. (2006). *Computing in Algebraic Geometry, Algorithms and Computation in Mathematics.* Volume 16. Springer-Verlag, Berlin.

Emiris I . (1994). *Sparse Elimination and Applications in Kinematics*, Doctoral Thesis, Computer Science Division, University of California., Berkeley.

Faugère, J.C. (2002). *A New Efficient Algorithm for Computing Gröbner Bases Without Reduction to Zero (F_5).* In Proceedings of ISSAC, ACM Press, 75-83.

Faugère, J.C. (1999). *A New Efficient Algorithm for Computing Gröbner Bases (F4), Effective Methods in Algebraic Geometry (Saint-Malo, 1998).* Journal of Pure and Applied Algebra, 139, no. 1-3, 61–88.

Faugère, J.C., Gianni, P., Lazard, D., and Mora, T. (1993). *Efficient Computation of Zero- Dimensional Gröbner Bases By Change of Ordering.* Journal of Symbolic Computation, 16, 329–344.

Farr, J.B., and Pearce, R. (2005). *Working with Multivariate Polynomials in Maple.* In Proceedings of Maple Summer Workshop.

Greuel, G.M. (2000). *Computer Algebra and Algebraic Geometry – Achievements and Perspectives.* Journal of Symbolic Computation, 30, 253-289.

Gianni, P., Trager, B., and Zacharias, G. (1988). *Gröbner Bases and Primary Decompositions of Polynomial Ideals.* Journal of Symbolic Computation, 6,149–167.

Kapur, D., Saxena, T., and Yang, L. (1994). *Algebraic and Geometric Reasoning Using Dixon Resultants.* In Proc. Internat. Symp. Symbolic Algebraic Comput. ISSAC'94, 99-107.

Kapur, D., and Saxena, T. (1995). Comparison of various multivariate resultant formulations. ISSAC'95. Montreal Canada, ACM, 189 – 194.

Knuth, D.E. (1981). *The Art of Computer Programming, Vol. 2, Seminumerical Algorithm.* 2nd Edition. Addison Wesley, Reading, MA.

Lebrun, J., and Selesnick, I. (2001). Gröbner bases and wavelet design. Joutnal of Symb. Comput. 37 (2004), 227 – 259.

Manocha, D. (1994). *Solving Systems of Polynomial Equations.* IEEE Computer Graphic and Applications.

Marinari, M.G., Moller, H.M., and Mora, T. (1996). *On Multiplicities in Polynomial System Solving.* Trans. Amer. Math. Soc., 348, no. 8, 3283-3321.

Monagan, M.B., and Pearce, R. (2004). *The Polynomial Ideals Maple package.* In Proceedings of Maple Summer Workshop.

Monagan, M.B., Geddes, K.O., Heal, K.M., Labahn, G., Vorkoetter, S. M., McCarron, J., DeMarco, P. (2005). *Maple (10) Introductory Programming Guide,* Maplesoft Waterloo Maple Inc.

Moller, H.M. (1993). *On Decomposing Systems of Polynomial Equations with Finitely Many Solutions.* AAECC. 4:217-30.

Pohst, M.E., and Yun, D.Y.Y. (1981). *On Solving System of Algebraic Equations via Ideal Bases and Elimination Theory.* In Proceedings Symposium on Symbolic and Algebraic Manipulations.

Steel, A. Gröbner Basis Timings Page, available from *http: //magma.maths.usyd.edu.au/users/allan/gb.*

Stewart, I. (2003). Galois Theory – 3rd edition. Chapman & Hall/CRC mathematics. ISBN 1-58488 – 393 – 6.

Wang, D.M. (1993). An elimination method for polynomial systems. *J. Symb. Comput.*, 16, 83 – 114.