
Usage Guidelines:
Please refer to usage guidelines at contact lib-eprints@bbk.ac.uk. or alternatively
A Geometric View of Cryptographic Equation Solving

S. Murphy and M.B. Paterson

Information Security Group
Royal Holloway
University of London
Egham, Surrey TW20 0EX, U.K.

Abstract. This paper considers the geometric properties of the Relin-
cearisation algorithm and of the XL algorithm used in cryptology for equa-
tion solving. We give a formal description of each algorithm in terms of
projective geometry, making particular use of the Veronese variety. We
establish the fundamental geometrical connection between the two algo-
rithms and show how both algorithms can be viewed as being equivalent
to the problem of finding a matrix of low rank in the linear span of a col-
lection of matrices, a problem sometimes known as the MinRank prob-
lem. Furthermore, we generalise the XL algorithm to a geometrically
variant algorithm, which we term the GeometricXL algorithm. The
GeometricXL algorithm is a technique which can solve certain equation
systems that are not easily soluble by the XL algorithm or by Groebner
basis methods.

keywords. Projective Geometry, Veronese Variety, Determinantal Va-
rity, Multivariate Polynomials, Cryptology, Linearisation, Relinearisa-
tion, XL Algorithm, GeometricXL Algorithm.

AMS Classification. 68W30, 14N05, 94A20.

1 Introduction

The solution of a multivariate polynomial equation system is a classical prob-
lem in algebraic geometry and computer algebra [11, 12]. There has also been
much recent interest in cryptology in techniques for solving multivariate equa-
tion systems over finite fields. Various classical methods, such as Buchberger’s
algorithm [3] and other related algorithms for computing a Gröbner basis [14, 15,
23], have been considered in a cryptographic context. Furthermore, the obvious
method to attempt to solve such equation systems is the Linearisation algo-
rithm [21], which has been considered in cryptology. In the Linearisation algo-
rithm, the equation system is regarded as a linear system. This naive Linearisation
algorithm has been adapted to give other methods, such as the Relinearisation
algorithm [21] and the XL (extended linearisation) algorithm [10], which have
been proposed as being particularly appropriate in cryptology. The geometric

* M.B. Paterson was supported by EPSRC research grant GR/S42637.
aspects of the Relineralisation algorithm and the XL algorithm are the main concern of this paper.

The comments and methods of this paper about solution methods for multivariate equation systems always apply in a field of characteristic zero. However, we are concerned with solution methods for the multivariate equation systems that arise in cryptology, so in this paper we consider such systems over a finite field \( \mathbb{F} \). We sometimes require that the positive characteristic \( p \) of the finite field \( \mathbb{F} \) is not too small, and we make this statement more precise in Section 2.2. We usually consider multivariate polynomial systems \( f_1 = \ldots = f_m = 0 \) consisting of \( m \) homogeneous polynomials \( f_1, \ldots, f_m \in \mathbb{F}[x_0, x_1, \ldots, x_n] \) of the same degree \( d \). This condition is not at all restrictive as any polynomial \( f \) of degree \( d \) in \( n \) variables can be transformed into a homogeneous polynomial in \( n + 1 \) variables by the homogenising transformation

\[
    f(x_1, \ldots, x_n) \mapsto x_0^d f\left(\frac{x_1}{x_0}, \ldots, \frac{x_n}{x_0}\right).
\]

For simplicity, our discussion is based on multivariate quadratic systems (\( d = 2 \)), though our comments are usually more generally applicable.

The general geometrical structures that are required to analyse properties of the Relineralisation and XL algorithms are discussed in Section 2. In our geometric analysis, we make particular use of a structure known as the Veronese Variety, which we discuss in Section 3. The Relineralisation algorithm is based on the Linearisation algorithm, and we consider the geometric properties of the Linearisation algorithm in Section 4, before discussing the geometric properties of the Relineralisation algorithm in Section 5. The related XL algorithm is then discussed in Section 6, which leads to the definition of a new geometrically invariant version of the XL algorithm, the GeometricXL algorithm, in Section 7. The paper finishes with some general comments and observations in Section 8.

2 Vector Spaces and Projective Geometry

In this section, we give a brief description of the general algebraic and geometric structures that we use in our analysis of the Relineralisation algorithm and the XL algorithm.

2.1 The Symmetric Power of a Vector Space

In this paper, we make extensive use of the symmetric power of a vector space, which we now define. This is most naturally done in the language of the tensor product of vector spaces [7]. For simplicity, we give an approach that uses vector space bases, but it is just as possible to give an abstract explanation of a tensor product.

Suppose that \( \{e_0, e_1, \ldots, e_{n-1}, e_n\} \) is the basis for the \((n + 1)\)-dimensional vector space \( V \) over \( \mathbb{F} \). We can define a set of \((n + 1)^2\) formal symbols \( \{e_i \otimes e_j\} \) \((0 \leq i, j \leq n)\). For our purposes, we regard the tensor product \( V \otimes V \) as an
A vector in $V \otimes V$ has $(n+1)^2$ components and so is naturally represented by a square $(n+1) \times (n+1)$ array or matrix, with the $(i, j)$ component of the vector in $V \otimes V$ being the $(i, j)$-entry of the matrix. Thus the tensor product space $V \otimes V$ can be thought of as the vector space of $(n+1) \times (n+1)$ matrices, with a basis vector $e_i \otimes e_j$ being the matrix with 1 in position $(i, j)$ and 0 everywhere else. In this matrix formulation, the inclusion mapping $\iota$ from $V \times V$ to $V \otimes V$ is given by $(v_1, v_2) \mapsto v_1 v_2^T$ for column vectors $v_1, v_2 \in V$.

One subspace of the tensor product vector space that is of particular interest is the subspace of symmetric tensors. The definition of a symmetric tensor in $V \otimes V$ is clear. If $t = (t_{ij})$ is a tensor in $V \otimes V$, then $t$ is a symmetric tensor if $t_{ij} = t_{ji}$ for all $i$ and $j$. In the matrix formulation of $V \otimes V$, $t$ is a symmetric matrix, so the set of all symmetric tensors is the subspace of symmetric matrices. Thus the set of all symmetric tensors forms a subspace of $V \otimes V$, which is called the symmetric square or second symmetric power of $V$ [17]. The symmetric square has dimension $\frac{1}{2}(n+1)(n+2)$, and we denote the symmetric square by $S^2(V)$. In the matrix formulation of $V \otimes V$, a matrix is in the symmetric square of $V$ if and only if it is a symmetric matrix, so the symmetric square $S^2(V)$ can be thought of as the vector space of symmetric matrices.

We can of course generalise the above construction to the $d$-fold tensor product $V \otimes \ldots \otimes V$. A tensor $t = (t_{i_1 \ldots i_d})$ is a symmetric tensor if

$$t_{i_1 \ldots i_d} = t_{\sigma(i_1) \ldots \sigma(i_d)}$$

for all $i_1, \ldots, i_d$, where $\sigma$ is any permutation of $d$ objects. The set of all symmetric tensors forms a subspace of $V \otimes \ldots \otimes V$, called the $d$th symmetric power of the vector space $V$, and we denote it by $S^d(V)$. The dimension of vector space $S^d(V)$ is $(\frac{n+d}{d})$ [8], the number of monomials of degree $d$ in $n + 1$ variables [17].

### 2.2 The Symmetric Power of a Dual of a Vector Space

The dual space $V^*$ of a finite-dimensional vector space $V$ over $\mathbb{F}$ of dimension $n + 1$ is defined to be the vector space of all linear functionals on $V$, that is any mapping $\sigma_a : V \to \mathbb{F}$, where $a \in V$, of the form $x \mapsto a^T x$ for all $x \in V$. Thus the dual space $V^*$ also has dimension $n + 1$ and can be thought of as the vector space of all homogeneous linear polynomials $a_0 x_0 + \ldots + a_n x_n$ in $(n + 1)$ variables (with the 0-polynomial).

As $V^*$ is a vector space, we can also define its $d$th symmetric power $S^d(V^*)$. It can similarly be seen that this $d$th symmetric power of the dual space, $S^d(V^*)$, can be thought of as the vector space of all homogeneous polynomials of degree $d$ in $(n + 1)$ variables (with the 0-polynomial).

In this paper, we are sometimes specifically concerned with the case that $d < p$, where $d$ is the degree of the homogeneous system and $p$ the positive
characteristic of \( \mathbb{F} \). In this case, we can take formal partial derivatives of a homogeneous polynomial of degree \( d \). If we let \( D_{x_i} \) denote taking such a formal partial derivative with respect to \( x_i \), so \( D_{x_i}f = \frac{\partial f}{\partial x_i} \), then

\[ D_{x_i} : \mathbb{S}^d(V^*) \rightarrow \mathbb{S}^{d-1}(V^*) \]

that is taking a derivative maps a homogeneous degree \( d \) polynomial to a homogeneous degree \( d-1 \) polynomial. More generally, if \( x = x_1 \ldots x_k \) is a monomial of degree \( k \) (\( k \leq d < p \)) and \( D^k_{x_i} \) denotes taking the \( k \)-th order partial derivative with respect to the monomial \( x_i \), then

\[ D^k_{x_i} : \mathbb{S}^d(V^*) \rightarrow \mathbb{S}^{d-k}(V^*) \]

Moreover, \( D^k_{x_i} \) is a linear transformation between these vector spaces.

We can also use such \( k \)-th order partial derivative mapping \( D^k_{x_i} \) to define subspaces of \( \mathbb{S}^{d-k}(V^*) \). For a homogeneous polynomial \( f \) of degree \( d \), so \( f \in \mathbb{S}^d(V^*) \), we define

\[ W_f^{(k)} = \langle D^k_{x_i}f \mid x_i \text{ is a monomial of degree } k \rangle \]

a subspace of \( \mathbb{S}^{d-k}(V^*) \). We can represent all the possible \( k \)-th order partial derivatives of \( f \) as a matrix in which each row is a vector \( D^k_{x_i}f \in \mathbb{S}^{d-k}(V^*) \). We call such a matrix a partial derivatives matrix and denote it by \( C_f^{(k)} \). By construction, the row space of this partial derivatives matrix \( C_f^{(k)} \) is the subspace \( W_f^{(k)} < \mathbb{S}^{d-k}(V^*) \) and its rank is the dimension of \( W_f^{(k)} \).

**Example 1.** Consider the polynomial \( f \in \text{GF}(37)[x_0,x_1,x_2] \) given by

\[
8x_0^3+34x_0^2x_1+20x_0^2x_2+26x_0x_1^2+8x_0x_1x_2+28x_0x_2^2+32x_1^3+3x_1^2x_2+34x_1x_2^2+25x_2^3.
\]

The first and second partial derivatives matrices of \( f \) are respectively given by

\[
C_f^{(1)} = \begin{pmatrix}
24 & 31 & 3 & 26 & 8 & 28 \\
34 & 15 & 8 & 22 & 6 & 34 \\
20 & 8 & 19 & 3 & 31 & 1
\end{pmatrix}
\quad \text{and} \quad
C_f^{(2)} = \begin{pmatrix}
11 & 31 & 3 \\
31 & 15 & 8 \\
8 & 3 & 19 \\
15 & 7 & 6 \\
6 & 31 & 1 \\
19 & 31 & 2
\end{pmatrix}.
\]

In order to use partial derivatives in this way, we generally assume that \( d < p \) in this paper when considering partial derivatives. In particular, this means that this paper is not directly concerned with the case when the finite field \( \mathbb{F} \) has characteristic 2 when discussing partial derivatives. The proper technical approach for considering formal partial derivatives in nonzero characteristic is to use a divided power ring and a contraction action in place of the multivariate polynomial ring \( \mathbb{F}[x_0, \ldots, x_n] \) and the formal derivative [19]. However, these two approaches are equivalent in the case when \( d < p \), that is the degree of the equation system is less than the positive field characteristic. In this case, the “partial derivatives” matrix is equivalent to the catalecticant matrix [19] in the divided power ring.
2.3 Projective Geometry

As in Section 2.1, we consider the vector space $V$ of dimension $n + 1$ over the finite field $\mathbb{F}$. Any invertible linear transformation $V \to V$ gives a well-defined mapping of the set of one-dimensional subspaces to itself, which is essentially just a change of co-ordinates and is known as a \textit{collineation}. The \textit{projective geometry} $\mathbb{P}(V)$ is the geometry obtained by considering the one-dimensional subspaces of $V$ under the group of all collineations, so

$$
\mathbb{P}(V) = \{ \langle (x_0, x_1, \ldots, x_n)^T \rangle \mid (x_0, x_1, \ldots, x_n)^T \in V \setminus \{0\} \}.
$$

This projective geometry $\mathbb{P}(V)$ is said to be of (projective) dimension $n$ and is generically denoted by $\text{PG}(n, \mathbb{F})$ where there is no danger of confusion. The vector subspaces of $V$ define the projective subspaces of $\mathbb{P}(V)$.

We now define some terms from projective geometry that we use in this paper. A (projective) \textit{line}, \textit{plane}, \textit{secundum} and \textit{hyperplane} are projective subspaces of (projective) dimension 1, 2, $(n - 2)$ and $(n - 1)$ respectively of $\text{PG}(V)$. The (projective) \textit{variety} $V(f_1, \ldots, f_m)$ of a set of homogeneous polynomials $\{f_1, \ldots, f_m\}$ in $(n + 1)$ variables over $\mathbb{F}$ is the subset of $\text{PG}(V)$ for which $f_1 = \ldots = f_m = 0$. A \textit{primal} of degree $d$ is a variety of a single homogeneous polynomial of degree $d$, and a \textit{quadric} is a primal of degree 2, that is a quadric is a variety defined by a single homogeneous quadratic polynomial.

The \textit{tangent space} to a variety is defined in the following way. Suppose that $P$ is a point of a primal $V(f)$ given by equation $f = 0$ for some homogeneous polynomial $f$ with the property that the formal partial derivatives $\left( \frac{\partial f}{\partial x_i} \right)_{|P}$ are not all zero. The tangent space to $V(f)$ at $P$ is denoted by $T_P(V(f))$ and is the hyperplane defined by the equation

$$
\left( \frac{\partial f}{\partial x_0} \right)_{|P} x_0 + \left( \frac{\partial f}{\partial x_1} \right)_{|P} x_1 + \ldots + \left( \frac{\partial f}{\partial x_n} \right)_{|P} x_n = 0.
$$

Suppose now that $f_1, \ldots, f_m$ are homogeneous polynomials of the same degree and that $P$ is a point of a variety $V(f_1, \ldots, f_m) = \bigcap_{i=1}^{m} V(f_i)$. Provided that each tangent space in the intersection is well-defined, the tangent space to the variety $V(f_1, \ldots, f_m)$ at $P$ is defined as

$$
T_P(V(f_1, \ldots, f_m)) = \bigcap_{i=1}^{m} T_P(V(f_i)).
$$

A \textit{chord} or \textit{secant} of a variety is a line joining a pair of points of that variety, and the \textit{chordal variety} or \textit{secant variety} of a variety is the variety containing all chords or secants to that variety. The \textit{pencil} generated by two primal varieties $V(f_1)$ and $V(f_2)$ of the same degree is the set of varieties

$$
\{ V(\lambda_1 f_1 + \lambda_2 f_2) \mid \lambda_1, \lambda_2 \in \mathbb{F} \text{ not both } 0 \}.
$$

The aspects of projective geometry relevant to this paper are discussed in [5, 18, 28].
The projective geometries of main interest in this paper are those formed by the $d^{th}$ symmetric powers of the vector space $V$ and its dual $V^*$, namely

$$\mathbb{P}(\mathbb{S}^d(V))$$ and $$\mathbb{P}(\mathbb{S}^d(V^*))$$,

which have (projective) dimension $N_d = (n+d) - 1$ (Section 2.1 and [17]). In particular, we denote the (projective) dimension of both $\mathbb{P}(\mathbb{S}^2(V))$ and $\mathbb{P}(\mathbb{S}^2(V^*))$ by $N$, where $N = N_2 = \frac{1}{2}(n + 1)(n + 2) - 1 = \frac{1}{2}n(n + 3)$. Furthermore, points in either of these projective geometries $\mathbb{P}(\mathbb{S}^2(V))$ or $\mathbb{P}(\mathbb{S}^2(V^*))$ can be thought of as nonzero $(n + 1) \times (n + 1)$ symmetric matrices and their scalar multiples (Section 2.1).

3 Veronese Varieties

Our geometric analysis of the Relinearisation algorithm and the XL algorithm makes extensive use of the geometrical structure known as the Veronese variety. In its most general form, the Veronese variety is a structure of $\mathbb{P}(\mathbb{S}_d(V))$, the projective geometry of the $d^{th}$ symmetric power of a vector space, though the case of the symmetric square $\mathbb{P}(\mathbb{S}_2(V))$ is of most interest to us.

3.1 The Veronese Surface

We first illustrate the Veronese variety by considering the Veronese variety generated by the projective geometry $\mathbb{P}(V)$, where $V$ is a vector space of dimension 3 (so $n = 2$) over $\mathbb{F}$. This projective geometry

$$\mathbb{P}(V) = \{ (x_0, x_1, x_2)^T \mid (x_0, x_1, x_2)^T \in V \setminus \{0\} \}$$

is also known as the projective plane $\mathbb{P}(2, \mathbb{F})$. This Veronese variety is a subset of $\mathbb{P}(\mathbb{S}^2(V))$, a projective geometry of dimension $N = \frac{1}{2}(2 \cdot 5) = 5$,

$$\mathbb{P}(\mathbb{S}^2(V)) = \{ (y_{00}, y_{01}, y_{02}, y_{11}, y_{12}, y_{22})^T \mid (y_{00}, \ldots, y_{22})^T \in \mathbb{S}^2(V) \setminus \{0\} \}.$$  

The Veronese embedding is the mapping $\varphi_V : \mathbb{P}(V) \rightarrow \mathbb{P}(\mathbb{S}^2(V))$ defined by

$$(x_0, x_1, x_2)^T \mapsto \left( x_0^2, x_0x_1, x_0x_2, x_1^2, x_1x_2, x_2^2 \right)^T.$$  

The Veronese variety $\mathcal{V}_V$ is the image of the projective plane $\mathbb{P}(V)$ under this mapping, so

$$\mathcal{V}_V = \varphi_V(\mathbb{P}(V)) \subset \mathbb{P}(\mathbb{S}^2(V)).$$

In this particular case of the projective plane, the Veronese variety $\mathcal{V}_V$ is known as the Veronese surface. The Veronese embedding $\varphi_V$ is a bijection, so $\mathcal{V}_V$ contains $q^2 + q + 1$ points. Thus the Veronese surface $\mathcal{V}_V$ is known as a variety of dimension 2 as it is in one-to-one correspondence with a 2-dimensional projective space. Furthermore, the Veronese surface $\mathcal{V}_V$ has order 4, as it intersects a generic $(5 - 2) = 3$-dimensional subspace in 4 points.
We also give another useful method of defining the Veronese surface. In Section 2.1, we saw that the points of projective space \( \mathbb{P}(\mathbb{S}^2(V)) \) can be identified with the elements of the vector space of \( 3 \times 3 \) symmetric matrices, that is matrices of the form
\[
\begin{pmatrix}
 y_{00} & y_{01} & y_{02} \\
y_{01} & y_{11} & y_{12} \\
y_{02} & y_{12} & y_{22}
\end{pmatrix}.
\]

In this matrix formulation, the Veronese embedding \( \varphi_V : \mathbb{P}(V) \to \mathbb{P}(\mathbb{S}^2(V)) \) is given by
\[
\begin{pmatrix}
x_0 \\
x_1 \\
x_2
\end{pmatrix} \mapsto \begin{pmatrix}
x_0 \\
x_1 \\
x_2
\end{pmatrix} \begin{pmatrix}
x_0x_1 & x_0x_2 & x_1x_2 \\
x_0x_1 & x_1^2 & x_1x_2 \\
x_0x_2 & x_1x_2 & x_2^2
\end{pmatrix}.
\]

It is clear to see that a point \( P \in \mathbb{P}(\mathbb{S}^2(V)) \) is in \( \mathcal{V}_V = \text{Im}(\varphi_V) \) if and only if the matrix corresponding to \( P \) has rank 1, that is if and only if all the 2-minors (\( 2 \times 2 \) sub-determinants) vanish. Thus the Veronese surface \( \mathcal{V}_V \) in \( \mathbb{P}(\mathbb{S}^2(V)) \) can be defined as the set of all points \( P = ((y_{00}, y_{01}, y_{02}, y_{11}, y_{12}, y_{22})^T) \) such that all six 2-minors of the above matrix are zero, namely
\[
0 = y_{00}y_{11} - y_{01}^2, \quad 0 = y_{00}y_{22} - y_{02}^2, \quad 0 = y_{11}y_{22} - y_{12}^2, \\
0 = y_{00}y_{12} - y_{01}y_{02}, \quad 0 = y_{02}y_{11} - y_{01}y_{12} \quad \text{and} \quad 0 = y_{01}y_{22} - y_{02}y_{12}.
\]

### 3.2 Veronese Varieties of Degree 2

We can define Veronese varieties of higher dimension by a similar process. The projective geometry of a vector space \( V \) of dimension \( n + 1 \) is defined as
\[
\mathbb{P}(V) = \{ (x_0, x_1, \ldots, x_n)^T \mid (x_0, x_1, \ldots, x_n)^T \in V \setminus \{0\} \},
\]

a projective geometry of dimension \( n \). The corresponding projective geometry of the symmetric square of \( V \), \( \mathbb{S}^2(V) \), is defined by
\[
\mathbb{P}(\mathbb{S}^2(V)) = \{ (y_{00}, y_{01}, \ldots, y_{ij}, \ldots, y_{nn})^T \mid y_{ij} \in \mathbb{F}, i \geq j \}.
\]

This is a projective geometry of dimension \( N = \frac{1}{2}n(n + 3) \) (Section 2.3). The Veronese embedding
\[
\varphi_V : \mathbb{P}(V) \to \mathbb{P}(\mathbb{S}^2(V))
\]
of the first projective space in the second is defined by
\[
(x_0, x_1, \ldots, x_n)^T \mapsto (x_0^2, x_0x_1, \ldots, x_0x_n, x_1^2, \ldots, x_1x_n, \ldots, x_n^2)^T.
\]

The Veronese variety \( \mathcal{V}_V \) of dimension \( n \) is the image of \( \mathbb{P}(V) \) under \( \varphi_V \), so
\[
\mathcal{V}_V = \varphi_V(\mathbb{P}(V)) \subset \mathbb{P}(\mathbb{S}^2(V)).
\]

The intersection of the Veronese variety \( \mathcal{V}_V \) with a generic \( (N - n) \)-dimensional subspace has \( 2^n \) points, so the Veronese variety is said to have order \( 2^n \).
The vector space $S^2(V)$ can also be thought of as the vector space of symmetric $(n + 1) \times (n + 1)$ matrices of dimension $(N + 1)$ (Section 2.1), that is matrices of the form
\[
\begin{pmatrix}
y_{00} & y_{01} & y_{02} & \cdots & y_{0n} \\
y_{01} & y_{11} & y_{12} & \cdots & y_{1n} \\
y_{02} & y_{12} & y_{22} & \cdots & y_{2n} \\
\vdots & \vdots & \vdots & \ddots & \vdots \\
y_{0n} & y_{1n} & y_{2n} & \cdots & y_{nn}
\end{pmatrix}.
\]

We can also similarly define $P (S^2(V))$ in terms of such symmetric $(n + 1) \times (n + 1)$ matrices. In this matrix formulation, the Veronese embedding $\varphi_V : \mathbb{P}(V) \to \mathbb{P}(S^2(V))$ is defined by
\[
\begin{pmatrix}
x_0 \\
x_1 \\
\vdots \\
x_n
\end{pmatrix} \mapsto \begin{pmatrix}
x_0 x_1 \\
x_0 x_1 x_2 \\
\vdots \\
x_0 x_1 x_2 \cdots x_n
\end{pmatrix} = \begin{pmatrix}
x_0^2 \\
x_0 x_1 \\
\vdots \\
x_0 x_1 x_2 \cdots x_n
\end{pmatrix}.
\]

As before, it is clear to see that a point $P \in V_V$ if and only if the matrix corresponding to $P$ has rank 1. An $(n + 1) \times (n + 1)$ symmetric matrix has $\frac{1}{2}n(n + 1)^2$ independent 2-minors [16], which must all vanish if the matrix has rank 1. However, each such 2-minor defines a quadric in $P (S^2(V))$, and a point $P \in P (S^2(V))$ is in the Veronese variety $V_V$ if and only if $P$ lies in the intersection of all the these quadrics. Thus the Veronese variety $V_V \subset P (S^2(V))$ can be defined as the intersection of $\frac{1}{2}n(n + 1)^2(n + 2)$ quadrics in $P (S^2(V))$.

Further information about Veronese varieties can be found in [2, 18, 27, 28]. A Veronese variety is an example of a determinantal variety [17, 19].

### 3.3 Higher Degree Veronese Varieties

The Veronese embedding $\varphi_V : \mathbb{P}(V) \to \mathbb{P}(S^2(V))$ can be generalised to degrees higher than 2. The higher degree Veronese embedding
\[
\varphi_V^{(d)} : \mathbb{P}(V) \to \mathbb{P}(S^d(V))
\]
is an embedding of $\mathbb{P}(V)$ in a projective space of (projective) dimension $N_d = \binom{n+d}{d} - 1$ and is defined by
\[
(x_0, x_1, \ldots, x_n)^T \mapsto (x_0^d, x_0^{d-1} x_1, \ldots, x_{n-1} x_n^{d-1}, x_n^d)^T.
\]

The higher degree Veronese variety $V_V^{(d)}$ of dimension $n$ is the image of $\mathbb{P}(V)$ under $\varphi_V^{(d)}$, so we have
\[
V_V^{(d)} = \varphi_V^{(d)} (\mathbb{P}(V)) \subset \mathbb{P}(S^d(V))
\].
3.4 Veronese Varieties of the Dual Space

We now consider the projective geometry $\mathbb{P}(S^d(V^*))$ of the symmetric power of the dual vector space $V^*$ (Section 2.3). In particular, if we consider the elements of $\mathbb{P}(V^*)$ and $\mathbb{P}(S^2(V^*))$ as (up to scalar multiplication) homogeneous linear and quadratic polynomials respectively, then the ordinary Veronese embedding

$$\varphi_{V^*} : \mathbb{P}(V^*) \rightarrow \mathbb{P}(S^2(V^*)),$$

is defined by the mapping

$$\langle a_0x_0 + \ldots + a_nx_n \rangle \mapsto \langle (a_0x_0 + \ldots + a_nx_n)^2 \rangle,$$

when the positive characteristic of $F$ is more than 2 ($p > 2$) [17, 19]. In this case, the corresponding Veronese variety $V_{V^*} = \varphi_{V^*}(\mathbb{P}(V^*))$ can be characterised as all homogeneous quadratic polynomials which are squares (up to scalar multiplication), that is

$$V_{V^*} = \{ \langle L^2 \rangle \mid L \text{ is a linear polynomial} \} \subset \mathbb{P}(S^2(V^*)) .$$

More generally, the higher degree Veronese variety of degree $d$ has a similar characterisation for $d < p$ [17, 19]. The higher degree Veronese variety $V_{V^*}^{(d)} = \varphi_{V^*}^{(d)}(\mathbb{P}(V^*))$ of $\mathbb{P}(S^d(V^*))$ is given by

$$V_{V^*}^{(d)} = \{ \langle L^d \rangle \mid L \text{ is a linear polynomial} \} \subset \mathbb{P}(S^d(V^*)) .$$

Thus the Veronese varieties arising from dual spaces in the case that $d < p$ are sets consisting of any polynomial which is the appropriate power of some linear polynomial.

4 A Geometric View of the Linearisation Algorithm

The Linearisation algorithm [21] is a very well-known and long-standing general technique to solve a multivariate equation system, in which the basic idea is to regard every monomial as an independent variable. This turns the original system of equations into a linear system of equations in the new variables. The new linear system is known as the linearised system and can be easily solved with basic linear algebra, and any solution of the original system is also a solution of the new linearised system. However, in situations where the rank of the new linearised system is significantly less than the number of monomials in the original system, the new linearised system can produce far too many possible incorrect solutions to the original system.

From a geometrical perspective, the Linearisation algorithm is fundamentally a technique in which a projective space is embedded in another projective space of higher dimension, with the intention that a nonlinear variety in the first space becomes a linear variety in the second larger space. This linear variety can
then be easily analysed using simple linear algebra, thus allowing us to reach conclusions about the original variety in the smaller space. In particular, if the original linear variety is the unique solution of a system of quadratic equations, then it may be possible with the **Linearisation** algorithm to solve this system using only linear algebra.

The **Relinearisation** algorithm and the **XL** algorithm are developments of the basic **Linearisation** algorithm, and both algorithms use the **Linearisation** algorithm. Thus any geometric analysis of the **Relinearisation** algorithm and the **XL** algorithm requires a thorough geometric understanding of the **Linearisation** algorithm.

### 4.1 Linearisation of a Quadric

The Veronese embedding $\varphi_V : \mathbb{P}(V) \to \mathbb{P}(S^2(V))$ induces a linearisation mapping $\varphi_V$ from the set of homogeneous quadratic polynomials in $F[x_0, x_1, \ldots, x_n]$ to the set of homogenous linear polynomials in $F[y_{00}, y_{ij}, \ldots, y_{nn}]$ defined by

$$\sum_{i=0}^{n} \sum_{j=0}^{i} a_{ij}x_i x_j \mapsto \sum_{i=0}^{n} \sum_{j=0}^{i} a_{ij}y_{ij}.$$  

We then say that $\bar{f} = \varphi_V(f)$ is the linearisation of the homogeneous quadratic polynomial $f = \sum_{i=0}^{n} \sum_{j=0}^{i} a_{ij}x_i x_j$. For such a quadratic polynomial $f$, the geometric structure defined by $Q_f = \{ \langle x_0, x_1, \ldots, x_n \rangle^T \mid f(x_0, x_1, \ldots, x_n) = 0 \} \subset \mathbb{P}(V)$ is a quadric (Section 2.3). Geometrically, the linearisation mapping $\varphi_V$ induces a mapping from the quadrics in $\mathbb{P}(V)$ to the hyperplanes of $\mathbb{P}(S^2(V))$, which we also denote by $\varphi_V$. Thus $\varphi_V$ is also a mapping in which the quadric $Q_f$ in $\mathbb{P}(V)$ is mapped to the hyperplane $H_f$ in $\mathbb{P}(S^2(V))$, so $H_f = \varphi_V(Q_f)$, where

$$H_f = \{ \langle y_{00}, \ldots, y_{ij}, \ldots, y_{nn} \rangle^T \mid \bar{f}(y_{00}, \ldots, y_{ij}, \ldots, y_{nn}) = 0 \} \subset \mathbb{P}(S^2(V)).$$

### 4.2 Linearisation of a Quadratic Equation System

Suppose $f \in F[x_0, x_1, \ldots, x_n]$ is a homogeneous quadratic equation with the (projective) point $P \in \mathbb{P}(V)$ as a solution of $f = 0$, so $P \in Q_f$. By construction, the point $\varphi_V(P) \in \mathbb{P}(S^2(V))$ is a solution of $\bar{f} = \varphi_V(f) = 0$, or equivalently $\varphi_V(P) \in H_f$. Suppose now that $P \in \mathbb{P}(V)$ is a solution of a system of $m$ such independent homogeneous quadratic equations $f_1 = \ldots = f_m = 0$, then $\varphi_V(P) \in H_{f_1}, \ldots, H_{f_m}$. We can define the projective subspace $H \subset \mathbb{P}(S^2(V))$ by

$$H = \bigcap_{i=1}^{m} H_{f_i} \subset \mathbb{P}(S^2(V)),$$

so we clearly have

$$\varphi_V(P) \in H \subset \mathbb{P}(S^2(V)).$$
Thus the solutions in $P(V)$ of a system of homogeneous quadratic polynomials are mapped to points in the intersection of hyperplanes in $P(S^2(V))$. The intersection of hyperplanes can be efficiently calculated by row reduction of a matrix, so a linear space containing $\varphi_V(P)$ can be easily obtained. If the original equation system has a unique solution (so $m > n$) and this space $H$ is a unique (projective) point, then necessarily $H$ is on the Veronese variety $\mathcal{V}_V$. We can then obtain the unique (projective) solution $P$ to the original equation system as

$$P = \varphi_V^{-1}(H).$$

This geometric technique for equation solving is a geometric description of the Linearisation algorithm. However, the Linearisation algorithm can give “parasitic” solutions, which are elements of $H$ which do not correspond to solutions of the original equation system. In fact, if we define the linearisation variety $L$ by

$$L = \mathcal{V}_V \bigcap H \subset P(S^2(V)),$$

then the solution set of the original equation system is given by

$$\varphi_V^{-1}(L) = \varphi_V^{-1}(\mathcal{V}_V \bigcap H) \subset P(V),$$

so the solution set is given by the intersection of the Veronese variety with the intersection of hyperplanes. Parasitic solutions can arise when this hyperplane intersection is not contained in the Veronese variety. However, the Veronese variety contains no non-trivial linear spaces, so the hyperplane intersection $H$ is only contained in the Veronese variety $\mathcal{V}_V$ if it is a single point. The solutions of the quadratic system $f_1 = \cdots = f_m = 0$ are therefore given by the system of linear equations $\bar{f}_1 = \cdots = \bar{f}_m = 0$ and the quadratic equations that define the Veronese variety $\mathcal{V}_V$. When the original equation system has a unique solution given by the point $P \in P(V)$, then the Linearisation algorithm succeeds when $\varphi_V(P) \in \mathcal{L} = H$, that is the Veronese quadratic equations are not needed to obtain a unique solution.

**Example 2.** Consider the following quadratic equation system

$$0 = 1 + x_1 + x_2 - x_1 x_2$$
$$0 = 2 + x_2 + x_1^2 - x_2^2$$
$$0 = x_1 + x_2 - 2x_1^2 + 2x_1 x_2 - x_2^2$$
$$0 = 3 + x_1 + 9x_2 + 8x_1^2 + 18x_1 x_2 + 22x_2^2$$
$$0 = 1 + 4x_1 + 3x_2 + 2x_1^2 - 3x_1 x_2 - 5x_2^2$$

with five equations in two variables over GF(37). Homogenising these equations by the addition of a variable $x_0$ gives

$$0 = f_1 = x_0^2 + x_0 x_1 + x_0 x_2 - x_1 x_2$$
$$0 = f_2 = 2x_0^2 + x_0 x_2 + x_1^2 - x_2^2$$
$$0 = f_3 = x_0 x_1 + x_0 x_2 - 2x_1^2 + 2x_1 x_2 - x_2^2$$
$$0 = f_4 = 3x_0^2 + x_0 x_1 + 9x_0 x_2 + 8x_1^2 + 18x_1 x_2 + 22x_2^2$$
$$0 = f_5 = x_0^2 + 4x_0 x_1 + 3x_0 x_2 + 2x_1^2 - 3x_1 x_2 - 5x_2^2.$$
We thus take $V$ to be the vector space of dimension 3 over GF(37), so $n = 2$ and $N = \frac{1}{2}(2 \cdot 5) = 5$. The above equation system now defines a variety in $\mathbb{P}(V)$. The Veronese embedding $\varphi_V : \mathbb{P}(V) \to \mathbb{P}(S^2(V))$ induces a linearisation mapping $\varphi_V$, which we can use to obtain the equation system

\[
\begin{align*}
0 &= f_1 = y_{00} + y_{01} + y_{02} - y_{12} \\
0 &= f_2 = 2y_{00} + y_{02} + y_{11} - y_{22} \\
0 &= f_3 = y_{01} + y_{02} - 2y_{11} + 2y_{12} - y_{22} \\
0 &= f_4 = 3y_{00} + y_{01} + 9x_{02} + 8y_{11} + 18y_{12} + 22y_{22} \\
0 &= f_5 = y_{00} + 4y_{01} + 3y_{02} + 2y_{11} - 3y_{12} - 5y_{22}.
\end{align*}
\]

Each of these linear equations defines a hyperplane $H_{f_i}$, so we have

\[
H = \bigcap_{i=1}^{5} H_{f_i} = \langle (1, 2, 3, 4, 6, 9)^T \rangle \subset \mathbb{P}(S^2(V)).
\]

Applying the inverse Veronese embedding gives

\[
\varphi_V^{-1}(H) = \langle (1, 2, 3)^T \rangle \subset \mathbb{P}(V).
\]

Thus we have $(x_0, x_1, x_2) = \lambda(1, 2, 3)$, which is the only solution as $H$ contains a single (projective) point. To obtain the solution to the original nonhomogeneous equation system, we set $x_0 = 1$, that is we take $\lambda = 1$ to obtain $(x_1, x_2) = (2, 3)$.

□

In general, a system of $m$ homogeneous quadratic equations in $\mathbb{P}(V)$ leads to $m$ hyperplanes in $\mathbb{P}(S^2(V))$. These hyperplanes intersect in a space of dimension $N - m$. Thus linearisation transforms the original problem in $n$ dimensions into a problem in $\frac{1}{2}n(n + 3) - m$ dimensions.

5. A Geometric View of the Relinearisation Algorithm

The Relinearisation algorithm [21] is a technique that can sometimes be used when the Linearisation algorithm fails, that is the generated solution contains parasitic solutions. The technique of linearisation gives a subspace of a projective space that contains all solutions. The Relinearisation algorithm applies a further linearisation mapping to this subspace with the aim of recovering this solution.

5.1 Relinearisation of a Linearisation Variety

When the Linearisation algorithm fails, we know that the Veronese embedding $\varphi_V(P)$ of a solution $P \in \mathbb{P}(V)$ of the original homogeneous equation system lies in the linearisation variety $\mathcal{L} = V \cap H$. However, the linearisation variety is the intersection of quadrics, so we have

\[
\mathcal{L} = \bigcap_{i=1}^{s} Q_{f_i}.
\]
where \( i = 1, \ldots, s \) with \( s \leq \frac{1}{12} n(n+1)^2(n+2) \) and \( \hat{f}_i \) is a homogeneous quadratic polynomial in \( \mathbb{F}[y_0, \ldots, y_i, \ldots, y_m] \).

The Relinearisation algorithm is essentially the algorithm obtained by applying a further linearisation mapping to the linearisation variety \( \mathcal{L} \). The Veronese embedding

\[
\varphi_{S^2(V)} : \mathbb{P} (S^2(V)) \rightarrow \mathbb{P} (S^2(S^2(V)))
\]

is a mapping of a projective space of dimension \( N = \frac{1}{2} n(n + 3) \) to a projective space of dimension at most \( \frac{1}{2} N(N + 3) \). The corresponding linearisation mapping \( \varphi_{S^2(V)} \) maps quadrics in \( \mathbb{P} (S^2(V)) \) to hyperplanes in \( \mathbb{P} (S^2(S^2(V))) \). This mapping \( \varphi_{S^2(V)} \) is the relinearisation mapping, and applying it to the linearisation variety gives

\[
\varphi_{S^2(V)} (\mathcal{L}) = \bigcap_{i=1}^{s} \varphi_V (Q_{\hat{f}_i}) = \bigcap_{i=1}^{s} H_{\hat{f}_i}.
\]

Suppose a point \( P \in \mathbb{P}(V) \) is a solution of the original homogeneous quadratic equation \( f_1 = \ldots = f_m = 0 \) in \( \mathbb{F}[x_0, x_1, \ldots, x_n] \), then (by construction) we have

\[
\varphi_{S^2(V)} (\varphi_V (P)) \in \varphi_{S^2(V)} (\mathcal{L}).
\]

Thus a mapping of a solution lies in the intersection of hyperplanes in a projective space, which can be easily calculated with basic algebra. If the original equation system has a unique solution and \( \bigcap_{i=1}^{s} H_{\hat{f}_i} \) is a unique (projective) point, then

\[
P = \varphi_V^{-1} \left( \varphi_{S^2(V)}^{-1} \left( \varphi_{S^2(V)} (\mathcal{L}) \right) \right).
\]

Thus the Relinearisation algorithm offers a technique for finding the solution to a system of quadratic equations. Furthermore, even if the Relinearisation algorithm fails to find the solution, the variety \( \varphi_{S^2(V)} (\mathcal{L}) \) could itself be relinearised to find a solution and so on.

### 5.2 An Efficient Relinearisation Algorithm

The Relinearisation algorithm is actually performed in a slightly different manner to that described above for reasons of efficiency [21]. The projective subspace

\[
\mathcal{H} = \bigcap_{i=1}^{m} \mathcal{H}_{\hat{f}_i} \subset \mathbb{P} (S^2(V))
\]

given by the intersection of the hyperplanes defined by the polynomials \( f_1, \ldots, f_m \) has (projective) dimension \( N - m \). Thus \( \mathcal{H} \) is the projectivisation of a vector space over \( \mathbb{F} \) of dimension \( N + 1 - m \). If we suppose that \( U \) is a generic vector space over \( \mathbb{F} \) of dimension \( N + 1 - m \), then we can define a bijective substitution mapping

\[
\psi_U : \mathbb{P}(U) \rightarrow \mathcal{H} \subset \mathbb{P} (S^2(V)).
\]
As $\psi_U$ is bijective, there exists an inverse mapping $\psi_U^{-1}: \mathcal{H} \to \mathbb{P}(U)$, so we can then define an equivalent linearisation variety $L' = \psi_U^{-1}(\mathcal{L}) \subset \mathbb{P}(U)$. This equivalent linearisation variety $L'$ is the intersection of $s$ quadrics, where $s \leq \frac{1}{2}n(n + 1)^2(n + 2)$.

The Veronese embedding for $\mathbb{P}(U)$ is $\varphi_U: \mathbb{P}(U) \to \mathbb{P}(S^2(U))$, where the projective geometry $\mathbb{P}(S^2(U))$ has dimension $\frac{1}{2}(N - m)(N - m + 3)$. Relinearisation of the equivalent linearisation variety $L'$ is achieved by applying the corresponding linearisation mapping $\overline{\varphi}_U$. The resulting variety $\overline{\varphi}_U(L')$ is the intersection of hyperplanes, so is easily calculated. If $P$ is a solution of the original equation system, then

$$\varphi_U(\psi_U^{-1}(\varphi_V(P))) \in \overline{\varphi}_U(L').$$

Thus if the original equation system has a unique solution and $\overline{\varphi}_U(L')$ is a unique (projective) point $P$, then the solution of the original equation system is given by

$$P = \varphi_V^{-1}\left(\varphi_U(\psi_U^{-1}(\overline{\varphi}_U(L'))))\right).$$

This is clearly a more efficient way of implementing the Relinearisation algorithm as it is performing calculations in the projective geometry $\mathbb{P}(S^2(U))$, which has smaller dimension than the original projective geometry $\mathbb{P}\left(S^2\left(S^2(V)\right)\right)$.

**Example 3.** Consider the following quadratic equation system

$$
\begin{align*}
0 &= 1 + x_1 + x_2 - x_1x_2 \\
0 &= 2 + x_2 + x_1^2 - x_2^2 \\
0 &= x_1 + x_2 - 2x_1^2 + 2x_1x_2 - x_2^2
\end{align*}
$$

with three equations in two variables over GF(37). This is the equation system given by the first three equations of Example 2 and has the unique solution $(x_1, x_2) = (2, 3)$. There are clearly not enough equations in this equation system to obtain this solution by the Linearisation algorithm. As before, we can homogenise these equations by the addition of a variable $x_0$ to give

$$
\begin{align*}
0 &= f_1 = x_0^2 + x_0x_1 + x_0x_2 - x_1x_2 \\
0 &= f_2 = 2x_0^2 + x_0x_2 + x_1^2 - x_2^2 \\
0 &= f_3 = x_0x_1 + x_0x_2 - 2x_1^2 + 2x_1x_2 - x_2^2,
\end{align*}
$$

which also defines a variety in $\mathbb{P}(V)$, where $V$ is a vector space of dimension 3, so $n = 2$. We can now apply the linearisation mapping $\varphi_V$ induced by the Veronese embedding $\varphi_V: \mathbb{P}(V) \to \mathbb{P}\left(S^2(V)\right)$ to give

$$
\begin{align*}
0 &= \gamma_1 = y_{00} + y_{01} + y_{02} - y_{12} \\
0 &= \gamma_2 = 2y_{00} + y_{02} + y_{11} - y_{22} \\
0 &= \gamma_3 = y_{01} + y_{02} - 2y_{11} + 2y_{12} - y_{22}.
\end{align*}
$$

The projective subspace $\mathcal{H}$ defined by the intersection of the subspaces $\mathcal{H}_{\mathcal{T}_i}$ of $S^2(V)$ defined by these equations is given by

$$\mathcal{H} = \langle (1, 0, 0, 0, 1, 2)^T, (0, 1, 0, 1, 1, 1)^T, (0, 0, 1, 13, 1, 14)^T \rangle \subset \mathbb{P}\left(S^2(V)\right).$$
If we let $U$ be a 3-dimensional vector space over $\text{GF}(37)$, then we can define a substitution mapping $\psi_U: \mathbb{P}(U) \to \mathcal{H}$ based on a $6 \times 3$ matrix $A$ with the property that if $u$ is a nonzero vector in $U$, then $\langle z \rangle = \psi_U((u)) \in \mathcal{H} \subset \mathbb{P}(V)$, where $z = Au \in S^2(V)$. The columns of $A$ define $\mathcal{H}$, so $A$ is given by

$$A = \begin{pmatrix} 1 & 0 & 0 & 1 & 2 \\ 0 & 1 & 0 & 1 & 1 \\ 0 & 0 & 1 & 3 & 14 \end{pmatrix}^T.$$  

The Veronese surface $V \subset \mathbb{P}(S^2(V))$ is defined as the intersection of the six quadrics

$$0 = y_0 y_1 - y_0^2, \quad 0 = y_0 y_2 - y_1^2, \quad 0 = y_1 y_2 - y_2^2,$$

$$0 = y_0 y_{12} - y_{01} y_{02}, \quad 0 = y_{02} y_{11} - y_{01} y_{12} \quad \text{and} \quad 0 = y_{01} y_{22} - y_{02} y_{12}.$$  

There exist six symmetric $6 \times 6$ matrices $M_i$ ($1 \leq i \leq 6$) such that the above quadrics defining the Veronese variety $V \subset \mathbb{P}(S^2(V))$ are given by $0 = y^T M_i y$. The linearisation variety $\mathcal{L} = V \cap U$ is contained in $\mathbb{P}(S^2(V))$. We use the equivalent linearisation variety $\mathcal{L}' = \psi_U^{-1}(\mathcal{L}) \subset \mathbb{P}(U)$ in a space of smaller dimension. Applying the substitution mapping $y = Az$ we obtain quadrics defining the equivalent linearisation variety $\mathcal{L}' \subset \mathbb{P}(U)$ given by $0 = (Az)^T M_i (Az) = z^T (A^T M_i A) z$. Thus the equivalent linearisation variety $\mathcal{L}'$ is defined by the intersection of the quadrics

$$0 = u_0 u_1 + 13 u_0 u_2 + 36u_1^2$$

$$0 = 2u_0^2 + u_0 u_1 + 14 u_0 u_2 + 36u_2^2$$

$$0 = 36u_0^2 + 24 u_0 u_2 + 25 u_1 u_2 + 33u_2^2$$

$$0 = u_0^2 + u_0 u_1 + u_0 u_2 + 36u_1 u_2$$

$$0 = 36u_0 u_1 + 36u_1^2 + 13u_2^2$$

$$0 = 2u_0 u_1 + 36 u_0 u_2 + u_1^2 + 13u_1 u_2 + 36u_2^2.$$  

We can now relinearise $\mathcal{L}' \subset \mathbb{P}(U)$ by applying the linearisation mapping $\varphi_U$ induced by the Veronese embedding $\varphi_U: \mathbb{P}(U) \to \mathbb{P}(S^2(U))$ to obtain $\varphi_U(\mathcal{L}')$ as the intersection of the hyperplanes defined by

$$\begin{pmatrix} 0 & 1 & 13 & 36 & 0 & 0 \\ 2 & 1 & 14 & 0 & 0 & 36 \\ 36 & 0 & 24 & 0 & 25 & 33 \\ 1 & 1 & 1 & 0 & 36 & 0 \\ 0 & 36 & 0 & 36 & 0 & 13 \\ 0 & 2 & 36 & 1 & 13 & 36 \end{pmatrix} \begin{pmatrix} w_{00} \\ w_{01} \\ w_{02} \\ w_{11} \\ w_{12} \\ w_{22} \end{pmatrix} = \begin{pmatrix} 0 \\ 0 \\ 0 \\ 0 \end{pmatrix}.$$  

Reducing this linear system to echelon form, we obtain

$$\begin{pmatrix} 1 & 0 & 0 & 0 & 0 & 4 \\ 0 & 1 & 0 & 0 & 0 & 8 \\ 0 & 0 & 1 & 0 & 0 & 12 \\ 0 & 0 & 0 & 1 & 0 & 16 \\ 0 & 0 & 0 & 0 & 1 & 24 \\ 0 & 0 & 0 & 0 & 0 & 0 \end{pmatrix} \begin{pmatrix} w_{00} \\ w_{01} \\ w_{02} \\ w_{11} \\ w_{12} \\ w_{22} \end{pmatrix} = \begin{pmatrix} 0 \\ 0 \\ 0 \\ 0 \end{pmatrix}.$$
We can thus solve this linear system to obtain
\[ \varphi_U(L') = \langle (4, 8, 12, 16, 24, -1)^T \rangle = \langle (1, 2, 3, 4, 6, 9)^T \rangle \in \mathbb{P}(S^2(U)). \]

Having obtained this solution, we can now back-track through the various mappings to obtain the unique solution to the original equation system. Applying the first inverse Veronese embedding, we have
\[ \varphi_U^{-1}(\varphi_U(L')) = \langle (1, 2, 3)^T \rangle \in \mathbb{P}(U). \]

Applying the substitution mapping \( \psi_U \) by calculating \( A(1, 2, 3)^T \) gives us
\[ \psi_U(\varphi_U^{-1}(\varphi_U(L'))) = \langle (1, 2, 3, 4, 6, 9)^T \rangle \in \mathbb{P}(S^2(V)). \]

We can now apply the last inverse Veronese embedding to give the solution as
\[ \varphi_V^{-1}(\psi_U(\varphi_U^{-1}(\varphi_U(L')))) = \langle (1, 2, 3)^T \rangle \in \mathbb{P}(V). \]

Thus we have \( (x_0, x_1, x_2) = \lambda(1, 2, 3) \), so taking \( x_0 = 1 \) gives \( (x_1, x_2) = (2, 3) \) as the unique solution of the original nonhomogeneous equation system.

\[ \square \]

5.3 A Matrix Rank Formulation of the Relinearisation Algorithm

The quadratic equation system defines a collection of quadrics in \( \mathbb{P}(V) \). After linearisation, we obtain a subspace \( \mathcal{H} \) of \( \mathbb{P}(S^2(V)) \) of (projective) dimension \( N - m \). However, the projective geometry \( \mathbb{P}(S^2(V)) \) can be defined by the vector space of symmetric \( (n+1) \times (n+1) \) matrices (Section 2.1). Thus, in terms of the vector space of symmetric matrices, the subspace \( \mathcal{H} \) is generated by \( N - m \) symmetric matrices \( H_1, \ldots, H_{N-m} \), that is
\[ \mathcal{H} = \langle H_1, \ldots, H_{N-m} \rangle, \]
so any point in \( \mathcal{H} \) is a linear combination of the above matrices.

The original quadratic equation system is analysed by considering \( \mathcal{H} \cap \mathcal{V}_V \). However, in terms of the vector space of symmetric matrices, the points of the Veronese surface \( \mathcal{V}_V \) are given by the matrices of rank 1 (Section 3.2). Thus \( \mathcal{H} \cap \mathcal{V}_V \) is given by the matrices of rank 1 in \( \mathcal{H} \). We can thus potentially solve the equation system by finding \( \lambda_0, \ldots, \lambda_{N-m-1} \in \mathbb{F} \) such that
\[ \text{Rank} \left( \sum_{t=1}^{N-m} \lambda_t M_t \right) = 1. \]

Thus evaluating the 2-minors of \( \sum_{t=1}^{N-m} \lambda_t M_t \) gives a system of multivariate quadratic equations in the variables \( \lambda_1, \ldots, \lambda_{N-m-1} \). This equation system defines the linearisation variety \( L' \) used in the efficient Relinearisation technique of Section 5.2.
Example 4. Consider the quadratic equation system of Example 3, namely

\[
\begin{align*}
0 &= 1 + x_1 + x_2 - x_1 x_2 \\
0 &= 2 + x_2 + x_1^2 - x_2^2 \\
0 &= x_1 + x_2 - 2x_1^2 + 2x_1 x_2 - x_2^2.
\end{align*}
\]

We saw that after homogenisation and linearisation (Example 3) we obtain the subspace \( \mathcal{H} \) of \( \mathbb{P}(S^2(V)) \) given by

\[
\mathcal{H} = \langle (1, 0, 0, 0, 1, 2)^T, (0, 1, 0, 1, 1)^T, (0, 0, 1, 13, 1, 14)^T \rangle.
\]

Expressing \( \mathbb{P}(S^2(V)) \) in terms of symmetric matrices, we obtain

\[
\mathcal{H} = \langle H_1, H_2, H_3 \rangle,
\]

where

\[
H_1 = \begin{pmatrix} 1 & 0 & 0 \\ 0 & 0 & 1 \\ 0 & 1 & 2 \end{pmatrix}, \quad H_2 = \begin{pmatrix} 0 & 1 & 0 \\ 1 & 1 & 1 \\ 0 & 1 & 1 \end{pmatrix}, \quad \text{and} \quad H_3 = \begin{pmatrix} 0 & 0 & 1 \\ 13 & 1 & 0 \\ 1 & 1 & 14 \end{pmatrix}.
\]

An arbitrary linear combination of these generating matrices gives

\[
\lambda_1 H_1 + \lambda_2 H_2 + \lambda_3 H_3 = \begin{pmatrix} \lambda_1 & \lambda_2 & \lambda_3 \\ \lambda_2 & \lambda_2 + 13\lambda_3 & \lambda_1 + \lambda_2 + \lambda_3 \\ \lambda_3 & \lambda_1 + \lambda_2 + \lambda_3 & 2\lambda_1 + \lambda_2 + 14\lambda_3 \end{pmatrix}.
\]

Evaluating the 2-minors of \( \lambda_1 H_1 + \lambda_2 H_2 + \lambda_3 H_3 \) gives the system of nine quadratic equations described by the matrix equation

\[
\begin{pmatrix} 0 & 1 & 1 & 36 & 0 & 0 \\ 1 & 1 & 1 & 0 & 36 & 0 \\ 0 & 1 & 0 & 1 & 0 & 24 \\ 36 & 36 & 36 & 0 & 1 & 0 \\ 35 & 36 & 23 & 0 & 0 & 1 \\ 0 & 35 & 1 & 36 & 24 & 1 \\ 0 & 1 & 0 & 1 & 0 & 24 \\ 0 & 2 & 36 & 1 & 13 & 36 \\ 36 & 0 & 24 & 0 & 25 & 33 \end{pmatrix} \begin{pmatrix} \lambda_1 \\ \lambda_2 \\ \lambda_3 \\ \lambda_1 \lambda_2 \\ \lambda_1 \lambda_3 \\ \lambda_2 \lambda_3 \\ \lambda_1^2 \\ \lambda_1 \lambda_3 \\ \lambda_2 \lambda_3 \\ \lambda_3^2 \end{pmatrix} = \begin{pmatrix} 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \end{pmatrix}.
\]

We reduce this linear system of rank 5 to obtain

\[
\begin{pmatrix} 1 & 0 & 0 & 0 & 4 \\ 0 & 1 & 0 & 0 & 8 \\ 0 & 0 & 1 & 0 & 12 \\ 0 & 0 & 1 & 0 & 16 \\ 0 & 0 & 0 & 1 & 24 \end{pmatrix} \begin{pmatrix} \lambda_1^2 \\ \lambda_1 \lambda_2 \\ \lambda_1 \lambda_3 \\ \lambda_2^2 \\ \lambda_2 \lambda_3 \\ \lambda_3^2 \end{pmatrix} = \begin{pmatrix} 0 \\ 0 \\ 0 \\ 0 \\ 0 \end{pmatrix}.
\]

We thus have \( \lambda_1 = -12\lambda_3 \) and \( \lambda_2 = -24\lambda_3 \), so \( \lambda_3 = 3\lambda_1 \) and \( \lambda_2 = 2\lambda_1 \), so we obtain

\[
\begin{pmatrix} 1 & 0 & 0 \\ 0 & 0 & 1 \\ 0 & 1 & 2 \end{pmatrix} + 2 \begin{pmatrix} 0 & 1 & 0 \\ 1 & 1 & 1 \\ 0 & 1 & 1 \end{pmatrix} + 3 \begin{pmatrix} 0 & 0 & 1 \\ 13 & 1 & 0 \\ 1 & 1 & 14 \end{pmatrix} = \begin{pmatrix} 1 & 2 & 3 \\ 2 & 4 & 6 \\ 3 & 6 & 9 \end{pmatrix}.
\]
The matrix on the right has rank 1 and corresponds to the projective point \( \langle 1, 2, 3 \rangle \), which is the solution of Example 3. We note that the final linear system of both this example and that of Example 3 defining the equivalent linearisation variety \( L' \) are identical. \( \square \)

5.4 Failure of the Relinearisation Algorithm

Example 3 illustrates one of the complications that can arise during relinearisation. The six quadratic equations defining the Veronese surface in \( \mathbb{P}(S^2(V)) \) (projective dimension 5) are linearly independent. However, there is no guarantee that their respective restrictions to a given subspace are independent. In Example 3, the restriction of the six quadratic equations to the projective subspace \( H \) (projective dimension 2) gives a system of rank 5. The analysis of the Relinearisation algorithm given in [21] does not take this issue into account, so the estimates given there for its successful application can be overly optimistic. Example 5 illustrates this point.

Example 5. We consider eight homogeneous polynomials in four variables over \( \text{GF}(37) \) given by

\[
\begin{pmatrix}
17 & 18 & 18 & 12 & 5 & 21 & 11 & 22 & 4 & 32 \\
15 & 32 & 17 & 23 & 4 & 33 & 18 & 13 & 26 & 8 \\
10 & 32 & 20 & 20 & 8 & 27 & 32 & 19 & 20 & 10 \\
11 & 30 & 23 & 31 & 14 & 5 & 2 & 35 & 14 & 16 \\
9 & 11 & 3 & 17 & 24 & 10 & 16 & 3 & 27 & 23 \\
23 & 25 & 11 & 4 & 13 & 8 & 8 & 32 & 31 & 18 \\
13 & 17 & 5 & 29 & 19 & 18 & 23 & 34 & 17 & 16 \\
8 & 28 & 25 & 19 & 35 & 8 & 36 & 21 & 1 & 1
\end{pmatrix}
\begin{pmatrix}
x_0^2 \\
x_0x_1 \\
x_0x_2 \\
x_0x_3 \\
x_1^2 \\
x_1x_2 \\
x_1x_3 \\
x_2^2 \\
x_2x_3 \\
x_3^2
\end{pmatrix}
\]

If we let \( S \) denote the above 8×10 matrix over \( \text{GF}(37) \) and \( x \) the vector of quadratic monomials, then the equation system \( Sx = 0 \) has the unique (projective) solution \( \langle 1, 6, 14, 5 \rangle^T \). If this equation system had a ninth independent equation, then we could solve this system by the Linearisation algorithm. Thus the equation system \( Sx = 0 \) is almost fully linearised.

We consider the above equation system in terms of the vector space \( V \) of dimension 4 over \( \text{GF}(37) \), so \( n = 3 \). This equation system gives eight quadrics in \( \mathbb{P}(V) \). The Veronese embedding \( \varphi_V : \mathbb{P}(V) \rightarrow \mathbb{P}(S^2(V)) \) embeds this projective geometry of dimension 3 in one of dimension \( N = \frac{1}{2}(3 \cdot 6) = 9 \). This Veronese embedding \( \varphi_V \) induces a linearisation mapping \( \varphi_V \). Applying \( \varphi_V \) to this equation system gives the linear system \( Sg = 0 \), where \( (y_{00}, \ldots, y_{ij}, \ldots, y_{33})^T \) are the variables used to define \( \mathbb{P}(S^2(V)) \). Solutions to this linear system are contained in the intersection \( \mathcal{H} \subset \mathbb{P}(S^2(V)) \) of the 8 hyperplanes, a projective subspace \( \mathcal{H} \) with (projective) dimension 1 and defined by

\[
\mathcal{H} = \langle \langle 1, 0, 13, 21, 1, 31, 22, 20, 30, 0 \rangle^T, \langle 0, 1, 31, 22, 12, 15, 26, 17, 19, 35 \rangle^T \rangle.
\]
If we let $U$ denote a generic vector space of dimension 2 over GF(37), then $\mathbb{P}(U)$ is a projective geometry of dimension 1 (a projective line). We can now define a bijective substitution mapping $\psi_U : \mathbb{P}(U) \to \mathcal{H}$ based on the $10 \times 2$ matrix $A = \begin{pmatrix} 1 & 0 & 13 & 21 & 1 & 31 & 22 & 20 & 30 & 0 \\ 0 & 1 & 31 & 22 & 12 & 15 & 26 & 17 & 19 & 35 \end{pmatrix}^T$.

The Veronese variety $\mathcal{V}_V \subset \mathbb{P}(S^2(V))$ can be defined as the intersection of 20 quadrics. Thus there exist twenty $10 \times 10$ matrices $M_i$ such that $y^T M_i y = 0$. The linearisation variety is given by $L = \mathcal{V}_V \cap \mathcal{H} \subset \mathbb{P}(S^2(V))$. The substitution mapping $\psi_U$ allows us to define an equivalent linearisation variety $L' = \psi_U^{-1}(L) \subset \mathbb{P}(U)$ in a space of dimension 1. Applying the substitution mapping gives twenty quadrics $z^T (A^T M_i A) z$ ($i = 1, \ldots, 20$) defining the equivalent linearisation variety $L'$. Thus the equivalent linearisation variety $L'$ is given by $L u = 0$, where $u = (u^2_0, u_0 u_1, u^2_1)^T$ and $L$ is the $3 \times 20$ matrix

\[
\begin{pmatrix}
1 & 31 & 22 & 36 & 16 & 4 & 15 & 19 & 5 & 7 & 36 & 21 & 14 & 34 & 9 & 6 & 25 \\
36 & 6 & 15 & 12 & 21 & 34 & 13 & 21 & 33 & 22 & 18 & 32 & 30 & 1 & 16 & 23 & 3 & 28 & 31 & 12
\end{pmatrix}.
\]

The Relinearisation algorithm requires us to linearise the above linearisation variety $L'$. The Veronese embedding $\varphi_U : \mathbb{P}(U) \to \mathbb{P}(S^2(U))$ embeds $\mathbb{P}(U)$ in a projective space of dimension $\frac{1}{2} (1 \cdot 4) = 2$. When we apply this embedding to the above variety, we obtain the variety

$X = \{ (w_{00}, w_{01}, w_{11})^T \in \mathbb{P}(S^2(U)) \mid L(w_{00}, w_{01}, w_{11})^T = 0 \} \subset \mathbb{P}(S^2(U))$.

For the Relinearisation algorithm to succeed, we require that $X \subset \mathbb{P}(S^2(U))$ is a unique (projective) point. This condition requires that the matrix $L$ has rank 2. However, the matrix $L$ has rank 1 as every row is a multiple of $(1, 12, 36)$. Thus the direct Relinearisation algorithm fails to find the solution of this equation system.

This system could be easily solved from information given by the above process. For example, we know that $u^2_0 + 12u_0 u_1 + 36u^2_1 = (u_0 + 6u_1)^2 = 0$. However, such a technique would not work if we were solving a system with seven of the original eight equations. In any case, the main point of this example is to illustrate that even in an almost fully linearised equation system, the direct Relinearisation algorithm can fail. □

### 5.5 Tangent Spaces

An interesting characterisation for when Relinearisation succeeds or fails can be obtained by considering the tangent spaces to the Veronese variety. Suppose we have a system of $m$ quadrics intersecting in a unique (projective) point $P$ in $\mathbb{P}(V)$. The linearisation variety $\mathcal{L}$ is the intersection of the Veronese variety $\mathcal{V}_V$ with the subspace $\mathcal{H}$ defined by linearising the original quadratic system.
(Section 4.2). This linearisation variety $\mathcal{L}$ can be defined as the intersection of $s$ quadrics, so we have

$$\mathcal{L} = \mathcal{V}_V \bigcap \mathcal{H} = \bigcap_{i=1}^{s} Q_{\bar{f}_i} \subset \mathbb{P}(S^2(V)).$$

We first suppose that the Reлинаrisation algorithm succeeds for this system. In this case, we know that

$$\varphi_V(P) = \mathcal{L} = \bigcap_{i=1}^{s} Q_{\bar{f}_i},$$

so we have a full-rank system of quadrics whose intersection is $\varphi_V(P)$. The (projective) $(N-m-1)$-dimensional tangent space to the quadric $Q_{\bar{f}_i}$ at $\varphi_V(P)$ is denoted by $T_{\varphi_V(P)}(Q_{\bar{f}_i})$ (Section 2.3). The intersection of all these tangent spaces is the unique point $\varphi_V(P)$, that is

$$\varphi_V(P) = \bigcap_{i=1}^{s} T_{\varphi_V(P)}(Q_{\bar{f}_i}).$$

Conversely, if the intersection of these tangent spaces is not a unique point, then the Reлинаrisation algorithm fails. We now consider the linear subspace

$$\mathcal{H} \bigcap \bigcap_{i=1}^{s} T_{\varphi_V(P)}(Q_{\bar{f}_i}) \subset \mathbb{P}(S^2(V)),$$

which has the same dimension as

$$\bigcap_{i=1}^{s} T_{\varphi_V(P)}(Q_{\bar{f}_i}).$$

This gives us a criterion for the success or failure of the Reлинаrisation algorithm to provide a unique solution without actually having to relinearise. If the intersection of the linear space $\mathcal{H}$, given directly by linearising the quadratic system, and the tangent space to the Veronese variety at $\varphi_V(P)$ is not a single point, then the Reлинаrisation algorithm fails.

**Example 6.** We consider the equation system of Example 3 with unique solution $P = (1, 2, 3)^T$. In this case, the vector space $V$ has dimension 3 over GF(37) (so $n = 2$). The space $\mathcal{H}$ is the (projective) 2-dimensional subspace of $\mathbb{P}(S^2(V))$ given by the kernel of the matrix

$$\begin{pmatrix}
1 & 1 & 1 & 0 & -1 & 0 \\
2 & 0 & 1 & 1 & 0 & -1 \\
0 & 1 & 1 & 2 & 2 & -1
\end{pmatrix}.$$

The tangent space to the Veronese surface $\mathcal{V}_V$ at $\varphi_V(P)$ is a (projective) 2-dimensional subspace of $\mathbb{P}(S^2(V))$ given by the kernel of the matrix

$$\begin{pmatrix}
1 & 0 & 24 & 0 & 0 & 33 \\
0 & 1 & 24 & 0 & 12 & 29 \\
0 & 0 & 0 & 1 & 11 & 21
\end{pmatrix}.$$
We can construct a $6 \times 6$ matrix by combining these two matrices. This larger matrix has rank 5, so the intersection of the tangent space to the Veronese surface at $\varphi_V(P)$ with $\mathcal{H}$ is the unique (projective) point $P$. Thus the Relinearisation algorithm succeeds for Example 3.

By contrast, we can consider the equation system of Example 5 with unique (projective) solution $P = \langle (1, 6, 14, 5) \rangle$. In this case, the vector space $V$ has dimension 4 over $\text{GF}(37)$ (so $n = 3$). The space $\mathcal{H}$ is a (projective) 1-dimensional subspace of the 9-dimensional projective geometry $\mathbb{P}(S^2(V))$ and is given by the kernel of a $8 \times 10$ matrix. The tangent space to the Veronese variety at $\varphi_V(P)$ is a 3-dimensional subspace of $\mathbb{P}(S^2(V))$ given by the kernel of a $6 \times 10$ matrix. Combining these two matrices gives an $14 \times 10$ matrix that only has rank 8, so the intersection of the tangent space to the Veronese surface at $\varphi_V(P)$ with $\mathcal{H}$ is not a unique (projective) point. Thus the Relinearisation algorithm fails for Example 5. □

6 A Geometric View of the XL Algorithm

The XL or extended linearisation algorithm was proposed to be a “simplified and improved version of relinearisation” [10]. We now consider some geometric properties of the XL algorithm. The original description of the XL algorithm of [10] is given for a non-homogeneous equation system. We thus term the original XL algorithm description the AffineXL algorithm. There is a natural generalisation of the AffineXL algorithm to a homogeneous equation system, which we term the ProjectiveXL algorithm. The ProjectiveXL algorithm is thus more mathematically natural, and we also consider its properties.

6.1 The AffineXL Algorithm

Without loss of generality, we consider the application of the AffineXL algorithm to a quadratic equation system. The basic idea of the AffineXL algorithm is to multiply the polynomials of this original equation system by monomials of degree up to $D - 2$ to obtain many polynomials of degree at most $D$. We then regard this degree $D$ polynomial system as a linear system in the monomials of degree at most $D$. It is then hoped that the linear span of the generated polynomials in this larger system contains a univariate polynomial in one of the variables $x_i$. An ordering of the monomials of degree at most $D$ is chosen such that such a univariate polynomial in $x_i$ can be found simply by reducing the matrix of this system to echelon form. The generated univariate polynomial can be factored using Berlekamp’s algorithm [25] or some other method to give values for one of the variables $x_i$. We could then substitute these values for $x_i$ to obtain a smaller quadratic system. This smaller system could then potentially be analysed using the AffineXL algorithm or some other technique to enable a full solution to be found. Clearly, the smaller the value of $D$, the degree of the generated polynomials for which this is possible, the faster the AffineXL
− **Input.** $m$ homogeneous independent quadratic equations in $n+1$ variables.

1. Generate the $m\binom{D-2+n}{D-2}$ possible polynomials of degree at most $D$ that are formed by multiplying each of the polynomials of the original system by monomials of degree at most $D-2$.

2. Choose an ordering of the monomials of degree at most $D$. Linearise this new system of polynomials of degree at most $D$ and perform a Gaussian reduction. The ordering of monomials should be chosen in such a way that this process yields a univariate polynomial in just one of the variables.

3. Note that it is not always possible to find such an ordering, and in this case the **AffineXL** algorithm fails for degree $D$.

4. This univariate polynomial can be factored using Berlekamp’s algorithm [25]. This potentially allows the elimination of a variable from the original system of equations.

5. This process is repeated on the new smaller system and so on, potentially eliminating further variables.

6. Substitution is used to find values for the eliminated variables.

− **Output.** Solution set for the original equation system (if method is successful).

---

**Fig. 1.** Basic Description of the **AffineXL** Algorithm for a Quadratic System

The algorithm works. We give a fuller description of the basic form of the **AffineXL** algorithm in Figure 1 and a simple example in Example 7.

We note that such an ordering of monomials does not have to be a monomial ordering, in the sense of compatibility with multiplication, and which is required for Gröbner basis calculations [11]. The only requirement for the ordering of monomials in the **AffineXL** algorithm is that the ordering naturally partitions the set of monomials into two classes, with one class containing all the monomials in $x_i$ alone and the complementary class not containing any monomials in $x_i$. However, we do note that the lexicographic monomial ordering [11] naturally gives this partition, and it has been noted that the **AffineXL** algorithm works in a similar manner to the F4 algorithm [14] for the calculation of a Gröbner basis using the lexicographic ordering [1].

**Example 7.** We consider the homogenised version of the equation system defined by two quadratic polynomials $f_1$ and $f_2$ in two variables over $GF(37)$ given by

\[ f_1 = x_1^2 + 5x_1x_2 + 15 \quad \text{and} \quad f_2 = x_2^2 + 9x_1x_2 + 23. \]

We wish to find solutions to $f_1 = f_2 = 0$. The application of the XL algorithm to such a quadratic system is discussed in [6, 10]. In order to apply the **AffineXL** algorithm with $D = 2$, that is using the original equation system with no monomial multiplication, we would need to find a linear combination $\lambda_1f_1 + \lambda_2f_2$ which is a univariate polynomial in either solely in $x_1$ or solely in $x_2$. 

The equation system \( f_1 = f_2 = 0 \) can be represented as the kernel of the matrix
\[
\begin{pmatrix}
0 & 5 & 0 & 1 & 0 & 15 \\
1 & 9 & 0 & 0 & 0 & 23
\end{pmatrix}
\]
with respect to the column ordering \((x_2^2, x_1x_2, x_2, x_1^2, x_1, 1)\). Reducing this matrix to echelon form gives
\[
\begin{pmatrix}
1 & 0 & 0 & 13 & 0 & 23 \\
0 & 1 & 0 & 25 & 0 & 5
\end{pmatrix}
\]
Thus there is no polynomial in the linear span of \( f_1 \) and \( f_2 \) which is a univariate polynomial in \( x_1 \) alone. Similarly, there is no polynomial in the linear span of \( f_1 \) and \( f_2 \) which is a univariate polynomial in \( x_2 \) alone.

We next consider the linear span of the cubic polynomials \( x_i f_j \), that is the \( D = 3 \) case. However, this linear span does not contain any polynomials in \( x_1 \) alone or in \( x_2 \) alone. We therefore consider the \( D = 4 \) case and calculate all quartic polynomials \( x_i x_j f_j \), and find that the linear span of these polynomials contains
\[
x_1^4 + 10x_1^2 + 26 = (x_1 - 1)(x_1 - 10)(x_1 - 27)(x_1 - 36).
\]
We would thus obtain the four solutions to \( f_1 = f_2 = 0 \) in \( \text{GF}(37) \), namely
\[
(x_1, x_2) = (1, 19), (10, 31), (27, 6), \text{ or } (36, 18).
\]
Thus the application of the AffineXL algorithm requires that we multiply the two original polynomials by all monomials of degree 2 for the AffineXL algorithm to succeed, that is we take \( D = 4 \).

\[\square\]

6.2 The ProjectiveXL Algorithm

The AffineXL algorithm is designed for non-homogeneous polynomial equation systems (despite the comment to the contrary in [10]). However, any non-homogeneous equation system in variables \( x_1, \ldots, x_n \) can be transformed into a homogeneous system in the variables \( x_0, x_1, \ldots, x_n \) by the inclusion of a homogenising variable \( x_0 \). We thus give a description of an XL-type algorithm as it applies to a homogeneous multivariate quadratic system defined by \( f_1, \ldots, f_m \in \mathbb{F}[x_0, x_1, \ldots, x_n] \), and we term this version of the XL algorithm for a homogeneous equation system the ProjectiveXL algorithm.

Without loss of generality, we consider the application of the ProjectiveXL algorithm to a homogeneous quadratic equation system. In a similar manner to the AffineXL algorithm, we multiply the polynomials of this original equation system by monomials of degree \( D - 2 \) to obtain many polynomials of degree \( D \). We then regard this homogeneous degree \( D \) polynomial system as a linear system in the monomials of degree \( D \). The aim of the ProjectiveXL algorithm is that the linear span of the generated polynomials in this larger system contains a bivariate polynomial in two of the variables \( x_i \) and \( x_j \). An ordering of the degree \( D \) monomials is then chosen such that such a bivariate polynomial
can be easily found by a simple matrix reduction. Such a homogeneous bivariate polynomial \( f(x_i, x_j) \) of degree \( D \) could then potentially be factored directly. One common technique when \( x_j \neq 0 \) is to apply a univariate factorisation technique to \( x_j^{-D} f(x_i, x_j) \), which can be regarded as a univariate polynomial in \( \frac{x_i}{x_j} \). A factorisation of \( f(x_i, x_j) \) would allow us to substitute values of \( x_i \) by some multiple of \( x_j \), thus obtaining a smaller equation system.

In a similar manner to the AffineXL algorithm (Section 6.1), the ordering used by the ProjectiveXL algorithm does not have to be a *monomial ordering*, but merely one that partitions the monomials into a class containing monomials in \( x_i \) and \( x_j \) alone and the complementary class. Furthermore, we have already noted the connection between the AffineXL algorithm and Gröbner basis algorithms under the lexicographic ordering (Section 6.1). Similarly, the ProjectiveXL algorithm can be viewed as a variant of the R1 and R2 algorithms of [23], as these algorithms are Gröbner basis techniques based on monomial orderings in a homogenised equation system.

This ProjectiveXL algorithm thus retains all the features of the AffineXL algorithm, yet the homogeneous description can provide greater flexibility and fits more naturally into a geometric setting. We give a fuller description of the ProjectiveXL algorithm in Figure 2. The original or AffineXL algorithm can be thought of as the special case of the special case of the ProjectiveXL algorithm in which one of the two variables \( x_i \) and \( x_j \) is restricted to being the homogenising variable \( x_0 \). Consequently, the bivariate equation produced by the algorithm in this case can be regarded as a univariate equation in \( \frac{x_i}{x_j} \). The greater power offered by the ProjectiveXL algorithm is illustrated by Example 8.

**Example 8.** We consider the homogenised version of the equation system of Example 7. We thus consider the homogeneous quadratic polynomials \( f_1 \) and \( f_2 \) in three variables over \( \text{GF}(37) \) given by

\[
\begin{align*}
    f_1 &= 15x_0^2 + x_1^2 + 5x_1x_2 \\
    f_2 &= 23x_0^2 + x_2^2 + 9x_1x_2.
\end{align*}
\]

We wish to the ProjectiveXL algorithm with \( D = 2 \), that is using the original equation system with no monomial multiplication. We consider the monomial ordering \((x_0^2, x_0x_1, x_0x_2, x_1^2, x_1x_2, x_2^2)\), and the echelon form of the defining matrix of Example 7 is given by

\[
\begin{pmatrix}
    1 & 0 & 0 & 0 & 2 & 29 \\
    0 & 0 & 0 & 1 & 12 & 9
\end{pmatrix}
\]

with respect to this ordering. Thus the linear span of \( f_1 \) and \( f_2 \) contains

\[
23f_1 - 15f_2 = x_1^2 + 12x_1x_2 + 9x_2^2 = (x_1 - 2x_2)(x_1 - 23x_2),
\]

so we obtain \( x_1 = 2x_2 \) or \( x_1 = 23x_2 \). Substituting these two values into \( f_1 \) gives

\[
\begin{align*}
    15x_0^2 + 14x_2^2 &= 15(x_0 - 2x_2)(x_0 - 35x_2) \\
    15x_0^2 + 15x_2^2 &= 15(x_0 - 6x_2)(x_0 - 31x_2)
\end{align*}
\]

respectively. We thus obtain the full (projective) solution as

\[
\langle(x_0, x_1, x_2)^T\rangle \in \{\langle(1, 1, 19)^T\rangle, \langle(1, 10, 31)^T\rangle, \langle(1, 27, 6)^T\rangle, \langle(1, 36, 18)^T\rangle\}.
\]
\begin{itemize}
    \item \textbf{Input.} \( m \) homogeneous independent quadratic equations in \( n + 1 \) variables.
    \item 1. Generate the \( m(D-2)^{n+1} \) possible polynomials of degree \( D \) that are formed by multiplying each of the polynomials of the original system by some monomial of degree \( D-2 \).
    \item 2. Choose an ordering of the degree \( D \) monomials. Linearise the new system of polynomials of degree \( D \) and perform a Gaussian reduction. The ordering of monomials should be chosen in such a way that this process yields a polynomial in just two of the original variables, say \( x_i \) and \( x_j \).
    \item 3. Note that it is not always possible to find such an ordering, and in this case the \texttt{ProjectiveXL} algorithm fails for degree \( D \).
    \item 4. This bivariate polynomial in \( x_i \) and \( x_j \) can be considered to be a univariate polynomial equation in \( \frac{x_i}{x_j} \). This univariate polynomial can be factored using Berlekamp’s algorithm \cite{25}. This potentially allows the elimination of a variable from the original system of equations.
    \item 5. This process is repeated on the new smaller system and so on, potentially eliminating further variables.
    \item 6. Substitution is used to find values for the eliminated variables.
\end{itemize}

\textbf{Output.} Solution set for the original equation system (if method is successful).

\begin{figure}[h]
\centering
\includegraphics[width=\textwidth]{fig2.png}
\caption{Basic Description of the \texttt{ProjectiveXL} Algorithm for a Quadratic System}
\end{figure}

Examples 7 and 8 show that the \texttt{ProjectiveXL} algorithm can be much more efficient than the \texttt{AffineXL} algorithm. On essentially the same equation system, the \texttt{ProjectiveXL} algorithm only required the use of quadratic polynomials (\( D = 2 \)), whereas the \texttt{AffineXL} algorithm required the use of quartic polynomials (\( D = 4 \)). Furthermore, the \texttt{ProjectiveXL} algorithm offers far more scope for minimising the value of \( D \) than the \texttt{AffineXL} algorithm. In an equation system with \( n \) variables, the \texttt{AffineXL} algorithm offers \( n \) different methods of constructing a suitable univariate polynomial of minimal degree (\( D \)), one for each variable. By contrast, the \texttt{ProjectiveXL} algorithm applied to the equivalent homogeneous equation system offers \( \left( \binom{n+1}{2} \right) \approx \frac{1}{2}n^2 \) different methods of constructing a suitable bivariate polynomial. Thus the \texttt{AffineXL} algorithm can be seen as a very small special case of the \texttt{ProjectiveXL} algorithm which restricts itself to a small and usually arbitrary set of special cases of the \texttt{ProjectiveXL} algorithm.

\subsection{6.3 Geometric Aspects of the \texttt{ProjectiveXL} Algorithm}

We now discuss the geometric aspects of the \texttt{ProjectiveXL} algorithm. This requires the use of the geometric terms \textit{primal}, \textit{secundum} and \textit{collineation}, which are defined in Section 2.3. We suppose that the homogeneous quadratic system has a unique (projective) solution. The homogeneous quadratic system defines
a system of quadrics in \( \mathbb{P}(V) \) which intersect in a unique projective point \( P \) corresponding to this unique solution. In the \texttt{ProjectiveXL} algorithm with degree \( D \), we multiply each polynomial by monomials of degree \( D - 2 \). Geometrically, this gives a system of primals of degree \( D \) that have a unique intersection at the (projective) point \( P \). Clearly any linear combination of the defining polynomials of the above primals gives another primal which also contains \( P \). The next step in the \texttt{ProjectiveXL} algorithm is to find a degree \( D \) primal whose defining polynomial is in the linear span of the polynomials defining the generated degree \( D \) primals, but which involves only two coordinates \( x_i \) and \( x_j \). Such a primal is defined by some bivariate polynomial

\[
g(x_i, x_j) = a_0 x_i^D + a_1 x_i^{D-1} x_j + \ldots a_{D-1} x_i x_j^{D-1} + a_D x_j^D.
\]

We note that the secundum \( S = \{ x \in \mathbb{P}(V) | x_i = x_j = 0 \} \) is contained in the primal defined by \( g(x_i, x_j) \). The bivariate polynomial \( g \) factorises over some extension field \( \mathbb{F} \) as

\[
g(x_i, x_j) = (\theta_1 x_i - \theta_1' x_j) \ldots (\theta_D x_i - \theta_D' x_j).
\]

If we define \( \mathbb{V} \) to be the vector space of dimension \( n + 1 \) over this extension field \( \mathbb{F} \), then each of these factors defines a hyperplane in \( \mathbb{P}(\mathbb{V}) \). Thus the primal defined by \( g \) is a product of hyperplanes in \( \mathbb{P}(\mathbb{V}) \), each of which contain the secundum \( S \). However, if the original equation system has a unique (projective) solution in \( \mathbb{F} \), then we need only consider the hyperplanes defined by the linear factors of \( g(x_i, x_j) \) which are defined over \( \mathbb{F} \). Thus we know the solution point \( P \) lies on one such hyperplane. We can analyse each such hyperplane by projecting the whole system into that hyperplane. This effectively removes a variable from the original system, and we can now examine the smaller system by the same method and so on.

In the \texttt{ProjectiveXL} algorithm, the fundamental aim is to find a primal defined by a bivariate polynomial. However, the property of being defined by a bivariate polynomial is not a geometrical property of the primal. A collineation of the projective geometry can transform a primal defined by a bivariate equation into a primal defined by a polynomial that is not bivariate. This is illustrated by Example 9.

\textbf{Example 9.} We consider the homogeneous quadratic polynomials in three variables over \( \text{GF}(37) \) given by

\[
f_1 = 6x_0^2 + 2x_0 x_1 + 3x_0 x_2 + \quad x_1^2 + 16x_1 x_2 + 3x_2^2
\]

and

\[
f_2 = 18x_0^2 + 35x_0 x_1 + 15x_0 x_2 + 26x_1^2 + 12x_1 x_2 + \quad x_2^2.
\]

We wish to apply the \texttt{ProjectiveXL} algorithm to the system \( f_1 = f_2 = 0 \), and there are three possible pairs of variables, namely \((x_0, x_1), (x_0, x_2)\) and \((x_1, x_2)\), in which we can construct a bivariate polynomial. Unfortunately, in all three cases, we are forced to use quartic polynomials \((D = 4)\) before we can do so.
However, this polynomial system is derived from that of Example 8 by the linear mapping
\[
\begin{pmatrix}
  x_0 \\
  x_1 \\
  x_2
\end{pmatrix} \mapsto \begin{pmatrix}
  2 & 26 & 10 \\
  26 & 4 & 13 \\
  33 & 21 & 2
\end{pmatrix} \begin{pmatrix}
  x_0 \\
  x_1 \\
  x_2
\end{pmatrix},
\]
but the equation system of Example 8 can be solved by only using quadratic polynomials \((D = 2)\). In geometrical terms, both this equation system and that of Example 8 define a pair of intersecting quadrics in \(\PG(2, \GF(37))\), and there is a collineation mapping one pair to the other. Thus this equation system and that of Example 8 are geometrically equivalent. \(\square\)

7 A Geometrically Invariant XL Algorithm

The aim of the ProjectiveXL algorithm for a homogeneous equation system with a small number of (projective) solutions is to find a primal defined by a bivariate polynomial which contains the points corresponding to the solutions. However, as we saw in Section 6.3 the property of being defined by a bivariate primal is not a geometrical property of the primal. Nonetheless, a primal defined by a bivariate polynomial does have definite geometric properties that are geometrically invariant. This section considers these properties, using the geometrical terms defined in Section 2.3, to derive the GeometricXL algorithm, which is invariant under collineations of the projective space. By contrast, Gröbner basis algorithms and XL-type algorithms are not geometrically invariant, though we note that the equation solving algorithm of \([22, 24]\) is geometrically invariant.

7.1 The GeometricXL Algorithm

Suppose we have a homogeneous equation system \(f_1 = \ldots = f_m = 0\) in \((n + 1)\) variables \(x_0, x_1, \ldots, x_n\) over a finite field \(\mathbb{F}\), and that this system has a few (projective) solutions. As before, we suppose that \(V\) denotes the vector space of dimension \((n + 1)\) over \(\mathbb{F}\). The ProjectiveXL algorithm generates a number of primals of degree \(D\) whose intersection contains the (projective) points corresponding to the solutions. As discussed in Section 6.3, the next step of the ProjectiveXL algorithm is to find a primal of degree \(D\) defined by a bivariate polynomial \(g\), which factorises over some extension field \(\overline{\mathbb{F}}\) as
\[
g(x_i, x_j) = (\theta_1 x_i - \theta'_1 x_j) \ldots (\theta_D x_i - \theta'_D x_j).
\]
If \(\overline{V}\) denotes the vector space of dimension \((n + 1)\) over the extension field \(\overline{\mathbb{F}}\), then the variety in \(\overline{\mathbb{P}(V)}\) defined by \(g(x_i, x_j)\) consists of \(D\) (not necessarily distinct) hyperplanes from the pencil of hyperplanes in \(\overline{\mathbb{P}(V)}\) generated by the hyperplanes given by the equations \(x_i = 0\) and \(x_j = 0\). Over \(\overline{\mathbb{F}}\), the polynomial \(g\) splits into factors that are irreducible over \(\mathbb{F}\). The variety in \(\mathbb{P}(V)\) described by an irreducible factor of \(g\) consists of the intersection of \(\mathbb{P}(V)\) with the conjugate hyperplanes of \(\overline{\mathbb{P}(V)}\) defined by this irreducible factor. This intersection is a
secundum of \( \mathbb{P}(V) \) since all of the conjugate hyperplanes come from a single pencil. This property of the primal being composed of hyperplanes from a pencil is clearly invariant under collineations, and it is this property of the primal, rather than that of being defined by a bivariate polynomial, that we exploit. A collineation of \( \mathbb{P}(V) \) maps the primal defined by \( g \) to one defined by

\[
(\theta_1 L - \theta'_1 L') \ldots (\theta_D L - \theta'_D L'),
\]

where \( L \) and \( L' \) are some linear polynomials over \( \mathbb{F} \). The GeometricXL algorithm is the generalisation of the ProjectiveXL algorithm which attempts to find primals of the above generalised form.

Suppose the multiplication step of the ProjectiveXL algorithm yields homogeneous polynomials \( h_1, \ldots, h_k \) of degree \( D \). In order to use a primal of the above form, we need to find a homogeneous polynomial \( h \) of degree \( D \) and \( \lambda_1, \ldots, \lambda_k \in \mathbb{F} \) such that

\[
h = \sum_{i=1}^{k} \lambda_i h_i = \prod_{j=1}^{D} (\theta_j L - \theta'_j L')
\]

for some linear polynomials \( L \) and \( L' \). Geometrically, a factor \( (\theta_j L - \theta'_j L') \) of the above expression defines a hyperplane in a pencil of hyperplanes defined by the hyperplanes \( L = 0 \) and \( L' = 0 \) (Section 2.3). Thus the primal \( \mathcal{V}(h) \) defined by \( h \) can be thought as a product of \( D \) hyperplanes all from the same pencil.

We now suppose that \( D \) is smaller than the positive characteristic \( p \) of the finite field \( \mathbb{F} \). We can take the formal \((D - 1)^{th}\) partial derivative of the above expression with respect to any monomial \( x = x_{j_1} \ldots x_{j_{D-1}} \) of degree \((D - 1)\). As in Section 2.2, we use the notation \( \mathbf{D}_x^{D-1} \) to denote the formal \((D - 1)^{th}\) partial derivative with respect to a degree \((D - 1)\) monomial \( x \), so we can obtain the linear polynomial

\[
\mathbf{D}_x^{D-1} h = \sum_{i=1}^{k} \lambda_i \mathbf{D}_x^{D-1} h_i = a_x L + a'_x L',
\]

where \( a_x \) and \( a'_x \) are constants. However, any such linear polynomial can be represented by a (row) vector of length \( n + 1 \), so this expression can be interpreted as a vector expression. Thus the partial derivatives matrix \( C_{h_i}^{(D-1)} \) of Section 2.2, whose rows are the various \((D - 1)^{th}\) partial derivatives of \( h_i \), is given by

\[
C_{h_i}^{(D-1)} = (\mathbf{D}_x^{D-1} h_i),
\]

so we obtain the matrix equation

\[
C_h^{(D-1)} = \sum_{i=1}^{k} \lambda_i C_{h_i}^{(D-1)} = (a_x L + a'_x L').
\]

The matrix on the right-hand side clearly has rank 2 as its rows are linear combinations of two vectors, so in the notation of Section 2.2, the vector subspace
Input. \( m \) homogeneous independent quadratic equations in \( n + 1 \) variables.

1. Generate the \( m \binom{D-2+n}{D-2} \) possible polynomials of degree \( D \) that are formed by multiplying each of the polynomials of the original system by some monomial of degree \( D - 2 \). The degree \( D \) is required to be less than the characteristic of the finite field.
2. Find a basis \( S \) of the linear span of all the polynomials generated by the first step.
3. Calculate the matrix \( C_{D-1}^f \) of \( (D-1) \)th partial derivatives for each polynomial \( f \in S \).
4. Find a linear combination of these partial derivative matrices \( C_{D-1}^f \) which has rank 2 (or lower) by considering the 3-minors or some other method.
5. Note that this it is not always possible to find such a linear combination, and in this case the GeometricXL algorithm fails for degree \( D \).
6. Using this linear combination, construct a polynomial in the linear span of \( S \) that is known to have factors, and then factorise this polynomial. This potentially allows the elimination of a variable from the original system of equations.
7. This process is repeated on the new smaller system and so on, potentially eliminating further variables.
8. Substitution is used to find values for the eliminated variables.

Output. Solution set for the original equation system (if method is successful).

Fig. 3. Basic Description of the GeometricXL Algorithm for a Quadratic System

We term this process the GeometricXL algorithm. The GeometricXL algorithm is a geometrically invariant generalisation of the ProjectiveXL algorithm. Having generated the polynomials of degree \( D \), we then analyse their partial derivatives matrices to try to determine a solution to the original equation system. We give a fuller description of the GeometricXL algorithm in Figure 3 and a simple illustration in Example 10. Furthermore, the GeometricXL algorithm may still be applicable even if the original condition that \( D < p \) is not true. In this case, a factorisation of the above type still gives rise to a linear combination of partial derivative matrices with rank 2, though a linear combination of partial derivative matrices with rank 2 does not necessarily correspond to a factorisation of that type.

Example 10. We consider the homogeneous quadratic polynomials in three variables over GF(37) of Example 9 given by

\[
\begin{align*}
  h_1 &= 6x_0^2 + 2x_0x_1 + 3x_0x_2 + x_1^2 + 16x_1x_2 + 3x_2^2 \\
  h_2 &= 18x_0^2 + 35x_0x_1 + 15x_0x_2 + 26x_1^2 + 12x_1x_2 + x_2^2.
\end{align*}
\]
The matrix of the linear combination of partial derivatives is thus given by
\[
\begin{pmatrix}
\lambda_1 D_{x_1} h_1 + \lambda_2 D_{x_2} h_2 \\
\lambda_1 D_{x_1} h_1 + \lambda_3 D_{x_2} h_2 \\
\lambda_1 D_{x_2} h_1 + \lambda_2 D_{x_2} h_2
\end{pmatrix}
= \begin{pmatrix}
12\lambda_1 + 36\lambda_2 & 2\lambda_1 + 35\lambda_2 & 3\lambda_1 + 15\lambda_2 \\
2\lambda_1 + 35\lambda_2 & 2\lambda_1 + 15\lambda_2 + 16\lambda_2 & 12\lambda_2 \\
3\lambda_1 + 15\lambda_2 & 16\lambda_1 + 12\lambda_2 & 6\lambda_1 + 2\lambda_2
\end{pmatrix}.
\]

This matrix has rank 2, so on taking its determinant, we obtain
\[0 = 34\lambda_1^3 + 28\lambda_1^2\lambda_2 + 23\lambda_1\lambda_2^2 + 7\lambda_2^3 = 34(\lambda_1 - 10\lambda_2)(\lambda_1 - 28\lambda_2)(\lambda_1 - 33\lambda_2),\]
so \(\lambda_1 = 10\lambda_2, \lambda_1 = 28\lambda_2\) or \(\lambda_1 = 33\lambda_2\). We thus obtain the following polynomials in the linear span of \(h_1\) and \(h_2\),
\[
\begin{align*}
10h_1 + h_2 &= 4x_0^2 + 18x_0x_1 + 8x_0x_2 + 36x_1^2 + 24x_1x_2 + 31x_2^2 \\
28h_1 + h_2 &= x_0^2 + 17x_0x_1 + 25x_0x_2 + 17x_1^2 + 16x_1x_2 + 11x_2^2 \\
33h_1 + h_2 &= 31x_0^2 + 27x_0x_1 + 3x_0x_2 + 33x_1^2 + 33x_1x_2 + 26x_2^2.
\end{align*}
\]

We have given all three for completeness, even though these three polynomials are necessarily linearly dependent. Each of these polynomials factorises, so we have
\[
\begin{align*}
10h_1 + h_2 &= (8x_1 + 25x_2)(x_0 + 15x_1 + 14x_2) \\
28h_1 + h_2 &= (x_0 + 24x_1 + 16x_2)(x_0 + 30x_1 + 36x_2) \\
33h_1 + h_2 &= 31(x_0 + 25x_1 + 15x_2)(x_0 + 26x_1 + 3x_2).
\end{align*}
\]

We can now substitute \(x_0 = -(8x_1 + 25x_2)\) into \(h_1\) (for example) to obtain
\[36x_1^2 + 11x_1x_2 + 15x_2^2 = 36(x_1 - 18x_2)(x_1 - 30x_2)\]
Taking the first factor, we have \(x_1 = 18x_2\) so \(x_0 = -(8x_1 + 25x_2) = 16x_2\), which gives \(\langle 16, 18, 1 \rangle^T\) as a solution. This is the image of the solution \(\langle 1, 27, 6 \rangle^T\) of Example 8 under the matrix of Example 9. We can calculate all four solutions similarly to obtain
\[\langle x_0, x_1, x_2 \rangle^T \in \{\langle 1, 8, 31 \rangle^T, \langle 1, 14, 14 \rangle^T, \langle 1, 15, 7 \rangle^T, \langle 1, 32, 6 \rangle^T\}.\]
These are the images of the solutions of Example 8 under the matrix of Example 9.

In general, computing the \((D - 1)\)th partial derivatives in terms of the \(\lambda_1, \ldots, \lambda_k\) in a successful application of the GeometricXL algorithm yields a linear system in \(\lambda_1, \ldots, \lambda_k\) of rank 2. However, the matrix of this linear system has rank two if and only all its 3-minors vanish. Thus evaluating all the 3-minors of this partial derivatives matrix gives a homogeneous cubic equation system in \(\lambda_1, \ldots, \lambda_k\). If we can find any solution of this cubic system by any method, then we can obtain a factorisation of the above type for some polynomial in the linear span of \(h_1, \ldots, h_k\).

The most obvious method to try to solve this cubic system is the Linearisation algorithm. There are \(\binom{n+D-1}{D-1}\) monomials in \((n+1)\) variables of degree \((D-1)\), so the partial derivatives matrix is an \(\binom{n+D-1}{D-1}\times(n+1)\) matrix. There are \(\binom{n+1}{3}\)
3-minors of an \( l \times (n+1) \) matrix, where in this case \( l = \binom{n+D-1}{D-1} \sim \frac{n^{D-1}}{(D-1)!} \) for large \( n \). Thus for an equation system with many variables (large \( n \)), the GeometricXL algorithm gives a homogeneous cubic system containing about \( \frac{1}{6} \binom{n^D}{D-1} \) cubic equations in \( k \) variables \( \lambda_1, \ldots, \lambda_k \), that is about \( \frac{k}{6}k^3 \) cubic monomials. Thus if \( k < \frac{n^D}{D-1} \), it may be possible to find a solution by linearisation, and hence a factorisation that may allow us to eliminate a variable from the original equation system. Furthermore, if we have vastly more cubic equations than cubic monomials, we may be able to analyse the system much more efficiently by only selecting a random subset of cubic equations for linearisation and still have reasonable confidence in our solution.

The GeometricXL algorithm is considerably more efficient than either XL-type algorithms or Gröbner basis techniques for certain equation systems, and an example of such an equation system is given in Example 11. This example illustrates the method of the GeometricXL algorithm in generating a succession of cubic systems using the 3-minors of partial derivatives matrices and then solving these cubic systems in order to find solutions to the original equation system.

**Example 11.** We give five homogeneous quartic polynomials \( f_1, f_2, f_3, f_4, f_5 \) in five variables over GF(37) in Appendix A. The Appendix then describes how to find the unique (projective) solution for the system \( f_1 = f_2 = f_3 = f_4 = f_5 = 0 \) using the GeometricXL algorithm. The solution method does not require the generation of any higher degree polynomials, so \( D = 4 \).

For comparison, we also calculated the unique solution of the system of Appendix A using both Gröbner basis techniques (including F4 [14]) and traditional XL algorithms. Calculation of this solution using Gröbner basis techniques with either lexicographic or graded reverse lexicographic monomial orderings typically requires the generation of polynomials of degree \( D = 14 \). Similarly, solving this equation system using the AffineXL or ProjectiveXL algorithm typically requires the generation of polynomials of degree \( D = 14 \). In a typical example of the ProjectiveXL algorithm, the final stage is the row reduction of a 5005\( \times \)3060 matrix of rank 3055 to give a quintic bivariate equation, which can then be solved.

\[ \square \]

### 7.2 Geometric Analysis of the GeometricXL Algorithm

We have seen that the GeometricXL algorithm works by constructing a polynomial \( h \in \langle h_1, \ldots, h_k \rangle \) such that \( h \in \mathbb{F}[L, L'] \), that is \( h \) is a polynomial in two linear polynomials \( L \) and \( L' \). We construct such a polynomial of degree \( D \) by finding a polynomial \( h \) for which the rank of the partial derivatives matrix \( C_h^{D-1} \) has rank 2. A basis for the row space of \( C_h^{D-1} \) then gives \( L \) and \( L' \). This is the situation (for rank 2) discussed by Proposition 1 of [4].

Geometrically, the constructed polynomial \( h \) of degree \( D \) is an element of the projective geometry of the \( D^{th} \) symmetric power of the dual space \( \mathbb{F} \langle S^D (V^*) \rangle \).
This projective geometry contains the degree $D$ Veronese variety
\[ V_v^{(D)} = \varphi_v^{(D)}(\mathbb{P}(V^*)) . \]

In the case that $D < p$, the positive characteristic of $\mathbb{F}$, the polynomial $h$ is in this Veronese variety $V_v^{(D)}$ if and only if $h = \lambda L^D$ for some linear polynomial $L$ and $\lambda \in \mathbb{F}$ (Section 3.4). An equivalent condition is that its partial derivatives matrix $C_h^{(D-1)}$ has rank 1. Geometrical aspects of this situation are discussed in [26]. Thus we could define a rank-one version of GeometricXL in which we find a partial derivatives matrix $C_h^{(D-1)}$ of rank 1. In certain situations, this can give a very efficient algorithm, as illustrated in Example 12.

**Example 12.** Consider the equation system over GF(37) given by the first four homogenised equations of Example 2, namely
\[
\begin{align*}
0 &= f_1 = x_0^2 + x_0 x_1 + x_0 x_2 - x_1 x_2 \\
0 &= f_2 = 2x_0^2 + x_0 x_2 + x_1^2 - x_2^2 \\
0 &= f_3 = x_0 x_1 + x_0 x_2 - 2x_1^2 + 2x_1 x_2 - x_2^2 \\
0 &= f_4 = 3x_0^2 + x_0 x_1 + 9x_0 x_2 + 8x_1^2 + 18x_1 x_2 + 22x_2^2 .
\end{align*}
\]

By calculating the partial derivatives matrix $\sum_{i=1}^4 \lambda_i C_{f_i}$ and evaluating its 2-minors, we can find two linear combinations of partial derivatives matrices having rank 1. We thus obtain
\[
\begin{align*}
f_1 + 11f_2 + 6f_3 + 20f_4 &= 9(x_0 + 20x_1 + 11x_2)^2 = 0 \\
\text{and} \quad f_1 + 29f_2 + 20f_3 + 7f_4 &= 6(x_0 + 27x_1 + 31x_2)^2 = 0 ,
\end{align*}
\]
from which we can easily deduce that $x_1 = 2x_0$ and $x_2 = 3x_0$. We note that there is no linear combination of the first three equations that has a similar factorisation as a square. Thus rank-one GeometricXL cannot be applied to the equation system $f_1 = f_2 = f_3 = 0$. \hfill \Box

We are primarily interested in the GeometricXL algorithm in the situation where the partial derivatives matrix $C_h^{(D-1)}$ has rank 2. However, any matrix of rank 2 can be written as the sum of two matrices of rank 1, but a partial derivatives matrix of rank 1 indicates a point in the Veronese variety $V_v^{(D)}$. We can therefore show that any polynomial $h$ of degree $D$ has a partial derivatives matrix $C_h^{(D-1)}$ of rank 2 if and only if $h$ is on a line joining some pair of points in the Veronese variety $V_v^{(D)}$, that is $h$ lies on a chord or secant of the Veronese variety (Section 2.3). We denote the chordal or secant variety of the Veronese variety $V_v^{(D)}$, that is the set of all points in $\mathbb{P}(S^D(V^*))$ on some chord of $V_v^{(D)}$, by $S_v^{(D)}$. Geometrical properties of the secant variety of the Veronese variety are extensively discussed in [19, 20].

The natural geometrical interpretation of the GeometricXL algorithm is that it is a method that attempts to calculate the intersection of the variety $V(h_1, \ldots, h_k)$ generated by the polynomials $h_1, \ldots, h_k$ of degree $D$ with the secant variety $S_v^{(D)}$. 
The algebraic interpretation of the GeometricXL algorithm or any XL-type algorithm, is that it is a method that attempts to find a linear combination of a collection of matrices that has rank 2, a problem sometimes termed MinRank.

Certain other XL-type algorithms can now be seen geometrically as special cases of the GeometricXL algorithm. The rank-one GeometricXL algorithm of Example 12 is the special case when this intersection contains a point of the Veronese variety itself. When the Linearisation algorithm works, it would typically produce a polynomial of the form \( x_0 x_1^{D-1} + \lambda x_0^D = x_0^{D-1}(x_1 + \lambda x_0) \). Polynomials of this type form a subset of the secant variety \( S_{V}(D) \). Thus the Linearisation algorithm can typically be viewed as a special case of the GeometricXL algorithm in which we are constrained to take the intersection of the polynomial variety \( V(h_1, \ldots, h_k) \) with a subset of the secant variety of the Veronese variety.

The AffineXL and ProjectiveXL algorithms (Section 6.1 and 6.2) can also be considered special cases of the GeometricXL algorithm in which we are constrained to take the intersection of the polynomial variety \( V(h_1, \ldots, h_k) \) with particular subsets of the secant variety \( S_{V}(D) \). In the ProjectiveXL algorithm, this subset is defined by the hyperplanes \( x_i = 0 \) and \( x_j = 0 \), whereas in the AffineXL algorithm we are constrained to take the hyperplanes \( x_i = 0 \) and \( x_0 = 0 \). We illustrate this in Example 13.

**Example 13.** Suppose \( V \) is a vector space of dimension 3 over \( \mathbb{F} \). We consider the degree 3 Veronese embedding \( \varphi_{V}^{(3)}: \mathbb{P}(V) \to \mathbb{P}(S^3(V)) \). An element of the pencil defined by \( x_0 = 0 \) and \( x_1 = 0 \) is defined by \( x_0 + \theta x_1 = 0 \) for some \( \theta \in \mathbb{F} \cup \{ \infty \} \) (with the usual interpretation of \( \infty \)). The Veronese embedding of such an element of the pencil is defined by \( (1, \theta, 0, \theta^2, 0, 0, \theta^3, 0, 0, 0, 0) \). The set of such Veronese embeddings forms a normal rational curve, in this case a twisted cubic, in the subspace defined by equations \( w_{002} = w_{012} = w_{022} = w_{112} = w_{122} = w_{222} = 0 \), and these points span this space.

\[ \square \]

### 7.3 The GeometricXL Algorithm and the Relinearisation Algorithm

The Relinearisation algorithm can also be viewed in some sense as a special case of the AffineXL algorithm [10] and hence of the GeometricXL algorithm. However, the relationship between these algorithms is geometrically more complicated than the other special cases we have considered. We discuss this by considering the application of the GeometricXL algorithm and Relinearisation algorithm to a quadratic system that produces degree 4 equations.

During the degree 4 version of the GeometricXL algorithm, the points of \( \mathbb{P}(V) \) are mapped to points on a variety \( V_{V}^{(4)} \) in \( \mathbb{P}(S^4(V)) \), with generic quadrics being mapped to varieties of dimension \( n - 1 \) and order 8 that are the intersection of \( V_{V}^{(4)} \) with subspaces of \( \mathbb{P}(S^4(V)) \) of dimension \( N_4 - 1 - N_2 = N_4 - N - 1 \), where \( N_4 \) and \( N = N_2 \) are the dimensions of \( \mathbb{P}(S^4(V)) \) and \( \mathbb{P}(S^3(V)) \) respectively (Section 2.3). In relinearizing the same original system, the points are initially mapped to the Veronese variety \( V \subset \mathbb{P}(S^2(V)) \), and the equations become hyperplanes in that space. If we were to apply the Veronese embedding \( \varphi_{S^2(V)} \)
to the points of $\mathbb{P}(S^2(V))$, then they would be mapped to points on a larger Veronese variety $V_{S^2(V)}$ in the projective geometry $\mathbb{P}(S^2(S^2(V)))$ of dimension

$$N' = \frac{1}{8} n(n+3)(n^2 + 3n + 6) > N_4.$$  

However, the Veronese variety $V_V \subset \mathbb{P}(S^2(V))$ is contained in $\frac{1}{12} n(n+1)(n+2)$ linearly independent quadrics, which are mapped to linearly independent hyperplanes in $\mathbb{P}(S^2(S^2(V)))$. These hyperplanes intersect in a subspace of dimension $N_4$, and this subspace intersects the Veronese variety $V_{S^2(V)}$ in precisely the variety $V_{(4)}$ obtained by a degree 4 version of the GeometricXL algorithm. This can be seen by considering the fact that the quadrics in question have equations of the form $y_{ii}y_{jj} - y_{ij}^2 = 0$, $y_{ij}y_{jk} - y_{ii}y_{jk} = 0$ or $y_{ij}y_{kl} - y_{il}y_{kj} = 0$, and observing that they are mapped into hyperplanes with equations $z_{(ij)(jk)} = z_{(ii)(jk)}$ and so on, so the points contained in the intersection of all these hyperplanes have the same coordinates as those arising from degree 4 XL, but with some repeated.

Both the Relinearisation algorithm and the GeometricXL algorithm have the problem that they may consider polynomials that are not independent. In the Relinearisation algorithm, this can occur when restricting the Veronese equations to a subspace; whereas in the GeometricXL algorithm this can occur when generating higher degree equations. This is fundamentally the same problem in two different guises. However, in the case where the original equation system has a unique solution over the given field, then if (the possibly repeated application of) relinearisation succeeds in finding this solution, then carrying out an XL procedure of the corresponding degree also finds this solution without having to carry out the latter stages of the XL procedure.

7.4 Properties of the GeometricXL Algorithm

We have seen that the first stages of the GeometricXL algorithm can be interpreted as a search for points on the secant variety $S_{V^*}^{(D)}$, of the Veronese variety $V_{V^*}^{(D)}$, and that there is correspondence of this secant variety with a set of matrices of rank 2. Thus the points of this secant variety can be described by a set of cubic equations which are given by the 3-minors of these matrices. In order to formally specify the GeometricXL algorithm as a well-defined algorithm, it would be necessary to provide an algorithm for finding points on this variety. Unfortunately, this is likely to be difficult in general as there is no efficient method for solving a general system of cubic equations.

We therefore consider some more specialised algorithms. Suppose $W_D$ denotes the subspace of $\mathbb{P}(S^D(V^*))$ spanned by all the polynomials of degree $D$ generated by an XL-type process. Given a projective space $\Sigma$ contained in $S_{V^*}^{(D)}$, we can compute the subspace $W_D \cap \Sigma$ very efficiently using linear algebra. There are particular subspaces $\Sigma$ of the secant variety $S_{V^*}^{(D)}$ for which there are well established methods for finding points on the subspace. By choosing such a subspace, we can produce an efficient version of an XL-type algorithm.
We can regard the projective geometry $\mathbb{P}^{\left(\mathbb{V}^D\right)}$ as the space of all homogeneous polynomials of degree $D$. For a polynomial $h$ in the Veronese variety $\mathbb{V}^D$, we denote the tangent space to $\mathbb{V}^D$ at $h$ by $T_h(\mathbb{V}^D)$. This tangent space has dimension $n$ and is contained in the secant variety $\mathbb{S}^{\left(\mathbb{V}^D\right)}$. For example, the tangent space at the polynomial $x^D_0$ is given by

$$T_{x^D_0}(\mathbb{V}^D) = \{ \langle Lx^{D-1}_0 \rangle | L \text{ is a linear polynomial} \}.$$

If our homogeneous equation system is derived from some original non-homogeneous system, then we may not actually be interested in solutions with $x_0 = 0$, that is solutions lying in the “hyperplane at infinity”. In this case, if the space $\mathcal{W}_D$ of generated polynomials of degree $D$ contains $\langle Lx^{D-1}_0 \rangle$, then we can immediately deduce that any solutions of the original nonhomogeneous system lie in the hyperplane with equation $L = 0$. This essentially eliminates a variable from the system.

To determine whether $\mathcal{W}_D$ contains such a polynomial, we have only to calculate its intersection with $T_{x^D_0}(\mathbb{V}^D)$. If this intersection $\mathcal{W}_D \cap T_{x^D_0}(\mathbb{V}^D)$ has dimension $r > 0$, then we can find a space of dimension $n - r$ containing the solution, and the process can be repeated on the smaller system. There is a sense in which this process can be thought of a geometrically invariant version of the Linearisation algorithm in which a co-ordinate specific linear polynomial $x_i - x_0$ is replaced by an arbitrary linear polynomial. We note that this procedure is essentially equivalent to the method called ElimLin of [9], where it is derived in the context of considering the application of a SAT-solver to cryptology.

This general technique cannot be applied in the case when $\mathcal{W}_D \cap T_{x^D_0}(\mathbb{V}^D) = \emptyset$. It is then necessary to consider methods for choosing the smallest possible value of $D$ that enables this intersection to be non-empty. We restrict our attention now to a system of equations that has a single solution over the algebraic closure of a field $\mathbb{F}$, so as to increase the likelihood of this intersection being non-empty. A sufficient condition for the intersection of $\mathcal{W}_D$ and $T_{x^D_0}(\mathbb{V}^D)$ to be non-empty is for the dimension of $\mathcal{W}_D$ to be greater than or equal to $N_D - n$. The consideration of Hilbert series in [13] suggests that if the system of equations consists of $n + 1$ quadrics then the degree $d$ must be at least $n + 1$ for this to occur. However, for a generic system of $n + 1$ quadrics with an empty intersection, the dimension of $\mathcal{W}_D$ is $N_D - 1$. This suggests that it might be advantageous to seek a $D$ such that $\mathcal{W}_D \cap T_{x^D_0}(\mathbb{V}^D)$ has dimension $n - 1$, which occurs if the dimension of $\mathcal{W}_D$ is $N - 1$. This makes it possible to find $n$ hyperplanes whose (affine) intersection determines the solution precisely. However, if for some $D$ the dimension of $\mathcal{W}_D$ is $N_D - 1$, then linearisation of $\mathcal{W}_D$ directly yields the solution.

7.5 Problems with the GeometricXL Algorithm

An XL-type algorithm, including the GeometricXL algorithm, aims to produce a polynomial which can potentially be factored into many linear factors. However,
we usually have no \textit{a priori} method of determining which linear factor pertains to the true solution, and we may have to test each linear factor in turn. We would usually test each linear factor by using it to make a substitution and then applying the same technique to the smaller system. However, each of these smaller systems could give rise to a number of linear factors, only one of which pertains to the true solution, and so on. It is thus possible, in principle, that for a large enough $D$ such a proliferation of linear factors could lead to more possibilities than can be efficiently checked. In this case, a useful heuristic approach would seem to be to increase the degree $D$, which should generally greatly lower the number of linear factors.

8 Conclusions

We have given an extensive discussion of the geometrical properties of the XL-type algorithms for finding the solution to a multivariate equation system and put these algorithms on a firm geometrical footing. In particular, we have shown how XL-type algorithms are different techniques for finding points on the intersection of some subspace determined by the equations with the secant variety of the Veronese variety of some degree $D$. The different XL-type techniques which have been proposed are essentially those obtained by considering some subset of this secant variety rather than the full secant variety. The new method of this paper, the GeometricXL algorithm, generalises the previous methods by considering the full secant variety. As we demonstrated in Example 11, the GeometricXL algorithm can be considerably more efficient in some cases then either a standard XL algorithm or a Gröbner basis algorithm.

There are a number of obvious areas for future research. Firstly, the GeometricXL algorithm requires us to find a linear combination of a collection of matrices having rank 2. We can do this by considering the 3-minors of these matrices to obtain a cubic equation system, which we may be able to solve. However, it may be that there is a more efficient way in some cases of finding such a linear combination of matrices having rank 2. Secondly, the reducible linear combinations of polynomials produced by the GeometricXL algorithm are of a very particular form. Ideally, we would like some efficient method of determining in many cases when a linear combination of polynomials is reducible. Finally, the GeometricXL algorithm as described in Figure 3 is only generally applicable when the positive characteristic of the field is not too small. However, the fundamental geometric results we have been discussing are true in any characteristic [4, 19, 20]. In particular, a point on the secant variety of the Veronese variety corresponds to a factorisation of a homogeneous polynomial to give $\prod (\theta_j L - \theta'_j L')$ (Section 7.1). Furthermore, this secant variety is defined by a set of cubic polynomials ([19] Theorem 1.56). Thus it may be possible to construct an algorithm to find a solution to a multivariate equation system by finding the intersection of the span of this system with the secant variety of the Veronese variety. Such an algorithm would work over a field of any characteristic.
Acknowledgements

We would like to thank the anonymous referees for their comments on the paper.

References

A  Using the GeometricXL Algorithm to solve Example 11

We specify the five quartic polynomials $f_1, \ldots, f_5$ of Example 11 below. These are homogeneous quartic polynomials $f_1, f_2, f_3, f_4, f_5$ in five variables over GF(37). We describe how to solve this equation system using the GeometricXL algorithm with $D = 4$ to systematically eliminate variables from the system.

Five Variables

The coefficients of these polynomials $f_i$ with respect to lexicographic monomial ordering $x_0^4, x_0 x_1, \ldots, x_3 x_4^2, x_4^4$ are given below.

\[
\begin{array}{cccccccccccc}
16 & 30 & 32 & 36 & 13 & 11 & 0 & 0 & 36 & 28 & 12 & 5 & 15 & 29 \\
4 & 19 & 12 & 2 & 12 & 9 & 28 & 2 & 27 & 33 & 8 & 13 & 22 & 17 \\
27 & 20 & 17 & 27 & 5 & 28 & 32 & 2 & 29 & 3 & 2 & 15 & 5 \\
17 & 17 & 13 & 22 & 16 & 9 & 4 & 29 & 13 & 8 & 10 & 5 & 33 & 27 \\
27 & 34 & 32 & 32 & 0 & 0 & 21 & 2 & 31 & 12 & 33 & 11 & 17 & 9 \\
22 & 2 & 17 & 7 & 24 & 5 & 25 & 13 & 32 & 31 & 28 & 19 & 24 & 22 \\
36 & 6 & 5 & 13 & 33 & 9 & 28 & 30 & 0 & 16 & 9 & 9 & 4 & 5 \\
22 & 31 & 29 & 5 & 17 & 34 & 16 & 16 & 15 & 7 & 35 & 2 & 27 & 2 \\
23 & 10 & 15 & 25 & 6 & 31 & 0 & 26 & 13 & 18 & 1 & 2 & 23 & 8 \\
22 & 7 & 20 & 32 & 36 & 2 & 30 & 24 & 24 & 19 & 35 & 9 & 35 & 12 \\
36 & 24 & 12 & 27 & 7 & 35 & 19 & 6 & 6 & 1 & 20 & 27 & 36 & 10 \\
11 & 30 & 1 & 33 & 17 & 8 & 35 & 27 & 11 & 18 & 13 & 36 & 29 & 13 \\
23 & 6 & 12 & 31 & 28 & 9 & 26 & 23 & 27 & 34 & 9 & 36 & 20 & 5 \\
32 & 5 & 14 & 24 & 34 & 20 & 20 & 17 & 0 & 30 & 2 & 25 & 2 & 4 \\
36 & 30 & 28 & 35 & 1 & 35 & 9 & 7 & 16 & 28 & 29 & 23 & 24 & 35 \\
19 & 21 & 33 & 28 & 24 & 32 & 15 & 6 & 36 & 18 & 15 & 26 & 11 & 1 \\
18 & 33 & 17 & 10 & 8 & 4 & 21 & 3 & 1 & 4 & 13 & 29 & 10 & 13 \\
24 & 4 & 23 & 10 & 8 & 10 & 36 & 6 & 19 & 5 & 26 & 2 & 36 & 28 \\
11 & 20 & 27 & 24 & 25 & 10 & 8 & 24 & 2 & 31 & 0 & 34 & 20 & 36 \\
25 & 11 & 30 & 32 & 22 & 7 & 26 & 26 & 32 & 17 & 11 & 3 & 20 & 23 \\
3 & 8 & 1 & 18 & 23 & 35 & 34 & 3 & 7 & 7 & 32 & 22 & 23 & 17 \\
32 & 4 & 5 & 33 & 4 & 22 & 25 & 21 & 31 & 7 & 22 & 0 & 17 & 27 \\
35 & 6 & 4 & 2 & 6 & 23 & 10 & 19 & 0 & 4 & 11 & 33 & 10 & 6 \\
1 & 36 & 32 & 36 & 32 & 25 & 33 & 7 & 25 & 10 & 7 & 1 & 26 & 25 \\
\end{array}
\]

We apply the GeometricXL algorithm to this equation system. Thus we need to find $\lambda_1, \ldots, \lambda_5$ such that

$$
\lambda_1 C_{f_1}^{(3)} + \lambda_2 C_{f_2}^{(3)} + \lambda_3 C_{f_3}^{(3)} + \lambda_4 C_{f_4}^{(3)} + \lambda_5 C_{f_5}^{(3)}
$$

has rank 2, where $C_{f_i}^{(3)}$ is the matrix of third partial derivatives for each polynomial $f_i$. There are 35 monomials of degree 3, so the matrices $C_{f_i}$ are $35 \times 5$ matrices. We give the transpose of each of these matrices $C_{f_i}^{(3)}$ below, where each row has 35 entries and is written below across two rows.
We now consider the 3-minors (3×3 sub-determinants) of the matrix $\sum_{i=1}^{5} \lambda_i C_f$, as polynomials in $\lambda_1, \lambda_2, \lambda_3, \lambda_4, \lambda_5$. There are $65450$ 3-minors of a $5\times35$ matrix, so we obtain 65450 homogeneous cubic equations in $\lambda_1, \lambda_2, \lambda_3, \lambda_4, \lambda_5$. We give below as an example the coefficients of the “upper left” such minor with respect to the lexicographical ordering $\lambda_1^3, \lambda_1^2 \lambda_2, \ldots, \lambda_4 \lambda_5^2, \lambda_5^3$.

As there are only 35 cubic monomials in $\lambda_1, \ldots, \lambda_5$, this cubic system clearly has the potential for solution by linearisation (Section 4.1), and the linearisation
A matrix is a $65450 \times 35$ matrix. This matrix has rank 34 and the first 34 rows of the echelon form are the matrix $(I_{34}|v)$, where the components of the vector $v$ of length 34 are given below.

By considering the appropriate components of $v$, we obtain

$$0 = (\lambda_1 \lambda_5^2 + 24\lambda_5^3) = (\lambda_2 \lambda_5^2 + 25\lambda_5^3) = (\lambda_3 \lambda_5^2 + 14\lambda_5^3) = (\lambda_4 \lambda_5^2 + 16\lambda_5^3).$$

As $\lambda_5 = 0$ would give a matrix of rank 0, we obtain

$$\lambda_1 = 13\lambda_5, \quad \lambda_2 = 12\lambda_5, \quad \lambda_3 = 23\lambda_5 \text{ and } \lambda_4 = 21\lambda_5.$$

We can now construct the polynomial $$g = 13f_1 + 12f_2 + 23f_3 + 21f_4 + f_5,$$

The coefficients of this polynomial with respect to the lexicographic monomial ordering $x_0^4, x_0^3x_1, \ldots, x_3x_4^3, x_4^4$ are given by the array below.

The 35×5 matrix $C_g^{(3)} = 13C^{(3)}_f + 12C^{(3)}_f + 23C^{(3)}_f + 21C^{(3)}_f + C^{(3)}_f$ of third partial derivatives can be used to find the factorisation of $g$. The transpose of $C_g^{(3)}$ is given by the array below.

This matrix $C_g^{(3)}$ has rank 2 (by construction) and any row is a linear combination of the rows $(1, 0, 7, 24, 12)$ and $(0, 1, 28, 21, 12)$. Thus the linear factors of $g$ are a linear combination of $x_0 + 7x_2 + 24x_3 + 12x_4$ and $x_1 + 28x_2 + 21x_3 + 12x_4$.

The factorisation of $g$ is Quadratic $\times$ Linear$^2$, so in this case we need only find this unique linear factor of $g$. If $g$ had more distinct linear factors, then we would be in the situation discussed in Section 7.4. We can now factorise $g$ by a small search through all the possible such linear combinations or by some other method to find that the only linear factor of $g$ is

$$x_4 + 21x_3 + 31x_2 + 6x_1 + 28x_0.$$
original equation system. This gives a new equation system $f_1' = f_2' = f_3' = f_4' = 0$ of four independent quartic equations in the four variables $x_0, x_1, x_2, x_3$. The coefficients of these polynomials with respect to lexicographic monomial ordering are given below.

$$
\begin{align*}
35 & 13 & 21 & 26 & 13 & 10 & 0 & 33 & 15 & 23 & 5 & 13 & 13 & 8 & 2 & 34 & 5 & 28 \\
\end{align*}
$$

We now apply the GeometricXL algorithm to this new equation system. The matrices $C^{(3)}_{f_i}$ of third partial derivatives for each polynomial $f_i'$ are $20 \times 4$ matrices, and we give the transpose $C^{T}_{f_i}$ of each of these matrices below.

$$
\begin{align*}
26 & 4 & 15 & 8 & 15 & 20 & 0 & 21 & 30 & 18 & 30 & 26 & 16 & 2 & 31 & 30 & 19 & 8 & 3 \\
8 & 0 & 30 & 18 & 26 & 2 & 31 & 19 & 8 & 3 & 18 & 19 & 28 & 30 & 19 & 5 & 3 & 24 & 4 \\
3 & 7 & 23 & 30 & 29 & 22 & 36 & 8 & 9 & 26 & 36 & 19 & 3 & 16 & 0 & 18 & 13 & 21 & 9 & 34 \\
7 & 29 & 22 & 36 & 36 & 19 & 3 & 16 & 0 & 18 & 27 & 27 & 5 & 20 & 17 & 29 & 32 & 5 & 30 & 17 \\
30 & 36 & 9 & 26 & 3 & 0 & 18 & 21 & 9 & 34 & 5 & 17 & 29 & 5 & 30 & 17 & 4 & 27 & 14 \\
15 & 31 & 1 & 12 & 21 & 28 & 30 & 34 & 18 & 20 & 33 & 2 & 1 & 19 & 0 & 16 & 14 & 23 & 27 & 30 \\
\end{align*}
$$

As before, we need to find a linear combination of these matrices with rank 2, so we consider the 3-minors of the matrix $\sum_{i=1}^{4} \lambda_i C_{f_i}$. There are 4560 3-minors of a $20 \times 4$ matrix, so we obtain 4560 homogeneous cubic equations in the 20 cubic monomials in $\lambda_1, \lambda_2, \lambda_3, \lambda_4$. The 4560$\times$20 linearisation matrix for this cubic system in $\lambda_i$ has rank 19, and the first 19 rows of the echelon form are the matrix $(I_{19} | v')$, where the vector $v'$ is of length 19 with components given below.

$$
\begin{align*}
23 & 34 & 30 & 12 & 2 & 17 & 29 & 15 & 6 & 32 & 11 & 1 & 30 & 27 & 33 & 28 & 26 & 3 & 16 \\
\end{align*}
$$

By considering the appropriate components of $v$, we obtain

$$
0 = (\lambda_1 \lambda_4^2 + 32 \lambda_4^2) = (\lambda_2 \lambda_4^2 + 28 \lambda_4^2) = (\lambda_3 \lambda_4^2 + 16 \lambda_4^2)
$$

As $\lambda_4 = 0$ would give a matrix of rank 0, we obtain

$$
\lambda_1 = 5 \lambda_4, \quad \lambda_2 = 9 \lambda_4 \quad \text{and} \quad \lambda_3 = 21 \lambda_4.
$$

We can now construct the polynomial $g' = 5f_1' + 9f_2' + 21f_3' + f_4'$. The coefficients of this polynomial with respect to the lexicographic monomial ordering
\[x_0^4, x_0^3 x_1, \ldots, x_2 x_3^3, x_3^4\] are given by the array below.

\[
\begin{array}{cccccccccc}
26 & 13 & 0 & 1 & 2 & 10 & 3 & 26 & 13 & 6 & 19 & 1 & 25 & 28 & 33 & 17 & 23 & 28 \\
8 & 20 & 27 & 12 & 0 & 27 & 20 & 4 & 2 & 0 & 32 & 11 & 15 & 5 & 3 & 8 & 20
\end{array}
\]

We calculate \(C^{(3)} g' = 5C^{(3)} f_1' + 9C^{(3)} f_2' + 21C^{(3)} f_3' + C^{(3)} f_4'\), the 20×4 matrix of third partial derivatives of \(g'\). Its transpose is given by the array below.

\[
\begin{array}{cccccccccccccccccc}
6 & 17 & 18 & 27 & 3 & 2 & 13 & 19 & 33 & 34 & 19 & 35 & 0 & 34 & 3 & 16 & 12 & 0 & 27 & 29 \\
32 & 18 & 30 & 26 & 2 & 19 & 33 & 27 & 19 & 16 & 34 & 3 & 12 & 0 & 27 & 27 & 30 & 12 & 11 & 11 & 36 \\
15 & 27 & 26 & 24 & 13 & 33 & 34 & 19 & 16 & 9 & 0 & 3 & 16 & 0 & 27 & 29 & 30 & 12 & 11 & 11 & 36
\end{array}
\]

The matrix \(C^{(3)} g'\) has rank 2, so any row of \(C^{(3)} g'\) is a linear combination of the two rows \((1, 0, 23, 24, 12)\) and \((0, 1, 6, 4, 12)\). Thus the linear factors of \(g'\) are a linear combination of \(x_3 + 32 x_2 + 21 x_1 + 11 x_0\).

**Three Variables**

We can now eliminate a second variable. The substitution \(x_3 = -(32 x_2 + 21 x_1 + 11 x_0)\) in the four variable equation system gives an equation system \(f_1'' = f_2'' = f_3'' = 0\) of three independent quartic equations in the three variables \(x_0, x_1, x_2\).

The coefficients of these polynomials \(f_1'', f_2'', f_3''\) with respect to lexicographic ordering are given by the array below.

\[
\begin{array}{cccccccccc}
31 & 30 & 35 & 11 & 0 & 33 & 23 & 22 & 6 & 22 & 8 & 7 & 6 & 6 & 36 \\
1 & 11 & 3 & 14 & 3 & 36 & 35 & 32 & 5 & 0 & 30 & 21 & 12 & 13 & 4 \\
19 & 15 & 30 & 8 & 0 & 9 & 14 & 13 & 29 & 6 & 5 & 27 & 3 & 28 & 0
\end{array}
\]

We give the transpose of the 10×3 matrices \(C^{(3)} f_i''\) of third partial derivatives for each polynomial \(f_i''\) below.

\[
\begin{array}{cccccccc}
4 & 32 & 25 & 7 & 0 & 21 & 27 & 7 & 12 & 21 \\
32 & 7 & 0 & 27 & 7 & 12 & 7 & 5 & 24 & 36 \\
25 & 0 & 21 & 7 & 12 & 21 & 5 & 24 & 36 & 13 \\
24 & 29 & 18 & 19 & 6 & 33 & 25 & 27 & 10 & 0 \\
29 & 19 & 6 & 25 & 27 & 10 & 17 & 15 & 11 & 4 \\
18 & 6 & 33 & 27 & 10 & 0 & 15 & 11 & 4 & 22 \\
12 & 16 & 32 & 32 & 0 & 36 & 10 & 26 & 21 & 36 \\
16 & 32 & 0 & 10 & 26 & 21 & 9 & 14 & 12 & 20 \\
32 & 0 & 36 & 26 & 21 & 36 & 14 & 12 & 20 & 0
\end{array}
\]

We consider the 3-minors of the matrix \(\sum_{i=1}^3 \lambda_i C f_i''\) to obtain 120 homogeneous cubic equations in the 10 cubic monomials in \(\lambda_1, \lambda_2, \lambda_3\). The 120×10 linearisation matrix for this system has rank 9, and the first 9 rows of the echelon form are the matrix \((I_9 | v'')\) where \(v''\) is a vector of length 9 with components given below.

\[
\begin{array}{ccccccccccc}
31 & 18 & 10 & 20 & 7 & 8 & 14 & 16 & 13
\end{array}
\]
We thus obtain the equations
\[
\begin{align*}
\lambda_1 \lambda_2^2 + 8 \lambda_3^3 & = \lambda_2 \lambda_3^2 + 13 \lambda_3^3 = 0, \quad \text{so} \quad \lambda_1 = 29 \lambda_3 \quad \text{and} \quad \lambda_2 = 24 \lambda_3,
\end{align*}
\]
as only nonzero solutions are permissible. We can now construct the polynomial
\[ g'' = 29f'''_1 + 24f'''_2 + f'''_3. \]
The coefficients of this polynomial with respect to the lexicographic monomial ordering are given below.

\[
\begin{array}{ccccccccccccc}
17 & 2 & 7 & 34 & 35 & 17 & 4 & 13 & 27 & 15 & 32 & 31 & 21 & 33 & 30
\end{array}
\]

The transpose of the matrix
\[
C g'' = 29C f'''_1 + 24C f'''_2 + C f'''_3
\]
of third partial derivatives is given by the array below.

\[
\begin{array}{ccccccccccccc}
\end{array}
\]

This matrix has rank 2 and is spanned by the rows (1, 0, 20) and (0, 1, 8), so the linear factors of \( g'' \) are linear combinations of \((x_0 + 20x_2)\) and \((x_1 + 8x_2)\). Thus we find that the only linear factor of \( g'' \) is
\[ x_2 + 27x_1 + 17x_0. \]

**Two Variables**

We can now make the substitution \( x_2 = -(17x_0 + 27x_1) \) to obtain the bivariate equation system
\[
\begin{align*}
f'''_1 & = 35x_0^4 + 25x_1^2x_0 + 5x_0^2x_1 + 31x_0x_1^3 + 8x_1^4 \\
& = (x_1 - 2x_0)(x_1 + 31x_0)(8x_1^2 + 36x_0x_1 + 31x_0^2)
\end{align*}
\]

and
\[
\begin{align*}
f'''_2 & = 5x_0^4 + 14x_0^2x_1 + 27x_1^2x_0 + 35x_0x_1^3 + 13x_1^4 \\
& = (x_1 - 2x_0)(13x_1^2 + 24x_1^2x_0 + x_0^2 + 16x_0^3)
\end{align*}
\]

Thus we can deduce that \( x_1 = 2x_0 \), and hence find the unique (projective) solution to the original equation system as
\[
(x_0, x_1, x_2, x_3, x_4)^T = \langle 1, 2, 3, 4, 5 \rangle^T.
\]