The geometry of multivariate polynomial division and elimination — Source link

Kim Batselier, Philippe Dreesen, Bart De Moor

Published on: 14 Feb 2013 - SIAM Journal on Matrix Analysis and Applications (Society for Industrial and Applied Mathematics)

Topics: Elimination theory, Gröbner basis, Cylindrical algebraic decomposition, Algebraic geometry and Linear algebra

THE GEOMETRY OF MULTIVARIATE POLYNOMIAL DIVISION AND ELIMINATION [*]

KIM BATSELIER[†], PHILIPPE DREESEN[†], AND BART DE MOOR [†]

Abstract. Multivariate polynomials are usually discussed in the framework of algebraic geometry. Solving problems in algebraic geometry usually involves the use of a Gröbner basis. This article shows that linear algebra without any Gröbner basis computation suffices to solve basic problems from algebraic geometry by describing three operations: multiplication, division and elimination. This linear algebra framework will also allow us to give a geometric interpretation. Multivariate division will involve oblique projections and a link between elimination and principal angles between subspaces (CS decomposition) is revealed. The main computational tool in this approach is the QR decomposition.

Key words. multivariate polynomial division, oblique projection, multivariate polynomial elimination, QR decomposition, CS decomposition, sparse matrices, principal angles

AMS subject classifications. 15A03,15B05,15A18,15A23

1. Introduction.

Traditionally, multivariate polynomials are discussed in terms of algebraic geometry. A major computational advance was made with the discovery of the Gröbner basis and an algorithm to compute them by Buchberger in the 1960's [8]. This has sparked a whole new line of research and algorithms in computer algebra. Applications of multivariate polynomials are found in robotics [13], computational biology [31], statistics [15], signal processing and systems theory [7, 9, 10, 16]. In these applications, Gröbner bases are the main computational tool and most methods to compute these are symbolic. The aim of this article is to explore the natural link between multivariate polynomials and numerical linear algebra. The goal is in fact to show that basic knowledge of linear algebra enables one to understand the basics of algebraic geometry and already solve problems without the computation of any Gröbner basis. The main motivation to use numerical linear algebra is the existence a well-established body of numerically stable methods. It is also a natural framework in which computations on polynomials with inexact coefficients can be described. In this article we discuss multivariate polynomial multiplication, division and elimination from this numerical linear algebra point of view. An interesting result of this approach is that it becomes possible to interpret algebraic operations such as multivariate polynomial division and elimination geometrically. Furthermore, these geometrical interpretations do not change when the degree of the polynomials changes or a different monomial ordering is chosen. Applications are not limited to divisions and elimination. Several other problems which are traditionally discussed

[*]Kim Batselier is a research assistant at the Katholieke Universiteit Leuven, Belgium. Philippe Dreesen is supported by the Institute for the Promotion of Innovation through Science and Technology in Flanders (IWT-Vlaanderen). Bart De Moor is a full professor at the Katholieke Universiteit Leuven, Belgium. Research supported by Research Council KUL: GOA/11/05 Ambiorics, GOA/10/09 MaNet, CoE EF/05/006 Optimization in Engineering (OPTEC) en PFV/10/002 (OPTEC), IOF-SCORES4CHEM, several PhD/postdoc & fellow grants; Flemish Government: FWO: PhD/postdoc grants, projects: G0226.06 (cooperative systems and optimization), G0321.06 (Tensors), G.0302.07 (SVM/Kernel), G.0320.08 (convex MPC), G.0558.08 (Robust MHE), G.0557.08 (Glycemia2), G.0588.09 (Brain-machine) research communities (WOG: ICCoS, ANMMM, MLDM); G.0377.09 (Mechatronics MPC) IWT: PhD Grants, Eureka-Flite+, SBO LeCoPro, SBO Climaqs, SBO POM, O&O-Dsquare Belgian Federal Science Policy Office: IUAP.

[†]Department of Electrical Engineering ESAT-SCD, KU Leuven / IBBT Future Health Department, 3001 Leuven, Belgium

in algebraic geometry such as the ideal membership problem and finding the roots of a multivariate polynomial system [12, 13] can also be viewed from this point of view. For example, Stetter [33, 34] demonstrated the link between solving multivariate polynomial systems and eigenvalue problems but still relies however on the computation of a Gröbner basis. Another problem which has already received a lot of attention from a numerical linear algebra point of view is the computation of the greatest common divisor of two polynomials with inexact coefficients [6, 11, 17, 44]. We now briefly discuss the main two algebraic operations that will be the focus of this article.

Multivariate polynomial division is the essential operation for computing the Gröbner bases of a multivariate polynomial system. A significant step in showing the link between multivariate polynomial division and linear algebra was the development of the F4 algorithm due to Faugère [18]. This method computes a Gröbner basis by means of Gaussian elimination. The method itself however 'emulates' polynomial division in the sense that it does not compute any quotients but only a remainder. The matrix that is reduced in this algorithm contains a lot of zeros and therefore sparse matrix techniques are used. Like F4, all implementations of polynomial division are found in computer algebra systems [20, 29]. In this article, multivariate polynomial division will be interpreted as a vector decomposition whereby the divisors and the remainder are described by elements of the row spaces of certain matrices. It will be shown that either can be found from an oblique projection and no row reductions are necessary. The main computational tool in our implementation is the QR decomposition.

Multivariate polynomial elimination was originally studied by Bézout, Sylvester, Cayley and Macaulay in the 1800's using determinants, also called resultants [26, 41]. This work formed the inspiration for some resultant-based methods to solve polynomial systems [2, 21, 28]. The advent of the Gröbner basis also made it possible to eliminate variables when using a lexicographic monomial ordering. A method which is also based entirely on linear algebra for multivariate polynomial elimination relies on the computation of the kernel of a matrix [43]. In this article the link between multivariate polynomial elimination and principal angles between subspaces is revealed. The main computational tool will be the QR decomposition together with an implicitly restarted Arnoldi iteration. All numerical examples were computed on a 2.66 GHz quad-core desktop computer with 8 GB RAM in MATLAB [32].

This article is structured as follows. Section 2 introduces some notation and basic concepts on the vector space of multivariate polynomials. Section 3 describes the operation of multivariate polynomial multiplication. This will turn out to be a generalization of the discrete convolution operation to the multivariate case. In Section 4 multivariate polynomial division is worked out as a vector decomposition and an algorithm together with a numerical implementation is given. Finally, Section 5 describes the multivariate polynomial elimination problem as finding the intersection of two subspaces and the link is made with the cosine-sine decomposition. An elimination algorithm and implementation is also provided.

2. Vector Space of Multivariate Polynomials.

It is easy to see that the set of all multivariate polynomials over n variables up to degree d over the field of complex numbers $\mathbb{C}$ together with the addition and multiplication with a scalar form a vector space. This vector space will be denoted by $\mathcal{C}_d^n$. A canonical basis for this vector space consists of all monomials from degree 0 up to d. Since the total number of monomials in n variables from degree 0 up to degree d is given by

$$q = \binom{d+n}{n}$$

it follows that $\dim(\mathcal{C}_d^n) = q$. The degree of a monomial $x^a = x_1^{a_1} \ \ldots \ x_n^{a_n}$ is defined as $|a| = \sum_{i=1}^n a_i$. The degree of a polynomial p, $\deg(p)$, then corresponds with the degree of the monomial of p with highest degree. It is possible to order the terms of multivariate polynomials in different ways and results typically depend on which ordering is chosen. It is therefore important to specify which ordering is used. For a formal definition of monomial orderings together with a detailed description of some relevant orderings in computational algebraic geometry see [12, 13]. In the next paragraph the monomial ordering which will be used throughout the whole of this article is defined.

2.1. Monomial Orderings. Note that we can reconstruct the monomial $x^a = x_1^{a_1} \ \ldots \ x_n^{a_n}$ from the n-tuple of exponents $a = (a_1, \ \ldots \ , a_n) \in \mathbb{N}_0^n$. Furthermore, any ordering $>$ we establish on the space $\mathbb{N}_0^n$ will give us an ordering on monomials: if $a > b$ according to this ordering, we will also say that $x^a > x^b$.

Definition 2.1. *Graded Xel Order. Let a and $b \in \mathbb{N}_0^n$. We say $a > b$ if*

$$|a| = \sum_{i=1}^n a_i > |b| = \sum_{i=1}^n b_i, \ or\, |a| = |b| \ and\ a >_{xel} b$$

where $a >_{xel} b$ if, in the vector difference $a - b \in \mathbb{Z}^n$, the leftmost nonzero entry is negative.

Example 2.1. $(2,0,0) > (0,0,1)$ *because* $|(2,0,0)| > |(0,0,1)|$ *which implies* $x_1^2 > x_3$. *Likewise,* $(0,1,1) > (2,0,0)$ *because* $(0,1,1) >_{xel} (2,0,0)$ *and this implies that* $x_2x_3 > x_1^2$.

The ordering is graded because it first compares the degrees of the two monomials and applies the xel ordering when there is a tie. Once a monomial ordering $>$ is chosen we can uniquely identify the monomial with largest degree of a polynomial f according to $>$. This monomial is called the leading monomial of f and is denoted by $\mathrm{LM}(f)$. A monomial ordering also allows for a multivariate polynomial f to be represented by its coefficient vector. One simply orders the coefficients in a row vector, graded xel ordered, in ascending degree. The following example illustrates.

Example 2.2. *The polynomial* $f = 2 + 3x_1 - 4x_2 + x_1x_2 - 8x_1x_3 - 7x_2^2 + 3x_3^2$ *in* C_3^2 *is represented by the vector*

$$\begin{array}{cccccccccc} 1 & x_1 & x_2 & x_3 & x_1^2 & x_1x_2 & x_1x_3 & x_2^2 & x_2x_3 & x_3^2 \\ \big(2 & 3 & -4 & 0 & 0 & 1 & -8 & -7 & 0 & 3\ \big) \end{array}$$

where the graded xel ordering is indicated above each coefficient.

By convention a coefficient vector will always be a row vector. Depending on the context we will use the label f for both a polynomial and its coefficient vector. $(.)^T$ will denote the transpose of the matrix or vector (.). Having established the representation of multivariate polynomials by row vectors we now proceed to discuss three basic operations: multiplication, division and elimination.

3. Multivariate Polynomial Multiplication. Given two polynomials h and $f \in \mathcal{C}_d^n$, then their product hf does not lie in $\mathcal{C}_d^n$ anymore. It is easy to derive that polynomial multiplication can be written in this framework as a vector matrix product. Supposing $\deg(h) = m$ we can write

$$\begin{aligned} h\,f &= (h_0 \,+\, h_1\,x_1 \,+ h_2\,x_2 \,+ \ldots + h_k\,x_n^m)\,f \\ &= h_0\,f \,+ h_1\,x_1\,f \,+ h_2\;x_2\,f \,+ \ldots + h_k\,x_n^m\,f. \end{aligned}$$

This can be written as the following vector matrix product

$$h\,f = \begin{pmatrix} h_0 & h_1 & \ldots & h_m \end{pmatrix} \begin{pmatrix} f \\ x_1\,f \\ x_2\,f \\ \vdots \\ x_n^m\,f \end{pmatrix} \tag{3.1}$$

where each row of the matrix in the right hand side of (3.1) is the coefficient vector of $f, x_1 f, x_2 f, \ldots, x_n^m f$ respectively and x_n^m is $\mathrm{LM}(h)$. The multiplication of f with a monomial results in all coefficients of f being shifted to the right in its corresponding coefficient vector. Therefore the matrix which is built up from the coefficients of f in expression (3.1) is a quasi-Toeplitz matrix. In the univariate case this multiplication matrix corresponds with the discrete convolution operator which is predominantly used in linear systems theory. The polynomial f is then interpreted as the impulse response of a linear time-invariant system and h as the input signal. In this case, assuming $\deg(f) = n$, writing out (3.1) results in

$$h\,f = \begin{pmatrix} h_0 & h_1 & \ldots & h_m \end{pmatrix} \begin{pmatrix} f_0 & f_1 & f_2 & \ldots & f_n & 0 & 0 & \ldots & 0 \\ 0 & f_0 & f_1 & f_2 & \ldots & f_n & 0 & \ldots & 0 \\ 0 & 0 & f_0 & f_1 & f_2 & \ldots & f_n & \ldots & 0 \\ \vdots & \vdots & \vdots & \ddots & \ddots & \ddots & \ddots & \ddots & \vdots \\ 0 & 0 & 0 & \ldots & f_0 & f_1 & f_2 & \ldots & f_n \end{pmatrix}$$

where the multiplication operator is now a Toeplitz matrix. The following example illustrates the multiplication of two polynomials in $\mathcal{C}_2^2$.

Example 3.1. *$k = x_1^2 + 2x_2 - 9$ and $l = x_1x_2 - x_2$. The leading monomial of k is x_1^2. The multiplication is then given by*

$$\begin{pmatrix} -9 & 0 & 2 & 1 \end{pmatrix} \begin{pmatrix} l \\ x_1 l \\ x_2 l \\ x_1^2 l \end{pmatrix}.$$

The multiplication operator is then

$$\begin{array}{c} \\ l \\ x_1l \\ x_2l \\ x_1^2l \end{array} \begin{array}{c} \begin{array}{ccccccccccccccc} 1 & x_1 & x_2 & x_1^2 & x_1x_2 & x_2^2 & x_1^3 & x_1^2x_2 & x_1x_2^2 & x_2^3 & x_1^4 & x_1^3x_2 & x_1^2x_2^2 & x_1x_1^3 & x_2^4 \end{array} \\ \begin{pmatrix} 0 & 0 & -1 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & -1 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & -1 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & -1 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \end{pmatrix} \end{array}$$

where the columns were labelled according to the graded xel monomial ordering and the labels on the left indicate with which monomial l was multiplied. Multiplying this matrix with the coefficient vector of k on the left results in the vector

$$\begin{array}{ccccccccccccccc} 1 & x_1 & x_2 & x_1^2 & x_1x_2 & x_2^2 & x_1^3 & x_1^2x_2 & x_1x_2^2 & x_2^3 & x_1^4 & x_1^3x_2 & x_1^2x_2^2 & x_1x_1^3 & x_2^4 \\ \left(0\right. & 0 & 9 & 0 & -9 & -2 & 0 & -1 & 2 & 0 & 0 & 1 & 0 & 0 & \left.0\right) \end{array}$$

which is indeed the coefficient vector of $k\,l$.

The description of multiplication of multivariate polynomials in this linear algebra framework therefore leads in a natural way to the generalization of the convolution operation to the multidimensional case [6, 30]. In the same way, multivariate polynomial division will generalize the deconvolution operation.

4. Multivariate Polynomial Division.

For multivariate polynomial division it will be necessary to describe, for a given polynomial $p \in \mathcal{C}_d^n$, a sum of the form $h_1 f_1 + \ldots + h_s f_s$ where $h_1, \ldots, h_s, f_1, \ldots, f_s \in \mathcal{C}_d^n$ and where for which each $h_i f_i$ $(i = 1 \ldots s)$ the condition $\text{LM}(p) \geq \text{LM}(h_i f_i)$ applies. These sums will be described by the row space of the following matrix.

DEFINITION 4.1. *Given a set of polynomials $f_1, \ldots, f_s \in \mathcal{C}_d^n$, each of degree d_i $(i = 1 \ldots s)$ and a polynomial $p \in \mathcal{C}_d^n$ of degree d then the divisor matrix D is given by*

$$D = \begin{pmatrix} f_1 \\ x_1 f_1 \\ x_2 f_1 \\ \vdots \\ x_n^{k_1} f_1 \\ f_2 \\ x_1 f_2 \\ \vdots \\ x_n^{k_2} f_2 \\ \vdots \\ x_n^{k_s} f_s \end{pmatrix} \tag{4.1}$$

where each polynomial f_i is multiplied with all monomials x^{α_i} from degree 0 up to degree $k_i = deg(p) - deg(f_i)$ such that $LM(x^{\alpha_i} f_i) \leq LM(p)$.

Indeed, the row space of this D are all polynomials $\sum_{i=1}^{s} h_i f_i$ of degree $d = \deg(p)$ such that $\text{LM}(p) \geq \text{LM}(h_i f_i)$. The vector space spanned by the rows of D will be denoted $\mathcal{D}$. It is clear that $\mathcal{D} \subset \mathcal{C}_d^n$ and that $\dim(\mathcal{D}) = \text{rank}(D)$. Each column of D contains the coefficient of a certain monomial and hence the number of columns of D, $\#\text{col}(D)$, corresponds with $\dim(\mathcal{C}_d^n)$. This divisor matrix will be the key to generalize multivariate polynomial division in terms of linear algebra.

Everybody is familiar with the polynomial division for the univariate case. It is therefore quite surprising that this was generalized to the multivariate case only 40 years ago [13]. Let us start with the formal definition.

DEFINITION 4.2. *Fix any monomial order $>$ on $\mathcal{C}_d^n$ and let $F = (f_1, \ldots, f_s)$ be a s-tuple of polynomials in $\mathcal{C}_d^n$. Then every $p \in C_d^n$ can be written as*

$$p = h_1 f_1 + \ldots + h_s f_s + r \tag{4.2}$$

where $h_1, \ldots, h_s, r \in \mathcal{C}_d^n$. For each $i, h_i f_i = 0$ or $LM(p) \geq LM(h_i f_i)$, and either $r = 0$ or r is a linear combination of monomials, none of which is divisible by any of $LM(f_1), \ldots, LM(f_s)$.

The generalization lies obviously in extending the polynomials p and f in the univariate case to elements of $\mathcal{C}_d^n$ and sets of divisors F. The constraint on the remainder term for the univariate case, $\deg(r) < \deg(f)$, is also generalized. The biggest consequence of this new constraint is that the remainder can have a degree which is strictly higher than any of the divisors f_i. It now becomes clear why the divisor matrix was defined. The $h_i f_i$ terms of (4.2) are in this framework described by the row space $\mathcal{D}$ of the divisor matrix. This allows us to rewrite (4.2) as the following vector equation

$$p \;=\; h\,D + r$$

which leads to the insight: multivariate polynomial division corresponds with a vector decomposition. The vector p is decomposed into $h\,D$, which lies in $\mathcal{D}$, and into r. Since p can be any element of $\mathcal{C}_d^n$ and $\mathcal{D}$ is a subspace of $\mathcal{C}_d^n$ it therefore follows that there exists a vector space $\mathcal{R}$ such that $\mathcal{D} \oplus \mathcal{R} = \mathcal{C}_d^n$. In general there are many other subspaces $\mathcal{R}$ which are the complement of $\mathcal{D}$. The most useful $\mathcal{R}$ for multivariate polynomial division will be the vector space which is isomorphic with the quotient space $\mathcal{C}/\mathcal{D}$.

4.1. Quotient Space. Having defined the vector space $\mathcal{D}$ one can now consider the following relationship, denoted by $\sim$, in $\mathcal{C}_d^n$:

$$\forall\, p, r \in \mathcal{C}_d^n : p \sim r \Leftrightarrow p - r \in \mathcal{D}.$$

It is easily shown that $\sim$ is a equivalence relationship and therefore $\mathcal{C}_d^n$ is partitioned. Each of these partitions is an equivalence class

$$[p]_{\mathcal{D}} \;=\; \{r \in \mathcal{C}_d^n : p - r \in \mathcal{D}\}.$$

Since $p - r \in \mathcal{D}$, equation (3.1) tells us that this can be written as $h\,D$ and therefore

$$p \;=\; h\,D + r.$$

Hence the addition of the constraint that either $r = 0$, or r is a linear combination of monomials, none of which is divisible by any of $\mathrm{LM}(f_1), \ldots, \mathrm{LM}(f_s)$ allows then for the interpretation of the elements of the equivalence class as the remainders. The set of all the equivalence classes $[p]_{\mathcal{D}}$ is denoted by $\mathcal{C}/\mathcal{D}$ and is also a vector space. In fact, one can find a vector space $\mathcal{R} \subset \mathcal{C}_d^n$, isomorphic with $\mathcal{C}/\mathcal{D}$, such that $\mathcal{D} \oplus \mathcal{R} = \mathcal{C}_d^n$. This implies that

$$\begin{aligned}
\dim(\mathcal{R}) &= \dim(\mathcal{C}/\mathcal{D}) \\
&= \dim(\mathcal{C}_d^n) - \dim(\mathcal{D}) \\
&= \#col(D) - \mathrm{rank}(D) \\
&= \mathrm{nullity(D)}
\end{aligned}$$

which allows to determine the dimension of $\mathcal{R}$ in a straightforward manner. $\mathcal{R}$ being a finite-dimensional vector space implies that a basis can be formally defined.

Definition 4.3. *Any set of monomials which forms a basis of a vector space $\mathcal{R}$ such that $\mathcal{R} \cong \mathcal{C}/\mathcal{D}$ and $\mathcal{R} \subset \mathcal{C}_d^n$ is called a normal set. The corresponding canonical basis of $\mathcal{R}$ in $\mathcal{C}_d^n$ is denoted R such that $\mathcal{R} = row(R)$.*

Since $\mathcal{R} \subset \mathcal{C}^n_d$, the canonical basis R needs to be a monomial basis. These basis monomials (or standard monomials) are in fact representatives of the equivalence classes of a basis for $\mathcal{C}/\mathcal{D}$. Although a polynomial basis could be chosen for R this would make it significantly harder to require that every monomial of this basis should not be divisible by any of the leading monomials of $f_1, \ldots, f_s$. This will turn out to be easy for a monomial basis of R. Finding these standard monomials will translate itself into looking for a set of columns which are linearly dependent with respect to all other columns of the divisor matrix. Since $\dim(\mathcal{C}/\mathcal{D}) = \text{nullity}(D)$, it must be possible to find $\#\text{col}(D) - r$ linearly dependent columns with $r = \text{rank}(D)$. In the univariate case, D is by construction of full row rank and hence $r = d - d_0 + 1$. The number of linearly dependent columns is then $(d+1) - (d - d_0 + 1) = d_0$. This is in fact linked with the fundamental theorem of algebra which states that an univariate polynomial of degree d_0 over the complex field has d_0 solutions. In the multivariate case things are a bit more complicated. D is then in general neither of full row rank nor of full column rank. This implies a non-uniqueness of both the quotients and remainder.

4.2. Non-uniqueness of quotients. Suppose the rank of the matrix D is r. In general, the matrix will not be of full row rank and therefore there will be maximally $\binom{p}{r}$ possibilities of choosing r linearly independent rows. In practice, a basis for the row space of D is required for calculating the decomposition of p into $\sum_i h_i f_i$ terms. Therefore depending on which rows are chosen as a basis for $\mathcal{D}$ several decompositions are possible. Checking whether the quotients are unique hence involves a rank test of D. Note that Definition 4.2 does not specify any constraints on how to choose a basis for $\mathcal{D}$. In Subsection 4.5 it is explained how such a basis is chosen using a sparse rank-revealing QR decomposition. This non-uniqueness is expressed in computational algebraic geometry by the multivariate long division algorithm being dependent on the ordering of the divisors $f_1, \ldots, f_s$. Note however that the implementation described in Subsection 4.5 does not make the quotients unique as they will always depend on the ordering of $f_1, \ldots, f_s$ when constructing the divisor matrix D. In contrast, choosing a basis for $\mathcal{R}$ is constrained by its definition but not in a such way that only one possible basis is left.

4.3. Non-uniqueness of remainders. The constraint $\deg(r) < \deg(f)$ for the univariate case is replaced by $r = 0$, or r is a linear combination of monomials, none of which is divisible by any of $\text{LM}(f_1), \ldots, \text{LM}(f_s)$. This in general, is not sufficient to reduce the number of possible bases of $\mathcal{R}$ to only one. The following example illustrates this point.

Example 4.1. *Suppose $p = 9x_2^2 - x_1x_2 - 5x_2 + 6$ is divided by $f_1 = x_2 - 3$ and $f_2 = x_1x_2 - 2x_2$. Since $LM(p) = x_2^2$ one needs to construct the following divisor matrix*

$$D = \begin{array}{c} \\ f_1 \\ x_1 f_1 \\ x_2 f_1 \\ f_2 \end{array} \begin{array}{c} \begin{array}{cccccc} 1 & x_1 & x_2 & x_1^2 & x_1x_2 & x_2^2 \end{array} \\ \begin{pmatrix} -3 & 0 & 1 & 0 & 0 & 0 \\ 0 & -3 & 0 & 0 & 1 & 0 \\ 0 & 0 & -3 & 0 & 0 & 1 \\ 0 & 0 & -2 & 0 & 1 & 0 \end{pmatrix} \end{array}.$$

The null column corresponding with the monomial x_1^2 will surely be linearly dependent with respect to all other columns. The rank of D is 4 and any other column of D could be chosen as the second linearly dependent column. This gives the following set *of possible bases for $\mathcal{R}$: $\{\{1,x_1^2\},\{x_1,x_1^2\},\{x_2,x_1^2\},\{x_1^2,x_1x_2\},\{x_1^2,x_2^2\}\}$. The leading monomials of f_1 and f_2 are, according to the graded xel ordering, x_1x_2 and x_2 respectively. Therefore the set of possible bases for $\mathcal{R}$ is reduced to $\{\{1,x_1^2\},\{x_1,x_1^2\}\}$ since neither 1 nor x_1 are divisible by x_1x_2 or x_2. Note that since D is of full row rank this implies that the quotients h_1 and h_2 are unique. The matrix R corresponding with the normal set $\{1,x_1^2\}$ is*

$$R = \begin{array}{c} \begin{array}{cccccc} 1 & x_1 & x_2 & x_1^2 & x_1x_2 & x_2^2 \end{array} \\ \begin{pmatrix} 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 \end{pmatrix} \end{array}.$$

The row space of R is indeed such that $\mathcal{R}\oplus\mathcal{D}=\mathcal{C}_2^2$.

From the example it is clear that not every set of linearly dependent columns corresponds with a normal set which is suitable to describe multivariate polynomial division. The encoding of the graded monomial ordering in the columns of the divisor matrix allows us to find a suitable basis for $\mathcal{R}$ which satisfies the constraint that none of its monomials is divisible by any $\text{LM}(f_i)$ $(i = 1\ldots s)$. The key idea is to check each column for linear dependence with respect to all columns to its right, starting from the rightmost column. Before stating the main theorem we first introduce some notation and prove a needed lemma. In what follows a monomial will be called linear (in)dependent when its corresponding column of the divisor matrix D is linear (in)dependent with respect to another set of columns. Suppose the divisor matrix D has q columns. Then each column of D corresponds with a monomial $m_1,\ldots,m_q$ with $m_1 < m_2 < \ldots < m_q$ according to the monomial ordering. Suppose now that $\text{rank}(D) = r$ and therefore $c_r \triangleq q-r$ linearly dependent monomials can be found. We now introduce the following high-level algorithm which results in a special set of linearly dependent monomials. Note that in this algorithm each monomial label stands for a column vector of the divisor matrix D.

Algorithm 4.1. *Find a maximal set of linearly dependent monomials*
Input: *divisor matrix D*
Output: *a maximal set of linearly dependent monomials l*

```
l ← []
if $m_q = 0$ then
  $l ← [l, m_q]$
end if
for $i = q-1 : -1 : 1$ do
  if $m_i$ linearly dependent with respect to $\{m_{i+1},\ldots,m_q\}$ then
    $l ← [l, m_i]$
  end if
end for
```

Algorithm 4.1 finds a maximal set of monomials l which are linearly dependent with respect to all monomials to their right. We will label these c_r monomials of l as $l_1,\ldots,l_{c_r}$ such that $l_{c_r} < \ldots < l_2 < l_1$ according to the monomial ordering. The matrix D can then be visually represented as

$$D = \begin{array}{c} \begin{array}{ccccccccc} m_1 & \dots & l_{c_r} & \dots & l_k & \dots & l_1 & \dots & m_q \end{array} \\ \begin{pmatrix} & & \times & & \times & & \times & & \\ \dots & \dots & \times & \dots & \times & \dots & \times & \dots & \dots \\ & & \times & & \times & & \times & & \end{pmatrix} \end{array}.$$

EXAMPLE 4.2. *We revisit the divisor matrix of Example 4.1 and apply Algorithm 4.1. For this simple example checking the linear dependence was done using the svd-based 'rank' command in MATLAB. A monomial m_i was considered to be linearly dependent as soon as the rank did not increase when adding its column to the matrix containing $\{m_{i+1}, \dots, m_q\}$. It is easy to verify that this results in the following linearly dependent monomials: $l_1 = x_1^2, l_2 = 1$.*

The previous example indicates that Algorithm 4.1 produces the standard monomials of lowest degree. We now prove the following lemma.

LEMMA 4.4. *Given a divisor matrix D of rank r and the linearly dependent monomials $l_1, \dots, l_{c_r}$ found from Algorithm 4.1. Then any other set of c_r linearly dependent monomials $l'_1, \dots, l'_{c_r}$ with $l'_1 > l'_2 > \dots > l'_{c_r}$ satisfies the following conditions: $l'_1 \geq l_1, l'_2 \geq l_2, \dots, l'_{c_r} \geq l_{c_r}$.*

Proof. Let $\{l_k, \dots, m_q\}$ denote the set of all monomials from l_k up to m_q for a certain $k \in \{1, \dots, c_r\}$ and let q_1 denote the cardinality of $\{l_k, \dots, m_q\}$. From Algorithm 4.1 we know that $\{l_k, \dots, m_q\}$ contains k linearly dependent monomials and $q_1 - k$ linearly independent monomials. We now choose the largest k such that $l'_k < l_k$. $\{l_k, \dots, m_q\}$ will then contain at most $k-1$ l' monomials which implies that there are at least $q_1 - k + 1$ linearly independent monomials in $\{l_k, \dots, m_q\}$. This contradicts the fact that there are exactly $q_1 - k$ linearly independent monomials in $\{l_k, \dots, m_q\}$. □

This lemma states that the normal set which is found from Algorithm 4.1 is of minimal degree according to the monomial ordering. We can now prove the main theorem.

THEOREM 4.5. *Consider a divisor matrix D. Then a suitable monomial basis for $\mathcal{R}$ is found by Algorithm 4.1. None of the monomials corresponding with the linearly dependent columns found in this way are divisible by any of the leading monomials of $f_1, \dots, f_s$ and therefore serve as a basis for the vector space of remainder terms $\mathcal{R}$.*

Proof. Since $\mathcal{D} \oplus \mathcal{R} = \mathcal{C}_d^n$, any multivariate polynomial $p \in \mathcal{C}_d^n$ can be decomposed into $\sum_{i=1}^s h_i f_i \in \mathcal{D}$, spanned by a maximal set of linearly independent rows of D, and $r \in \mathcal{R}$, spanned by the monomials $l_1, \dots, l_{c_r}$ found from Algorithm 4.1. We can therefore write

$$p = \sum_{i=1}^{s} h_i f_i + r \quad \text{with } r = \sum_{i=1}^{c_r} a_i l_i \quad (a_i \in \mathbb{C}). \tag{4.3}$$

Suppose now that at least one of the monomials $l_1, \dots, l_{c_r}$ is divisible by a leading monomial of one of the polynomials $f_1, \dots, f_s$, say f_j. Let l_k be the monomial of highest degree which is divisible by $\mathrm{LM}(f_j)$. This implies that the division of $r - \sum_{i=1}^{k-1} a_i l_i$ by f_j can be written as

$$r - \sum_{i=1}^{k-1} a_i l_i = g f_j + r' \tag{4.4}$$

where $r' \neq 0$ and due to the definition (4.2) none of the monomials of r' are divisible by $\mathrm{LM}(f_j)$. In addition, all monomials $r'_1, \ldots, r'_t$ of r' satisfy $r'_i < l_k$ [13, p. 64-66]. By substituting (4.4) into (4.3) we have

$$\begin{aligned} p &= \textstyle\sum_{i=1}^{s} h_i\, f_i + r \\ &= \textstyle\sum_{i=1}^{s} h_i\, f_i + \sum_{i=1}^{k-1} a_i\, l_i + g f_j + r' \\ &= \textstyle\sum_{i=1}^{s} h'_i\, f_i + \sum_{i=1}^{k-1} a_i\, l_i + r'. \end{aligned}$$

From this last equation one can see that r' needs to contain $c_r - k + 1$ monomials none of which are divisible by any of the leading monomials of $f_1, \ldots, f_s$. If $\mathrm{LM}(r')$ is not divisible by any of the leading monomials of $f_1, \ldots, f_s$ then $\mathrm{LM}(r')$ is the new linearly dependent monomial l'_k. However, $l'_k < l_k$ which is a contradiction according to Lemma 4.4. If $\mathrm{LM}(r')$ is divisible by any of the leading monomials of $f_1, \ldots, f_s$ then the division procedure as in (4.4) can be repeated, leading to the same contradiction. ▯

The duality between the linearly dependent columns of D and the linearly independent rows of its kernel K implies the following corollary of Theorem 4.5.

Corollary 4.6. *A monomial basis for $\mathcal{R}$ can also be found from checking the rows of the kernel of D for linear independence from top to bottom. None of the monomials corresponding with the linearly independent rows are divisible by any of the leading monomials of $f_1, \ldots, f_s$.*

Corollary 4.6 will be useful when discussing a practical implementation. In computational algebraic geometry, the non-uniqueness of the remainder corresponds with the remainder being dependent on the order of the divisors $f_1, \ldots, f_s$. This is normally solved by computing the remainder of p being divided by a Gröbner basis instead. The difference between the Gröbner basis method and the algorithm described in this manuscript is discussed in Section 4.7. Note that the normal set which is found from Theorem 4.5 is also unique since changing the order of the divisors (rows) will not affect the linear dependence of the columns in Algorithm 4.1.

4.4. The Geometry of Polynomial Division. Having discussed the divisor matrix D and a canonical basis R for the quotient space it is now possible to interpret (4.2) geometrically. Since $p = \sum_{i=1}^{s} h_i f_i + r$ with $\sum_{i=1}^{s} h_i f_i \in \mathcal{D}$ and $r \in \mathcal{R}$, finding the $\sum h_i f_i$ terms is then equivalent to projecting p along $\mathcal{R}$ onto $\mathcal{D}$. The remainder r can then simply be found as $p - \sum_{i=1}^{s} h_i f_i$. Note that the remainder r can also be found from the projection of p along $\mathcal{D}$ onto $\mathcal{R}$. Figure 4.1 represents this in 3 dimensional Euclidean space. The whole 3 dimensional Euclidean space represents $\mathcal{C}_d^n$, the plane represents $\mathcal{D}$ and the long oblique line pointing to the left represents $\mathcal{R}$. Since $\mathcal{R}$ does not lie in $\mathcal{D}$ it is clear that $\mathcal{D} \oplus \mathcal{R} = \mathcal{C}_d^n$. The oblique projection of p along $\mathcal{R}$ onto $\mathcal{D}$ is given by the following expression

$$\sum_{i=1}^{s} h_i f_i \;=\; p/R^{\perp}\; [D/R^{\perp}]^{\dagger}\; D \tag{4.5}$$

where $p/R^{\perp}$ and $D/R^{\perp}$ are the orthogonal complements of p orthogonal on $\mathcal{R}$ and the rows of D orthogonal on $\mathcal{R}$ respectively [42]. A thorough overview of oblique projectors can be found in [37]. The dagger $\dagger$ stands for the Moore-Penrose pseudoinverse of a matrix. Note that expression (4.5) assumes that the basis for the vector spaces $\mathcal{D}$ and $\mathcal{R}$ are given by the rows of D and R.

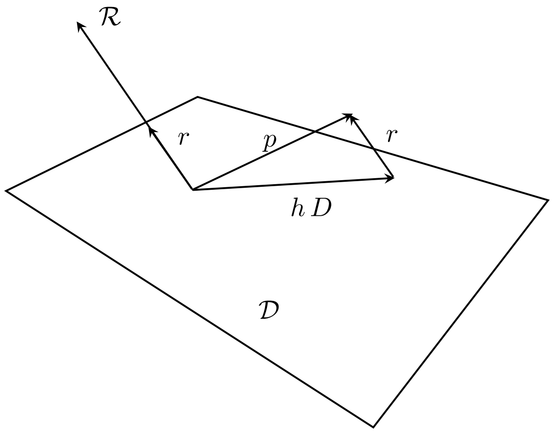

FIG. 4.1. ***The*** $\sum_{i=1} h_i f_i$ ***terms of the polynomial division of*** p ***by*** $F = \{f_1, \dots, f_s\}$ ***are found by projecting*** p ***along*** $\mathcal{R}$ ***onto*** $\mathcal{D}$***.***

4.5. Algorithm & Numerical Implementation. In this section a high-level algorithm and numerical implementation are presented for doing multivariate polynomial division. The outline of the algorithm is given in Algorithm 4.2. This is a high-level description since implementation details are ignored. The most important object in the algorithm is the divisor matrix D. From this matrix a basis for $\mathcal{D}$ and $\mathcal{R}$ are determined. The $\sum_i^s h_i f_i$ terms are then found from projecting p along $\mathcal{R}$ onto $\mathcal{D}$. The remainder is then found as $r = p - \sum_i^s h_i f_i$. The quotients h_i can easily be retrieved from solving the linear system $hD = \sum_i^s h_i f_i$.

```
ALGORITHM 4.2. Multivariate Polynomial Division
Input: polynomials $f_1, \dots, f_s, p \in \mathcal{C}_d^n$
Output: $h_1, \dots, h_s, r$ such that $p = \sum_i^s h_i f_i + r$
  $D \leftarrow$ Divisor matrix of $f_1, \dots, f_s$
  $A \leftarrow$ basis of vector space $\mathcal{D}$ determined from $D$
  $B \leftarrow$ monomial basis of vector space of remainders $\mathcal{R}$ determined from $D$
  $\sum_i^s h_i f_i \leftarrow$ project $p$ along $\mathcal{R}$ onto $\mathcal{D}$
  $r \leftarrow p - \sum_i^s h_i f_i$
  $h = \big(h_1, \dots, h_s\big) \leftarrow$ solve $hD = \sum_i^s h_i f_i$
```

We have implemented this algorithm in MATLAB and the code is available on request. The numerical implementation we propose uses 4 QR decompositions. The use of orthogonal matrix factorizations guarantees the numerical backward stability of the implementation. The third QR decomposition will dominate the cost of the method, which is $O((q+1)q^2)$ where q is the number of columns of D. Also note that q grows as $O(d^n)$ where $d = \deg(p)$ and n is the number of indeterminates. In addition, $M(d)$ also typically has a large amount of zero elements. An implementation using sparse matrix representations is therefore a logical choice. The implementation consists of three main steps: first, the rank of D, a basis for its row space and a basis for its kernel are computed. Second, the normal set is determined from the kernel and finally, the oblique projection is computed. Doing a full singular value decomposition in terms of a sparse matrix representation is too costly in terms of storage since the singular vectors will typically be dense. We therefore opt to use a sparse multifrontal multithreaded rank-revealing QR decomposition [14]. This sparse QR decomposition uses by default a numerical tolerance of $\tau = 20\ (q+s)\ \epsilon\ \max_j ||D_{*j}||_2$ where ϵ is the machine roundoff (about 10^{-16} since only a double-precision implementation of the sparse QR factorization is available), $\max_j ||D_{*j}||_2$ is the largest 2-norm of any row of D and D is s-by-q. The rank of D, a basis for its row space $\mathcal{D}$ and a basis for its kernel can all be derived from the following QR factorization

$$D^T\ P_d\ =\ Q_d\ R_d$$

where P_d corresponds with a column permutation which reduces fill-in and allows to determine the rank. The estimate for the rank r is given by the number of nonzero diagonal elements of R_d. The r leftmost columns of $D^T\ P_d$ span $\mathcal{D}$. An orthogonal basis for the kernel K is given by the remaining columns of Q_d. This first QR decomposition is a critical step in the implementation. Indeed, an ill-defined numerical rank indicates an inherent difficulty to determine the dimensions of $\mathcal{D}$ and $\mathcal{R}$. In practice however we have not yet seen this problem occur. Further work on how the approxi-rank gap [25] is influenced by perturbations on the coefficients of the divisors is required. Now, Corollary (4.6) is used to find the normal set. K is per definition of full column-rank, say $\dim(K) = c_r$, and from a second sparse QR decomposition

$$K^T\ P_k\ =\ Q_k\ R_k$$

the linearly independent rows of K are found as the leftmost c_r columns of $K^T\ P_k$. In fact, the factors Q_k and R_k do not need to be computed. The column permutation will work from the leftmost column of K^T to the right, which corresponds with checking the rows of K for linear independence from top to bottom. Corollary 4.6 then ensures a correct normal set for multivariate polynomial division is found. From this a canonical basis R for $\mathcal{R}$ can be constructed. With the first two steps completed one can now use (4.5) to find the projection of p onto $\mathcal{D}$ along $\mathcal{R}$. It is possible to simplify (4.5) in the following way. The orthogonal complement of p orthogonal on $\mathcal{R}$ is given by

$$p/R^{\perp}\ =\ p\,(I - R^T(R\,R^T)^{\dagger}\,R) \tag{4.6}$$

and likewise the orthogonal complement of $\mathcal{D}$ orthogonal on $\mathcal{R}$ by

$$D/R^{\perp}\ =\ D\,(I - R^T(R\,R^T)^{\dagger}\,R). \tag{4.7}$$

Implementing (4.5) involves calculating three matrix pseudo-inverses. We can reduce this to a normal matrix inverse by using another QR decomposition. In order to avoid confusion between the R of the QR decomposition and the basis of $\mathcal{R}$ an LQ decomposition is used with L lower triangular. In addition, as mentioned earlier, (4.5) requires bases for the vector spaces as rows of matrices. Using the LQ factorization therefore avoids the need to transpose all matrices. One can easily describe things in terms of a QR decomposition by taking the transpose of each of the matrices. Calculating the LQ factorization of

$$\begin{pmatrix} R \\ D \\ p \end{pmatrix}\ =\ L\,Q\ =\ \begin{pmatrix} L_R \\ L_D \\ L_p \end{pmatrix}\ Q \tag{4.8}$$

allows us to write

$$\begin{aligned} R &= L_R\, Q \\ D &= L_D\, Q \\ p &= L_p\, Q. \end{aligned}$$

Since R is a canonical basis all rows of R are orthonormal and will be contained in Q without any change. Hence, L_R will always be a unit matrix embedded into a rectangular structure

$$L_R \;=\; \begin{pmatrix} I_{c_r} & O \end{pmatrix}$$

where $c_r = \dim(\mathcal{R})$. This implies that $L_R\, L_R^T \;=\; I_{c_r}$. The next step is to replace p and R in (4.6) by their respective LQ decompositions

$$\begin{aligned} p/R^{\perp} &= p\,(I_q - R^T (R\, R^T)^{\dagger}\, R) \\ &= L_p\, Q\,(I_q - Q^T\, L_R^T\,(L_R\, Q\, Q^T\, L_R^T)^{\dagger}\, L_R\, Q) \\ &= L_p\, Q\,(I_q - Q^T\, L_R^T\,(L_R\, L_R^T)^{\dagger}\, L_R\, Q) \\ &= L_p\, Q\,(I_q - Q^T\, L_R^T\, L_R\, Q) \\ &= L_p\, Q\, Q^T\,(I_q - L_R^T\, L_R)\, Q \\ &= L_p\,(I_q - L_R^T\, L_R)\, Q. \end{aligned} \tag{4.9}$$

The simplifications in the different steps are possible since $L_R\, L_R^T \;=\; I_{c_r}$ and $Q\, Q^T \;=\; I_q$. The resulting expression is quite simplified and more importantly, no matrix pseudo-inverse is required anymore. Applying the same strategy of replacing $\mathcal{D}$ and R by their respective LQ decompositions in (4.7) results in

$$D/R^{\perp} \;=\; L_D\,(I_q - L_R^T\, L_R)\, Q. \tag{4.10}$$

From here on, W denotes the common factor $(I_q - L_R^T\, L_R)$. Using (4.9) and (4.10) in (4.5) we obtain

$$\begin{aligned} \sum_{i=1}^{s} h_i f_i &= p/R^{\perp}\ [D/R^{\perp}]^{\dagger}\ D \\ &= L_p\ W\ Q\ (L_D W Q)^{\dagger}\ D \\ &= L_p\ W\ Q\ Q^{\dagger}\ (L_D W)^{\dagger}\ D \\ &= L_p\ W\ (L_D W)^{\dagger}\ D \end{aligned} \tag{4.11}$$

which requires the calculation of only 1 matrix pseudo-inverse. Exploiting the structure of W allows to further simplify this expression. Since $W = (I_q - L_R^T\, L_R)$ and L_R is a unit matrix embedded in a rectangular structure it follows that

$$W \;=\; \begin{pmatrix} 0 & 0 \\ 0 & I_r \end{pmatrix}$$

where $r = q - c_r$ is the rank of D. Partitioning L_p into

$$L_p \;=\; \begin{pmatrix} L_{p_1} & L_{p_2} \end{pmatrix}$$

where L_{p_2} are the r rightmost columns and likewise L_D into

$$L_D \;=\; \begin{pmatrix} L_{D_1} & L_{D_2} \end{pmatrix}$$

simplifies (4.11) to $L_{p_2}\, L^{\dagger}_{D_2}$. We can therefore write the oblique projection of p along $\mathcal{R}$ on $\mathcal{D}$ as

$$\sum_{i=1}^{s} h_i f_i \;=\; L_{p_2}\, L^{\dagger}_{D_2}\, D. \tag{4.12}$$

Note that in this final expression the orthogonal matrix Q of (4.8) does not appear and it is therefore not necessary to calculate it explicitly. When L_{D_2} is of full column rank $L^{\dagger}_{D_2}$ can be obtained from a sparse Q-less QR decomposition. Writing

$$L_{D_2} \;=\; Q\, R_{\text{chol}}$$

reduces

$$L^{\dagger}_{D_2} \;=\; (L^T_{D_2}\, L_{D_2})^{-1}\, L^T_{D_2}$$

to solving the following matrix equation

$$R^T_{\text{chol}}\, R_{\text{chol}}\, L^{\dagger}_{D_2} \;=\; L^T_{D_2}$$

which can be solved by a forward substitution followed by a backward substitution. The factor $L_{p_2}\, L^{\dagger}_{D_2}$ in (4.12) specifies the linear combination of rows of D and therefore the decomposition of p into the $\sum_{i=1}^{s} h_i f_i$ terms. The remainder is then easily found as $r = p - \sum_{i=1}^{s} h_i f_i$.

4.6. Example. In this example we will replace x_1, x_2, x_3 with x, y, z respectively and divide the polynomial $p = -5 + 2x + y^2 + z^2 + 8xy^2$ by $F = \{f_1 = -4 + x^2 + y^2 + z^2, f_2 = -5 + x^2 + 2y^2, f_3 = -1 + xz\}$. The leading monomial of p according to the graded xel ordering is xy^2. The divisor matrix D is the following 5 by 20 matrix

$$D \;=\; \begin{pmatrix} f_1 \\ f_2 \\ xf_2 \\ f_3 \\ xf_3 \end{pmatrix}$$

The numerical tolerance for this example is $\tau = 5.468 \times 10^{-13}$. The rank is estimated to be 5 and therefore $\dim(\mathcal{R}) = 15$. The monomial basis for $\mathcal{R}$ is

$$\{1, x, y, z, x^2, xy, yz, x^3, x^2y, xyz, xz^2, y^3, y^2zyz^2, z^3\}.$$

The factor $L_{p2}\, L^{\dagger}_{D2}$ equals $\begin{pmatrix}1.0 & 0 & 4.0 & 0 & 0\end{pmatrix}$ and $\sum_i h_i f_i$ is therefore

$$\sum_i h_i f_i \;=\; 1.0\, f_1 + 4.0\, x\, f_2 \;=\; -4.0 - 20.0\, x + 1.0\, x^2 + 1.0\, y^2 + 1.0\, z^2 + 4.0\, x^3 + 8.0\, xy^2.$$

The remainder term r is easily found from the vector difference

$$r \;=\; p - \sum_i h_i f_i = -1.0\; + 22.0\, x - 1.0\, x^2 + 0.0\, y^2 + 0.0\, z^2 - 4.0\, x^3.$$

The total running time for computing both the quotients and the remainder was 0.011 seconds. The absolute errors for both the $\sum_i h_i f_i$ terms and r are bounded by above by 10^{-15}. Observe that, unlike in the univariate case, the leading monomial of the remainder is x^3 and has a larger degree than any of the divisors. We now perturb the coefficients of the divisors with noise of order 10^{-6} and divide p by $\{f_1 = -4.000001 + 0.000001\,y + x^2 + y^2 + z^2, f_2 = -5.000001 + x^2 + 2y^2, f_3 = -1 + 0.000001\,x^2 + xz\}$. Note that the noise introduced two extra terms: $10^{-6}\,y$ in f_1 and $10^{-6}\,x^2$ in f_3. The numerical rank of D remains 5 and the factor $L_{p2}\ L_{D2}^{\dagger}$ also does not change. The remainder term however now becomes

$$r \;=\; -1.0 + 22.000004\,x - 10^{-6}\,y - 1.0\,x^2 + 0.0\,y^2 + 0.0\,z^2 - 4.0x^3.$$

The noisy $10^{-6}\,y$ term ends up in the remainder and the coefficient of x is now also perturbed. Again, the absolute errors are bounded by above by 10^{-15}. The total running time was 0.013 seconds.

4.7. Gröbner Basis. In this section we discuss the difference between the results of the division algorithm described in this manuscript and the division of a multivariate polynomial by a Gröbner basis. We first start off with the definition of the Gröbner basis as in [13].

Definition 4.7. *Fix a monomial order. A finite subset $G = \{g_1, \ldots, g_t\}$ of a polynomial ideal $I = \langle f_1, \ldots, f_s\rangle$ is said to be a Gröbner basis if*

$$\langle LM(g_1), \ldots, LM(g_t)\rangle \;=\; \langle LM(I)\rangle$$

where $\langle LM(I)\rangle$ denotes the ideal generated by all leading monomials of I.

Or in other words, every leading monomial of an element of the ideal $I = \langle f_1, \ldots, f_s\rangle$ is divisible by at least one of the leading monomials of G. The Gröbner basis G of a given set of multivariate polynomials $f_1, \ldots, f_s$ hence generates the same polynomial ideal as $f_1, \ldots, f_s$. One attractive feature of a Gröbner basis is that the remainder will be independent on the ordering of the divisors. The normal set R which is found in Theorem 4.5 is however not necessarily the normal set R_G obtained when dividing by the corresponding Gröbner basis. This difference is due to the defining property of the Gröbner basis. Not all leading terms of I are necessarily divisible by at least one of the polynomials $f_1, \ldots, f_s$ and this implies that $R_G \subseteq R$.

Example 4.3. *We revisit the unperturbed example of Section 4.6 and first compute the Gröbner basis of $I = \langle f_1, f_2, f_3\rangle$ using Maple. This is*

$$G = \{g_1 = -1 + xz, g_2 = -5 + x^2 + 2y^2, g_3 = -3 + x^2 + 2z^2, g_4 = 2\,z - 3\,x + x^3\}.$$

Note that G contains four polynomials whereas F only three. Applying Algorithm 4.1 for G results in the following normal set $R_G = \{1, x, y, z, x^2, xy, yz, x^2y\}$, which is indeed a subset of R. Since the difference between R and R_G lies in the higher degrees, the remainder r_G from dividing p by G contains fewer terms of higher degree,

$$r_G = -1.0 + 10.0\,x + 8.0\,z - 1.0\,x^2 + 0.0\,x^3 + 0.0\,xy^2.$$

For this example the x^3 term of r does not appear in r_G. Computation of this remainder r_G took 0.012 *seconds in MATLAB.*

The orthogonal basis for $\mathcal{D}$ from the QR decomposition in Algorithm 4.2 does not correspond with a Gröbner basis since it will not satisfy Definition 4.7.

5. Multivariate Polynomial Elimination. Gaussian elimination is probably the most known form of elimination. It involves the manipulation of linear equations such that the solution set does not change and one of the resulting equations is univariate. The same idea is generalized by a Gröbner basis using a lexicographic monomial ordering. The problem of multivariate elimination can be stated as follows: Given a system of multivariate polynomials $f_1, \ldots, f_s$ and a proper subset of variables $x_e \subsetneq \{x_i : i = 1, \ldots, n\}$, find a polynomial $g = \sum_i^s h_i f_i$ ($h_1, \ldots, h_s$ being multivariate polynomials) in which all variables x_e are eliminated. The key in solving this problem will be a matrix which is very similar to the divisor matrix in that its row space describes 'linear combinations' of the form $\sum_{i=1}^s h_i f_i$ with $h_i, f_i \in \mathcal{C}_d^n$ for a certain degree d. The difference lies in the fact that the requirement $\mathrm{LM}(p) \geq \mathrm{LM}(h_i f_i)$ is dropped since there is no dividend p in this context. The resulting matrix is called the Macaulay matrix and is defined as follows.

Definition 5.1. *Given a set of polynomials $f_1, \ldots, f_s \in \mathcal{C}_d^n$, each of degree d_i ($i = 1, \ldots, s$) then the Macaulay matrix of degree d is the matrix containing the coefficients of*

$$M(d) = \begin{pmatrix} f_1 \\ x_1 f_1 \\ \vdots \\ x_n^{d-d_1} f_1 \\ f_2 \\ x_1 f_2 \\ \vdots \\ x_n^{d-d_s} f_s \end{pmatrix} \tag{5.1}$$

where each polynomial f_i is multiplied with all monomials from degree 0 up to $d - d_i$ for all $i = 1, \ldots, s$.

The reason (5.1) is called the Macaulay matrix is because it was Macaulay who introduced this matrix, drawing from earlier work by Sylvester [40], in his work on elimination theory, resultants and solving multivariate polynomial systems [26, 27]. This matrix depends explicitly on the degree d for which it is defined, hence the notation $M(d)$. The row space of $M(d)$ will be denoted by $\mathcal{M}_d$ and is the vector space of all linear combinations of the form $\sum_{i=1}^s h_i f_i$ with $\deg(h_i f_i) \leq d$. The polynomial g lies therefore obviously in $\mathcal{M}_d$ for some degree d. In addition, g also lies in the vector space spanned by all monomials which do not contain any element of x_e up to the same degree d.

Definition 5.2. *Given a proper subset of variables $x_e \subsetneq \{x_i : i = 1, \ldots, n\}$ then the elimination vector space $\mathcal{E}_d$ is the vector space of polynomials with maximal degree d spanned by all monomials that do not contain any of the variables of x_e.*

A canonical basis $E(d)$ for the vector space $\mathcal{E}_d$ is easily obtained. The following example illustrates.

Example 5.1. *Suppose $x_e = \{x_2, x_4\} \in \mathcal{C}_2^4$. A canonical basis for $\mathcal{E}_2$ is then* *given by*

$$\begin{array}{ccccccccccccccc}
1 & x_1 & x_2 & x_3 & x_4 & x_1^2 & x_1x_2 & x_1x_3 & x_1x_4 & x_2^2 & x_2x_3 & x_2x_4 & x_3^2 & x_3x_4 & x_4^2 \\
\end{array}$$
$$\begin{pmatrix}
1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0
\end{pmatrix}.$$

The fact that this canonical basis is orthogonal will prove useful in the implementation of the elimination algorithm later. Note that the $k \times q$ matrix $E(d)$ can always be written as

$$E(d) \;=\; \begin{pmatrix} I_k & 0 \end{pmatrix} P \tag{5.2}$$

where P is a column permutation and I_k is the unit matrix of dimension k. The intersection of $\mathcal{M}_d$ and $\mathcal{E}_d$ brings us naturally to the geometry of multivariate elimination.

5.1. The Geometry of Polynomial Elimination. The polynomial g which is to be found lies both in $\mathcal{M}_d$ and $\mathcal{E}_d$. Polynomial elimination therefore corresponds with finding an intersection of these two vector spaces. This is depicted in Figure 5.1. Here, both the row space of $M(d)$ and $E(d)$ are represented as 2-dimensional planes. The polynomial g corresponds with the vector lying in the 1-dimensional intersection. The condition that g lies in this intersection can be written as

$$g \;=\; x\,M(d) \;=\; y\,E(d) \tag{5.3}$$

where both x and y are unknown row vectors. The degree d for which (5.3) applies is not known a priori and will need to be found. So the problem of multivariate elimination can be summarized as finding the degree d and either x or y. Using (5.2), (5.3) can be written as

$$\begin{aligned}
x\,M(d) &= y\,E(d) \\
x\,M(d) &= y\,\begin{pmatrix} I_k & O \end{pmatrix} P \\
x\,M(d)P^T &= y\,\begin{pmatrix} I_k & O \end{pmatrix}.
\end{aligned}$$

Partitioning $M(d)\,P^T$ column-wise into

$$\begin{pmatrix} M_k(d) & M_{q-k}(d) \end{pmatrix}$$

such that $M_k(d)$ corresponds with the k leftmost columns of $M(d)\,P^T$ allows us to further write

$$x\begin{pmatrix} M_k(d)_k & M_{q-k}(d) \end{pmatrix} = \begin{pmatrix} yI_k & O \end{pmatrix}$$

or

$$\begin{cases} x\,M_k(d) &= \;\; yI_k \\ x\,M_{q-k}(d) &= \;\; O. \end{cases}$$

One can therefore find x as an element of the left null space of $M_{q-k}(d)$ which contains all $q-k$ columns of $M(d)$ corresponding with monomials that are eliminated. Instead of solving (5.3), we will go deeper into the geometry of elimination and explore the link with principal angles of vector spaces.

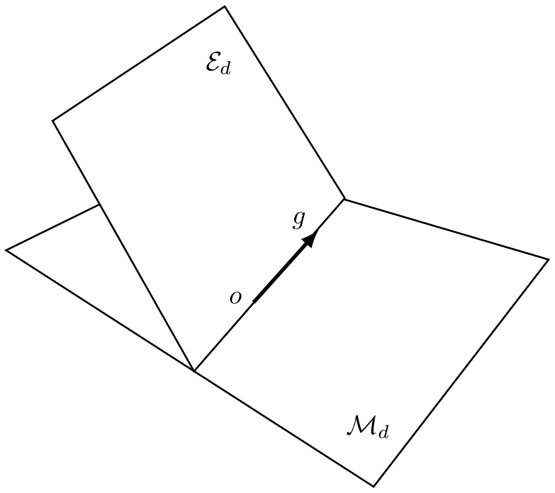

FIG. 5.1. ***The polynomial** $g = \sum_i^s h_i f_i$ **with all variables** x_e **eliminated lies in the intersection of** $\mathcal{M}_d$ **and** $\mathcal{E}_d$**.***

5.2. Principal Angles & CS Decomposition. In what follows we will always assume, without loss of generality, that the polynomials have real coefficients. The framework of principal angles, first proposed by Jordan in 1875 [19, 22], lends itself well to describe the intersection of vector spaces. They are defined as follows.

DEFINITION 5.3. *The principal angles* $0 \leq \theta_1 \leq \theta_2 \leq \ldots \theta_{min(d_1,d_2)} \leq \pi/2$ *between the vector spaces* S_1 *and* S_2 *of dimension* d_1 *and* d_2 *respectively, and the corresponding principal directions* $u_i \in S_1$ *and* $v_i \in S_2$ *are defined recursively as*

$$cos(\theta_k) \quad = \quad \max_{u_k \in S_1, v_k \in S_2} u_k^T\, v_k \textit{ for } k = 1, \ldots, min(d_1, d_2)$$

subject to

$$||u_k|| = ||v_k|| = 1,$$

and for $k > 1$

$$\begin{aligned} u_k^T\, u_i &= 0, \quad i = 1, \ldots, k-1, \\ v_k^T\, v_i &= 0, \quad i = 1, \ldots, k-1. \end{aligned}$$

Inspecting the principal angles between $\mathcal{M}_d$ and $\mathcal{E}_d$ will therefore reveal whether an intersection exists. Since the principal angles form an ordered set one simply needs to check whether $\cos(\theta_1) = 1$. The corresponding principal vectors u_1 or v_1 are then the sought-after polynomial g. Several numerical algorithms have been proposed for the computation of the principal angles and the corresponding principal directions [5]. In this article the SVD-based approach will be used.

THEOREM 5.4. *Assume that the columns of* Q_a *and* Q_b *form orthogonal bases for two subspaces of* $\mathcal{C}_d^n$*. Let*

$$A \;=\; Q_a^T\, Q_b,$$

and let the SVD of this $p \times q$ *matrix be*

$$A \;=\; Y\, C\, Z^T, \quad C \;=\; diag(\sigma_1, \ldots, \sigma_q),$$

where $Y^TY = Z^TZ = I_q$. If we assume that $\sigma_1 \geq \sigma_2 \geq \ldots \geq \sigma_q$, then the principal angles and principal vectors associated with this pair of subspaces are given by

$$cos(\theta_k) = \sigma_k(A), \quad U = Q_aY, \quad V = Q_bZ.$$

Proof. See [5, p. 582]. □

Theorem 5.4 provides a nice insight into the link between multivariate polynomial elimination and the CS decomposition (CSD). A thorough overview of the CSD and its relation to the generalized singular value decomposition (GSVD) is given in [3]. More information on the computational aspects of the CSD can be found in [38, 39]. It was Stewart [35] who first put forward an explicit CSD form. In this article we will present a particular thin version of the CSD presented in [3, 36].

THEOREM 5.5. *Let $Q \in \mathbb{R}^{q\times r}$ have orthonormal columns. Partition Q in the form*

$$Q = \begin{array}{cc} & r \\ k & Q_1 \\ q-k & Q_2 \end{array}.$$

Then there are orthogonal matrices $U_1 \in \mathbb{R}^{k\times k}, U_2 \in \mathbb{R}^{(q-k)\times(q-k)}$, and $V \in \mathbb{R}^{r\times r}$ such that

$$\begin{pmatrix} U_1^T & 0 \\ 0 & U_2^T \end{pmatrix} \begin{pmatrix} Q_1 \\ Q_2 \end{pmatrix} V = \begin{pmatrix} U_1^TQ_1V \\ U_1^TQ_2V \end{pmatrix} = \begin{pmatrix} \Sigma_1 \\ \Sigma_2 \end{pmatrix}$$

assumes the following form if $k < r$ and $q - k \geq r$:

$$\begin{array}{cc} & r \\ k & \Sigma_1 \\ q-k & \Sigma_2 \end{array} = \begin{array}{c|cc} & r-k & k \\ k & C & 0 \\ k & S & 0 \\ r-k & 0 & I \\ q-r-k & 0 & 0 \end{array}.$$

Here C and S are nonnegative diagonal matrices satisfying

$$C^2 + S^2 = I.$$

Proof. See [3, 36]. □

The reason this particular form of the CSD is chosen is because for the case of elimination $k < r$ and $q - k > r$. Suppose now that the columns of $Q(d)$ form an orthogonal basis for $\mathcal{M}_d$. Then, from (5.2) it is clear that the SVD of $E(d)\,Q(d)$ as in Theorem 5.4 corresponds with

$$\begin{pmatrix} I_k & 0 \end{pmatrix} P\,Q(d) = YCZ^T.$$

P will permute the rows of $Q(d)$ such that the top k rows correspond with the monomials which are not eliminated. We can therefore further write the SVD of $E(d)\,Q(d)$ as

$$\begin{pmatrix} I_k & 0 \end{pmatrix} \begin{pmatrix} Q_k(d) \\ Q_{q-k}(d) \end{pmatrix} = YCZ^T$$
$$Q_k(d) = YCZ^T$$

which is the upper part of the CSD decomposition, $U_1^T Q_1 V$, in Theorem 5.5. In this way the natural connection between the CS-decomposition and multivariate polynomial elimination is revealed. In the next section we will discuss an algorithm and an implementation to perform multivariate elimination.

5.3. Algorithm & Numerical Implementation. In this section a high-level algorithm for multivariate polynomial elimination is presented along with a numerical implementation. As mentioned before, Algorithm 5.1 is based on checking the intersection of the vector spaces $\mathcal{M}_d$ and $\mathcal{E}_d$ rather than solving (5.3). Since $\deg(g)$ is unknown, the algorithm iterates over the degree d with an initial value given by the maximal degree of the given polynomials $f_1, \ldots, f_s$. In each iteration the matrices $E(d)$ and $M(d)$ are constructed. As soon as there is an intersection between $\mathcal{M}_d$ and $\mathcal{E}_d$, an element of this intersection is returned as g.

```
ALGORITHM 5.1. Multivariate Elimination
Input: polynomials $f_1, \ldots, f_s \in \mathcal{C}_d^n$, monomial set $x_e$
Output: $g \in \mathcal{M}_d \cap \mathcal{E}_d$
  $d \leftarrow max(deg(f_1), deg(f_2), \ldots, deg(f_s))$
  $g \leftarrow [\,]$
  while $g = [\,]$ do
    $E(d) \leftarrow$ canonical basis for $\mathcal{E}_d$
    $M(d) \leftarrow$ Macaulay matrix of degree d
    if $\mathcal{M}_d \cap \mathcal{E}_d \neq \emptyset$ then
      $g \leftarrow$ element from intersection
    else
      $d \leftarrow d + 1$
    end if
  end while
```

We have implemented this algorithm also in MATLAB and its code is freely available on request. The same argument on the growth of the dimensions of the matrix applies for $M(d)$ and a sparse matrix representation is used for $E(d)$ and $M(d)$. In fact, $E(d)$ can be completely stored as a vector containing the indices of the monomials that span $\mathcal{E}_d$. In order to find the cosines of the principle angles an orthogonal basis for both vector spaces is required. An orthogonal basis for $\mathcal{M}_d$ can be found from the sparse rank-revealing QR decomposition of $M(d)^T$,

$$M(d)^T \, P \;=\; Q(d)\, R.$$

The rank r of $M(d)$ is estimated from the diagonal of R and the orthogonal basis for $\mathcal{M}_d$ are the r leftmost columns of $Q(d)$. The same tolerance τ is used for the rank determination as for polynomial division. Again, the rank-estimation here is an important step. Like for the polynomial division algorithm, computing this QR decomposition dominates the computational complexity, which is $O(qs^2)$. Backward numerical stability is guaranteed from the orthogonal matrix factorization. Then using Theorem 5.4 we need to inspect the singular values of $E(d)\,Q(d) = Q_k(d)$. In fact, since the principal angles form an ordered set it is sufficient to inspect the first singular value which can be determined, together with the first left and right singular vectors y and z, from an implicitly restarted Arnoldi iteration [24]. Since we only compute the largest singular value and corresponding singular vectors no problems of numerical instability occur (associated with the loss of orthogonality of subsequent singular vectors). As soon as this singular value lies, with a certain tolerance, close enough to one there is an intersection. Numerical experiments seem to indicate that the cosines of the principle angles are quite robust with respect to noise on the coefficients of $f_1, \ldots, f_s$. We therefore opt to choose the same numerical tolerance τ as for the rank-estimation. It is however possible that some loss of numerical accuracy occurs when determining the principal angle from its cosine. This can be avoided, as described in [5, p. 582-583] and [23, p. 6], by computing the sine of the principal angle. As soon as there is an non-empty intersection, the first principal vector is then the sought-after polynomial g which is retrieved as

$$g \;=\; E(d)^T\, y.$$

At each iteration additional rows are added to $M(d)$. A possible optimization of the implementation with respect to the computational complexity would therefore involve updating the QR factorization instead of recomputing it completely.

5.4. Example. From the following polynomial system in $\mathcal{C}^6_4$

$$\begin{cases} x_1^2 + x_3^2 - 1 &=& 0\\ x_2^2 + x_4^2 - 1 &=& 0\\ x_5x_3^3 + x_6x_4^3 - 1.2 &=& 0\\ x_5x_1^3 + x_6x_2^3 - 1.2 &=& 0\\ x_5x_3^2x_1 + x_6x_4^2x_2 - 0.7 &=& 0\\ x_5x_3x_1^2 + x_6x_4x_2^2 - 0.7 &=& 0 \end{cases}$$

we eliminate $x_e = \{x_1, x_2, x_3, x_4, x_5\}$ using Algorithm 5.1. For all $d < 10$ we have that $|C(1,1) - 1| > \tau$. For $d = 10, \tau = 1.46 \times 10^{-10}$ and $|C(1,1) - 1| = 4.44 \times 10^{-16} < \tau$. The Macaulay matrix $M(10)$ is a 9702×8008 matrix with 29106 nonzero elements which corresponds with a density of 3.7%. This justifies the use of a sparse matrix representation. The first principal vector is

$$g = 0.9011 - 0.0\,x_6 - 0.4335\,x_6^2 + 0.0\,x_6^3 + 0.0\,x_6^4 - 0.0\,x_6^5 - 0.0\,x_6^6 + 0.0\,x_6^7 - 0.0\,x_6^8.$$

The total running time took 31.87 seconds. All 0.0 coefficients are bounded from above by τ and can therefore be considered to be numerically zero. This implies that the roots of this particular polynomial system have only 2 distinct x_6 components, 1.4417 and -1.4417. From the Gröbner basis method the exact solutions for x_6 can be found: $\pm\frac{19}{330}\sqrt{627}$. These allows us to determine the absolute error of our numerical solutions: 7.40×10^{-11}. When eliminating $x_e = \{x_1, x_2, x_3, x_4\}$ the first singular value is exact 1 and $\tau = 4.69 \times 10^{-11}$ for $d = 8$. The principal vector is then

$$h = -0.0305 + 0.7141\,x_5^2 - 0.6994\,x_6^2 - 0.0004\,x_5^4 + 0.0009\,x_5^2x_6^2 - 0.0004\,x_6^4$$

which took 8.46 seconds to find. Note that h also contains 22 other nonzero coefficients, each of which is bounded from above by 10^{-19}. We have omitted these 'zeros' for the sake of presentation. We now add perturbations of 10^{-6} to the original polynomial system to obtain

$$\begin{cases} 1.000001\,x_1^2 + x_3^2 - 1 &=& 0\\ x_2^2 + x_4^2 - 1 &=& 0\\ x_5x_3^3 + x_6x_4^3 - 1.200001 &=& 0\\ x_5x_1^3 + x_6x_2^3 - 1.2 &=& 0\\ x_5x_3^2x_1 + 1.000001\,x_6x_4^2x_2 - 0.7 &=& 0\\ 1.000001\,x_5x_3x_1^2 + x_6x_4x_2^2 - 0.700001 &=& 0. \end{cases}$$

When eliminating $x_e = \{x_1, x_2, x_3, x_4, x_5\}$ we have again for $d = 10$ that $|C(1,1)-1| = 6.34 \times 10^{-14} < \tau = 1.46 \times 10^{-10}$. Since $d = 10$ the Macaulay matrix will have the same size and structure as the original polynomial system. The univariate polynomial in x_6 is now given by

$$\hat{g} = 0.9011 - 0.0\,x_6 - 0.4335\,x_6^2 + 0.0\,x_6^3 + 0.0\,x_6^4 - 0.0\,x_6^5 - 0.0\,x_6^6 + 0.0\,x_6^7 + 0.0\,x_6^8$$

for which $||g - \hat{g}||_2 < 10^{-7}$. This reflects the loss of precision due to the added noise. The running time was 30.08 seconds.

The example shows one strategy to solve multivariate polynomial systems. One could triangularize the polynomial system by using elimination and perform 'back substitution' through the repetitive use of univariate polynomial root-finders. A major concern about this strategy is that since the solutions of the univariate polynomial will only be approximate, the remaining polynomials to be solved will also be approximate and hence a polynomial system which is different from the original is being solved. There is little guarantee that the solutions of this new system are close to the original ones. Accumulated errors can build up rapidly when the number of variables are high and when polynomials of high degree are present. As one of the reviewers pointed out, there exist univariate rootfinders that approximate roots up to an arbitrary accuracy [4]. These could be possibly used to alleviate the problem of accumulated errors for the case of exact coefficients. An alternative strategy for multivariate polynomial root-finding would be to solve a generalized eigenvalue problem [33, 34]. How this can be done without the use of a Gröbner basis will be explained in further work.

5.5. Gröbner basis. In computational algebraic geometry, the tool for elimination is a Gröbner basis with respect to the lexicographic monomial ordering. This Gröbner basis has the triangular structure which was mentioned in the previous section. In addition to the problem of accumulating errors when doing the back substitution, the downside of this Gröbner basis method is that a Gröbner basis for lexicographic orderings suffers from a large size and is typically expensive to compute.

EXAMPLE 5.2. *We revisit the unperturbed example of 5.4 and compute the Gröbner basis of the polynomial system with respect to the lexicographic ordering* $(x_1 > x_2 > x_3 > x_4 > x_5 > x_6)$ *using Maple(TM) [1]:*

$$\left\{\begin{array}{rcl} g_1 : -6859 + 3300\,x_6^2 & = & 0 \\ g_2 : -6859 + 3300\,x_5^2 & = & 0 \\ g_3 : 133 - 330\,x_4\,x_6 + 361\,x_4^2 & = & 0 \\ g_4 : -6270\,x_5 + 3300\,x_4x_5x_6 + 6859\,x_3 & = & 0 \\ g_5 : 361\,x_4 - 330\,x_6 + 361\,x_2 & = & 0 \\ g_6 : -3300\,x_4x_5x_6 + 6859\,x_1 & = & 0 \end{array}\right.$$

This took 0.3 *seconds. After scaling the univariate polynomial* g *of our algorithm such that its constant term equals -6859 we can compare it with the exact result of the Gröbner basis. We then have that* $||g - g_1||_2 = 5.48 \times 10^{-11}$. *Also note that* g_2 *of the Gröbner basis, which eliminates* x_1, x_2, x_3, x_4 *from the original polynomial system, is univariate in* x_5 *and of degree two. Our computed result is bivariate in* x_5, x_6 *and of degree four. This shows that the result of elimination is not unique. Indeed, the set of all polynomials* $g = \sum_i h_i f_i$ *from which the monomials* x_e *are eliminated forms a polynomial ideal. These ideals are called elimination ideals. It is a classic result that a Gröbner basis with respect to the lexicographic ordering generates these elimination* *ideals. The computed bivariate polynomial in x_5, x_6 hence lies in the elimination ideal spanned by g_1, g_2.*

For the perturbed polynomial system it is impossible to print the Gröbner basis $\{\hat{g_1}, \hat{g_2}, \hat{g_3}, \hat{g_4}, \hat{g_5}, \hat{g_6}\}$ due to its large size. The reason for this large size is the appearance of higher order terms and of coefficients which consist of a large number of digits. The constant term of $\hat{g_1}$ for example has 169 digits. After proper scaling we can compute $||g_1 - \hat{g_1}||_2 = 0.68$ which indicates a much higher loss of precision.

6. Conclusion.

This article introduced a linear algebra framework in which three fundamental operations of multivariate polynomials were discussed: multiplication, division and elimination. It was shown how multiplication corresponds with a vector matrix product and division with a vector decomposition. Both the quotients and the remainder can be found from an oblique projection onto certain subspaces. Elimination was revealed to be linked with principal angles between subspaces and the CS decomposition.

The description of multivariate polynomials in a linear algebra setting suffers on the one hand from an inherent combinatorial explosion of dimensions. On the other hand, the matrices are extremely sparse and therefore a sparse matrix representation can be employed. In addition, the matrices described in this article are also very structured. Further exploitation of this sparseness and structure in the implementation of the algorithms are the topics of future research. One already mentioned optimization would be to update the QR factorization of the Macaulay matrix instead of recomputing it completely during the elimination algorithm. Another interesting and necessary avenue of future work is to achieve a better understanding of the numerical properties of the Divisor and Macaulay matrix. Especially the influence of perturbations on the rank revealing problem is of major importance since it might affect the choice of a suitable numerical tolerance.

Acknowledgements. The authors would like to thank M. Agudelo and D. Geebelen for useful discussions and comments. We would also like to thank the two anonymous referees for their many helpful comments.

REFERENCES

[1] Maple 16, *Maplesoft, a division of Waterloo Maple Inc*, 2012. Waterloo, Ontario.

[2] W Auzinger and H J Stetter, *An Elimination Algorithm for the Computation of All Zeros of a System of Multivariate Polynomial Equations*, in Int. Conf. on Numerical Mathematics, Singapore 1988, Birkhäuser ISNM 86, 1988, pp. 11–30.

[3] Z. Bai, *The CSD, GSVD, their Applications and Computations*, IMA preprint series 958,Institute for Mathematics and its Applications, University of Minnesota, MN, (1992).

[4] D. A. Bini and G. Fiorentino, *Design, Analysis, and Implementation of a Multiprecision Polynomial Rootfinder*, Numerical Algorithms, 23 (2000), pp. 127–173.

[5] Å. Björck and G. H. Golub, *Numerical Methods for Computing Angles Between Linear Subspaces*, Mathematics of Computation, 27 (1973), pp. pp. 579–594.

[6] P. Boito, *Structured Matrix Based Methods for Approximate Polynomial GCD*, 2011.

[7] N. K. Bose, *Applied Multidimensional Systems Theory*, Van Nostrand Reinhold, 1982.

[8] B. Buchberger, *Ein Algorithmus zum Auffinden der Basiselemente des Restklassenringes nach einem nulldimensionalen Polynomideal*, PhD thesis, Mathematical Institute, University of Innsbruck, Austria, 1965.

[9] B. Buchberger, *Gröbner Bases: A Short Introduction for Systems Theorists*, 2001, pp. 1–19.

[10] ———, *Gröbner Bases and Systems Theory*, Multidimensional Systems and Signal Processing, 12 (2001), pp. 223–251.

[11] R. M. Corless, P. M. Gianni, B. M. Trager, and S. M. Watt, *The Singular Value Decomposition for Polynomial Systems*, in ACM International Symposium on Symbolic and Algebraic Computation, 1995, pp. 195–207.

[12] D. A. Cox, J. B. Little, and D. O'Shea, *Using Algebraic Geometry*, Graduate Texts in Mathematics, Springer-Verlag, Berlin-Heidelberg-New York, March 2005.
[13] ———, *Ideals, Varieties and Algorithms*, Springer-Verlag, third ed., 2007.
[14] T. A. Davis, *Algorithm 915, SuiteSparseQR: Multifrontal Multithreaded Rank-Revealing Sparse QR Factorization*, ACM Transactions on Mathematical Software, 38 (2011).
[15] M. Drton, B. Sturmfels, and S. Sullivant, *Lectures on Algebraic Statistics*, vol. 39 of Oberwolfach Seminars, Springer, 2009.
[16] N. K. Bose (Ed.), *Multidimensional Systems Theory: Progress, Directions and Open Problems*, Reidel, 1985.
[17] Ioannis Z. Emiris, André Galligo, and Henri Lombardi, *Certified Approximate Univariate GCDs*, Journal of Pure and Applied Algebra, 117 118 (1997), pp. 229 – 251.
[18] J-C Faugère, *A new efficient algorithm for computing Gröbner bases (f4)*, Journal of Pure and Applied Algebra, 139 (1999), pp. 61–88.
[19] G. H. Golub and C. F. Van Loan, *Matrix Computations*, The Johns Hopkins University Press, 3rd ed., Oct. 1996.
[20] S. C. Johnson, *Sparse Polynomial Arithmetic*, SIGSAM Bull., 8 (1974), pp. 63–71.
[21] G. Jónsson and S. A. Vavasis, *Accurate Solution of Polynomial Equations Using Macaulay Resultant Matrices*, 2001.
[22] C. Jordan, *Essai sur la géométrie à n dimensions*, Bulletin de la Société Mathématique, 3 (1875), pp. 103–174.
[23] AV Knyazev and ME Argentati, *Principal Angles between Subspaces in an A-based Scalar Product: Algorithms and Perturbation Estimates*, SIAM Journal on Scientific Computing, 23 (2002), pp. 2008–2040.
[24] R. B. Lehoucq and D. C. Sorensen, *Deflation Techniques for an Implicitly Restarted Arnoldi Iteration*, SIAM Journal on Matrix Analysis and Applications, 17 (1996), pp. 789–821.
[25] TY Li and Z. Zeng, *A Rank-Revealing Method with Updating, Downdating, and Applications*, SIAM Journal on Matrix Analysis and Applications, 26 (2005), pp. 918–946.
[26] F. S. Macaulay, *On some Formulae in Elimination*, Proc. London Math. Soc., 35 (1902), pp. 3–27.
[27] ———, *The algebraic theory of modular systems*, Cambridge University Press, 1916.
[28] D. Manocha and J. F. Canny, *MultiPolynomial Resultant Algorithms*, J. Symbolic Computation, 15 (1993), pp. 99–122.
[29] M. Monagan and R. Pearce, *Sparse Polynomial Division Using a Heap*, Journal of Symbolic Computation, (2010).
[30] B. Mourrain and V. Y. Pan, *Multivariate Polynomials, Duality, and Structured Matrices*, Journal of Complexity, 16 (2000), pp. 110 – 180.
[31] L. Pachter and B. Sturmfels, eds., *Algebraic Statistics for Computational Biology*, Cambridge University Press, August 2005.
[32] MATLAB R2012a, *The Mathworks Inc.*, 2012. Natick, Massachusetts.
[33] H.J. Stetter, *Matrix Eigenproblems are at the Heart of Polynomial System Solving*, SIGSAM Bulletin, 30 (1996), pp. 22–5.
[34] ———, *Numerical Polynomial Algebra*, Society for Industrial and Applied Mathematics, Philadelphia, PA, USA, 2004.
[35] G.W. Stewart, *On the Perturbation of Pseudo-inverses, Projections and Linear Least Squares Problems*, SIAM Review, 19 (1977), pp. 634–662.
[36] G. W. Stewart, *Computing the CS decomposition of a Partitioned Orthonormal Matrix*, Numerische Mathematik, 40 (1982), pp. 297–306.
[37] G. W. Stewart, *On the Numerical Analysis of Oblique Projectors*, SIAM Journal on Matrix Analysis and Applications, 32 (2011), pp. 309–348.
[38] B. Sutton, *Computing the complete CS decomposition*, Numerical Algorithms, 50 (2009), pp. 33–65. 10.1007/s11075-008-9215-6.
[39] B. D. Sutton, *Stable Computation of the CS Decomposition: Simultaneous Bidiagonalization*, SIAM Journal on Matrix Analysis and Applications, 33 (2012), pp. 1–21.
[40] J. J. Sylvester, *On a Theory of the Syzygetic Relations of Two Rational Integral Functions, Comprising an Application to the Theory of Sturm's Functions, and That of the Greatest Algebraical Common Measure*, Trans. Roy. Soc. Lond., (1853).
[41] B. L. van der Waerden, *Modern Algebra, vol 2*, Frederick Ungar, New York, 1950.
[42] P. Van Overschee and B. De Moor, *Subspace Identification for Linear Systems: Theory, Implementation, Applications*, Kluwer Academic Publishers, 1996.
[43] Z. Zeng, *A numerical elimination method for polynomial computations*, Theor. Comput. Sci., 409 (2008), pp. 318–331.
[44] Z. Zeng and B.H. Dayton, *The approximate GCD of inexact polynomials*, in Proceedings

of the 2004 international symposium on Symbolic and algebraic computation, ISSAC '04, New York, NY, USA, 2004, ACM, pp. 320–327.