Extending Dixon's Resultant by Using Pseudo-Inverse Matrices and Its Applications to Geometric Reasoning *

Trân Quôc-Nam **

Research Institute for Symbolic Computation
RISC–Linz, Johannes Kepler University
A–4040 Linz, Austria
E-mail: tqnam@risc.uni-linz.ac.at

Abstract. Dixon's resultant method is an efficient way of simultaneously eliminating several variables from a system of nonhomogeneous polynomial equations at a time. However, the method only works for systems of $n+1$ generic n-degree polynomials in n variables and does not work for most algebraic and geometric problems. In this paper, by using techniques from pseudoinverse theory, we extend Dixon's resultant to an arbitrary system of $n+1$ nontrivial polynomials where the Dixon matrix can be singular or even nonsquare. Our method does not require any precondition. The extended method as an elimination method can be effectively used in geometric reasoning, computer aided geometric design (CAGD) and solid modeling.

1 Introduction

In applied mathematics such as geometric reasoning, CAGD and solid modeling, we often have to eliminate several variables from a system of equations (e.g. the problem of implicitization). Theoretically, the Gröbner bases method, Ritt's characteristic set method or successive Sylvester resultant computations can be used to achieve our demands. But they are inefficient and require a lot of memory space.

Meanwhile, Dixon's resultant is an efficient method to simultaneously eliminate several variables from a system of nonhomogeneous polynomial equations at a time. In contrast with successive resultant computation techniques which eliminate variables one by one and compute numerous intermediate resultants before finally computing the resultant of the whole set, Dixon's resultant creates a much smaller matrix and avoids such intermediate results, and hence is much faster and also saves memory space.

Unfortunately, Dixon's resultant does not work for most algebraic and geometric problems. Originally, Dixon's resultant was a method for obtaining the resultant of a set of three generic bidegree polynomials (see [Dix08]). This is a generalization of Cayley's formulation [Cay65] of Bezout's efficient method for computing the resultant of two univariate polynomials. Dixon had generalized his method to any $n+1$ generic n-degree polynomials in n variable – a condition which cannot be satisfied by most problems arising in practice. For most practical problems, the matrix set up in Dixon's method, the Dixon matrix, becomes singular or even nonsquare. As a consequence, Dixon's resultant vanishes identically, without providing any information

*

TALK AT THE WORKSHOP "AUTOMATED DEDUCTION IN GEOMETRY" IN SEPTEMBER 1996, TOULOUSE, FRANCE.

** Supported by the Austrian Science Foundation (FWF) project HySaX

about the common solutions of equations. There are some efforts to obtain nonzero conditions by perturbing the polynomial system (see [Chi90], [Can90]). However, these efforts either suffer from the facts that the method is not automatic, requires human expertise and enlarges the size of Dixon matrix with larger entries. Another approach of Kapur et al. (see [KSY94], [KS95]) is based on the idea of checking if there exists a submatrix of the Dixon matrix with one column less and strictly smaller rank than the Dixon matrix when the Dixon matrix is square and singular. However, this approach requires a precondition on the system of equations.

In this paper, by using techniques from pseudoinverse theory, we overcome the above obstacles and extend Dixon's resultant to an arbitrary system of $n+1$ nonzero polynomials where the Dixon matrix can be singular or even nonsquare. Our method does not require any precondition. Note that, this approach can also be used to extend other multivariate resultant methods such as Macaulay resultant or sparse resultant.

The structure of the paper is as follows. In the next section, we summarize some well-known results from Dixon's resultant method and the pseudoinverse theory. In Section 3, we present our main results. In Section 4, we present our main algorithm to compute extended Dixon's resultants. In Section 5, we present our experiences with the computing techniques.

2 Preliminaries

2.1 Dixon's Resultant

We assume that readers are familiar with Dixon's resultant method (see [Chi90], [Can90], [KSY94], [KS95]). In particular, the following notations are used in this paper.

Let $\mathcal{F} = \{p_1(x_1,\ldots,x_n),\ldots,p_{n+1}(x_1,\ldots,x_n)\}$ be a set of $n+1$ generic n-degree polynomials in n variables, i.e. there exist nonnegative integers m_1, m_2, ..., m_n such that $p_j = \sum_{i_1=0}^{m_1}\cdots\sum_{i_n=0}^{m_n} a_{j,i_1,\ldots,i_n} x_1^{i_1}\cdots x_n^{i_n}$, for $1 \le j \le n+1$. Let $d_{\max_i} = \max(\text{degree}(p_1,x_i),\ \ldots,\ \text{degree}(p_{n+1},x_i))$, for all $1 \le i \le n$. We form an $(n+1)\times(n+1)$ determinant $\Delta(x_1,\ldots,x_n,\alpha_1,\ldots,\alpha_n)$ as follows:

$$\begin{vmatrix} p_1(x_1,x_2,\ldots,x_n) & \cdots & p_{n+1}(x_1,x_2,\ldots,x_n) \\ p_1(\alpha_1,x_2,\ldots,x_n) & \cdots & p_{n+1}(\alpha_1,x_2,\ldots,x_n) \\ p_1(\alpha_1,\alpha_2,\ldots,x_n) & \cdots & p_{n+1}(\alpha_1,\alpha_2,\ldots,x_n) \\ \cdots & \cdots & \cdots \\ p_1(\alpha_1,\alpha_2,\ldots,\alpha_n) & \cdots & p_{n+1}(\alpha_1,\alpha_2,\ldots,\alpha_n) \end{vmatrix}$$

where α_1, α_2, ..., α_n are n new variables and $p_i(\alpha_1,\ \alpha_2,\ \ldots,\ \alpha_k,\ x_{k+1},\ \ldots,\ x_n)$ stands for uniformly replacing x_j by α_j for $1 \le j \le k$ in p_i. Each of $x_i = \alpha_i$, for all $1 \le i \le n$, is a zero of Δ, so they can be removed by dividing Δ by $\prod_{i=1}^{n}(x_i - \alpha_i)$. Let

$$\delta(x_1,\ldots,x_n,\alpha_1,\ldots,\alpha_n) = \frac{\Delta(x_1,\ldots,x_n,\alpha_1,\ldots,\alpha_n)}{(x_1-\alpha_1)\cdots(x_n-\alpha_n)}.$$

The polynomial δ, called the Dixon polynomial of $\mathcal{F}$, is of degree $((n+1-i)\cdot d_{\max_i})-1$ in α_i and $(i\cdot d_{\max_i})-1$ in x_i, for $1 \le i \le n$.

Any common zero of $\mathcal{F}$, say $(x_1 = c_1,\ldots,x_n = c_n)$ makes the Dixon polynomial vanish, no matter what the values of $\alpha_1,\ldots,\alpha_n$ are; hence, all the coefficients of the various power products of $\alpha_1,\ldots,\alpha_n$ in the Dixon polynomial vanish. Let $\mathcal{E}'$ be the set of all polynomials in $x_1,\ldots,x_n$ which are coefficients of the power products of $\alpha_1,\ldots,\alpha_n$ in δ. This set then has exactly

$$\prod_{i=1}^{n}((n+1-i)\cdot d_{\max_i}) = n!\cdot\prod_{i=1}^{n} d_{\max_i} = s$$

equations (one for each power product of $\alpha_1, \ldots, \alpha_n$), each of which is of degree $(i \cdot d_{\max_i}) - 1$ in x_i. Also, there are

$$\prod_{i=1}^{n}(i \cdot d_{\max_i}) = n! \cdot \prod_{i=1}^{n} d_{\max_i} = s$$

power products in $x_1, \ldots, x_n$ in the equations of $\mathcal{E}'$. Let D be the $s \times s$ coefficient matrix of $\mathcal{E}'$. Then

$$\mathcal{E}' \equiv DX = 0,$$

where X is the vector of the power products of $x_1, \ldots, x_n$. If each power product of $x_1, \ldots, x_n$ (including $x_1^0 \cdots x_n^0 = 1$) is viewed as a new variable, say v_i, then we get a set $\mathcal{E}$ of s homogeneous linear equations in s variables

$$\mathcal{E} \equiv Dv = 0$$

If there exists a common affine zero of $\mathcal{F}$, say $(x_1 = c_1, \ldots, x_n = c_n)$, then this also is a solution to $\mathcal{E}'$. This results in a nontrivial solution to $\mathcal{E}$ i.e. $(v_1 = 1, v_2 = c_1, \ldots)$. Hence, as before, if $\mathcal{F}$ has a common affine zero, then $\mathcal{E}$ has a nontrivial solution, implying that the determinant of D is zero. This gives a necessary condition on the coefficients of $p_1, \ldots, p_{n+1}$ for them to have a common zero. We call D the Dixon matrix and its determinant Dixon's resultant. It was proved by Dixon in [Dix08] that for $n+1$ generic n-degree polynomials in n variables, vanishing of Dixon's resultant is a necessary and sufficient condition for them to have a nontrivial projective zero, or a necessary condition for the existence of an affine zero.

However, the method does not work for most algebraic and geometric problems. For most practical problems, the polynomials are not generic and the matrix set up in Dixon's method, the Dixon matrix, becomes singular or even nonsquare. As a consequence, Dixon's resultant vanishes identically, without providing any information about the common solutions of equations.

2.2 Pseudoinverse Matrices

Probably the most familiar application of matrices is to the solution of systems of simultaneous linear equations. Let

$$Ax = b$$

be such a system, where b is a given vector and x is an unknown vector. If A is nonsingular then there exists a unique solution for x, given by

$$x = A^{-1}b.$$

In the general case, when A may be singular or nonsquare, there may sometimes be no solutions or a multiplicity of solutions. If the system has many solutions, we may desire to have not just one solution but a characterization of all solutions. It is easy to verify that if X is any matrix satisfying $AXA = A$, then $Ax = b$ has a solution if and only if

$$AXb = b,$$

and in this case the most general solution is

$$x = Xb + (I - XA)y,$$

where y is arbitrary. We will call such a matrix X a generalized inverse of A. Unlike the case of the nonsingular matrix, which has a single unique inverse for all purposes, there are different generalized inverses for different purposes. However, in this paper, we are only interested in a generalized inverse whose existence is unique for every matrix (square or nonsquare), namely the Moore–Penrose generalized inverse or pseudoinverse.

The concept of generalized inverses seems to have been first mentioned in print in 1903 by Fredholm [Fre03]. Pseudoinverses of matrices, whose existence was first noted by E. H. Moore (see [Moo20]), who defined a unique inverse (called by him the general reciprocal) for every finite matrix (square or nonsquare). In 1950, Bjerhammar rediscovered Moore's pseudoinverse and also noted the relationship of generalized inverses to solutions of linear systems (see [Bje51b], [Bje51a], [Bje68]). In 1955, Penrose [Pen55] sharpened and extended Bjerhammar's results on linear systems and showed that Moore's pseudoinverse, for a given matrix A, is the unique matrix A^+ satisfying the four equations (1)–(4) of the next section. The latter discovery has been so important and fruitful that this unique inverse is now commonly called the Moore–Penrose inverse or pseudoinverse.

Definitions and Fundamental Properties: In this paper, we use the following definitions and properties from the pseudoinverse theory. More details can be found in [Pen55], [BO71] or [BIG74]. All matrices in this section are defined over the complex number field. Clearly, analogous results can be obtained by assuming the matrices are defined over the real number field.

The following matrix equations are used to define generalized inverses of matrices where $()^*$ denotes the conjugate transpose of the matrix:

$$AXA = A, \tag{1}$$
$$XAX = X, \tag{2}$$
$$(AX)^* = AX, \tag{3}$$
$$(XA)^* = XA. \tag{4}$$

Definition 2.1 *A* pseudoinverse *of a matrix $A \in \mathbb{C}^{m\times n}$ is a matrix $X \in \mathbb{C}^{n\times m}$, denoted in particular by A^+, satisfying (1), (2), (3) and (4).*

Given a matrix $A \in \mathbb{C}^{m\times n}$. Let $X = A^{(1,2,4)}.A.A^{(1,2,3)}$, where $A^{(1,2,4)} = A^*(AA^*)^{(1)}$ and $A^{(1,2,3)} = (A^*A)^{(1)}A^*$. The identity $(BAA^* - CAA^*).(B - C)^* = (BA - CA).(BA - CA)^*$ along with the fact that $AA^* = 0$ implies $A = 0$ gives us the following result:

$$\text{if } BAA^* = CAA^* \text{ then } BA = CA. \tag{5}$$

Similarly, we have

$$\text{if } BA^*A = CA^*A \text{ then } BA^* = CA^* \tag{6}$$
$$\text{if } A^*AB = A^*AC \text{ then } AB = AC \tag{7}$$
$$\text{if } AA^*B = AA^*C \text{ then } A^*B = A^*C \tag{8}$$

By using (5), (6), (7) and (8), it is easy to verify that $A^{(1,2,4)}$ satisfies (1), (2) and (4); $A^{(1,2,3)}$ satisfies (1), (2) and (3); and X satisfies (1), (2), (3) and (4). Moreover, if X and Y are two matrices satisfying (1), (2), (3) and (4). Then

$$\begin{aligned} X &= X(AX)^* = XX^*A^* = X(AX)^*(AY)^* \\ &= XAY = (XA)^*(YA)^*Y = A^*Y^*Y = (YA)^*Y \\ &= Y. \end{aligned}$$

Therefore, we have the following theorem on the existence and uniqueness of the pseudoinverse of a matrix.

Theorem 2.1 *For any matrix A there exists a pseudoinverse matrix A^+. Moreover, the pseudoinverse is unique.*

In the following lemma, we give certain properties of the pseudoinverse of a matrix A. The proof of the lemma is straightforward and can be found in [BO71].

Lemma 2.1 *Let $A \in \mathbb{C}^{m\times n}$, $\lambda \in \mathbb{C}$. Then*

1. $(A^+)^+ = A$,
2. $(A^*)^+ = (A^+)^*$,
3. $(AA^*)^+ = (A^+)^* A^+$ *and* $(A^*A)^+ = A^+(A^+)^*$.
4. $\text{rank } A^+ = \text{rank } A = \text{rank } A^+A$.
5. $(\lambda A)^+ = \lambda^+ A^+$.
6. $A^+ = (A^*A)^+A^* = A^*(AA^*)^+$.
7. AA^+, A^+A, $I - AA^+$ *and* $I - A^+A$ *are Hermitian and idempotent.*
8. $A^+A = I_n$ *if and only if* A *is of full column rank.*
9. $AA^+ = I_m$ *if and only if* A *is of full row rank.*

Note that, in general we may not have $(AB)^+ = B^+A^+$.

3 Extending Dixon's Resultant

We first extend Dixon's resultant method for an arbitrary system $\mathcal{F} \subset K[x_1, \ldots, x_n]$ of $n+1$ equations in n variables, where K is a field of any characteristic set.

3.1 Polynomials of Constant Coefficients

We construct Δ as in the previous section, dividing its determinant by $(x_1 - \alpha_1)\cdots(x_n - \alpha_n)$ to get the Dixon polynomial δ. This polynomial may now have a degree less than or equal to $((n+1-i)\cdot d_{\max_i}) - 1$ in each α_i and less than or equal to $(i \cdot d_{\max_i}) - 1$ in each x_i for all $1 \le i \le n$. We construct the set of linear equation $\mathcal{E}$ from the Dixon polynomial as before, except that now $\mathcal{E}$ may have less than or equal to s equations in less than or equal to s variables. We continue to call the coefficient matrix of $\mathcal{E}$ the Dixon matrix D of size $s_1 \times s_2$ and $v = (v_1, \ldots, v_{s_2})$ the s_2-tuple of the power products of $x_1, \ldots, x_n$ which are viewed as new variables. Note that the Dixon matrix now may be singular or even nonsquare.

Definition 3.1 *Let $\mathcal{F} \subset K[x_1, \ldots, x_n]$ be a system of $n+1$ equations in n variables. We call $(b_1, \ldots, b_n) \in K^n$ a* nontrivial common root *of $\mathcal{F}$ if*

$$(v_1(b_1, \ldots, b_n), \ldots, v_{s_2}(b_1, \ldots, b_n)) \neq \mathbf{0}$$

Obviously, a common root of $\mathcal{F}$ in the algebraic torus $(K \setminus \{0\})^n$ is a nontrivial common root.

Theorem 3.1 *Given a system $\mathcal{F} \subset K[x_1, \ldots, x_n]$ of $n+1$ equations in n variables, D has a nonzero root if and only if*

$$\det(D^+D) = 0,$$

where D^+ is the pseudoinverse of D of size $s_2 \times s_1$.

Proof. Assume that $D\bar{x} = 0$ then $D^+D\bar{x} = 0$ where $\bar{x}$ is a nonzero root of D. Hence $\det(D^+D) = 0$. Conversely, if $\det(D^+D) = 0$ then there exists a nonzero root $\bar{x}$ of D^+D. Therefore $D\bar{x} = DD^+D\bar{x} = 0$. □

Since every matrix in $K^{s_1\times s_2}$ has its unique pseudoinverse (see Theorem 2.1) and we can even obtain the pseudoinverse in an algorithmic manner (see [Tra96]), the theorem means that we have a necessary and sufficient condition for the existence of a nonzero root of the system $\mathcal{E}$.

Corollary 3.1 *Let $\mathcal{F} \subset K[x_1, \ldots, x_n]$ be a system of $n+1$ equations in n variables. If $\mathcal{F}$ has a nontrivial common root then*

$$\det(D^+ D) = 0.$$

Proof. Straightforward from Definition 3.1 and Theorem 3.1. $\square$

3.2 General Case

We now extend the coefficient field of the polynomials to a polynomial ring $K[y_1, \ldots, y_m]$ where K is a field of any characteristic and the new indeterminates $y_1, \ldots, y_m$ are adjoined to the field K. Any ring homomorphism $\varphi : K[y_1, \ldots, y_m] \to K$ such that $y_i \mapsto a_i$ for some $a_i \in K$, $1 \le i \le m$, is called a specialization mapping. We also call $\varphi(D) = (\varphi(d_{ij}))$ a specialization of a Dixon matrix $D = (d_{ij})$. Note that, from now on, any affine zero of $\mathcal{F}$, if it exists, is not a solution to $\mathcal{E}$ but one of its specialization.

Lemma 3.1 *Let D, F, G be matrices of appropriate sizes with entries being in K; a, b, c be in K; and φ be a specialization mapping. Then we have:*

1. $\varphi(a \cdot F \cdot G) = a \cdot \varphi(F) \cdot \varphi(G)$,
2. $\varphi(b \cdot F + c \cdot G) = b \cdot \varphi(F) + c \cdot \varphi(G)$,
3. $\det(\varphi(D)) = \varphi(\det(D))$, *and* $\mathrm{rank}\varphi(D) \le \mathrm{rank} D$,
4. $(\varphi(F))^+ = \varphi(F^+)$.

Proof. (1), (2) and (3) are obvious. For proving (4), it is easy to check that $\varphi(F^+)$ satisfies Equation (1)–(4). $\square$

Let $\mathcal{F} \subset K[y_1, \ldots, y_m][x_1, \ldots, x_n]$ be a nonempty set of $n+1$ equations in n variables $x_1, \ldots, x_n$. For any m-tuple $(a_1, \ldots, a_m) \in K^m$, we denote the corresponding specialization mapping defined by the tuple φ_a.

Definition 3.2 *Let $\mathcal{F} \subset K[y_1, \ldots, y_m][x_1, \ldots, x_n]$ be a system of $n+1$ equations in n variables. We call $(a_1, \ldots, a_m, b_1, \ldots, b_n)$ a* nontrivial common root *of $\mathcal{F}$ if*

$$(v_1(b_1, \ldots, b_n), \ldots, v_{s_2}(b_1, \ldots, b_n)) \neq \mathbf{0},$$

i.e. $(b_1, \ldots, b_n)$ is a nontrivial common root of $\varphi_a(\mathcal{F})$.

Lemma 3.2 *Let $(a_1, \ldots, a_m)$ be an m-tuple in K^m, $\mathcal{F} \subset K[y_1, \ldots, y_m][x_1, \ldots, x_n]$ be a nonempty set of $n+1$ equations in n variables $x_1, \ldots, x_n$. $\varphi_a(D)$ has a nonzero root if and only if*

$$\varphi_a(\det(D^+ D)) = 0,$$

Proof. Let w be a nonzero root of $\varphi_a(D)$. Then

$$(\varphi_a(D))^+ \varphi_a(D) w = \varphi_a(D^+ D) w = 0$$

and hence

$$\varphi_a(\det(D^+ D)) = \det(\varphi_a(D^+ D)) = 0,$$

where $\det(D^+ D)$ is a polynomial in $K[y_1, \ldots, y_m]$. Conversely, if $\varphi_a(\det(D^+ D)) = 0$ then there exists a nonzero affine zero w of $\varphi_a(D))^+ \varphi_a(D)$. Therefore,

$$\varphi_a(D) \varphi_a(D))^+ \varphi_a(D) w = \varphi_a(D) w = 0.$$

$\square$

Lemma 3.3 *Given a system $\mathcal{F} \subset K[y_1, \ldots, y_m][x_1, \ldots, x_n]$ of $n+1$ equations in n variables. If $D \in K(y_1, \ldots, y_m)_{s_2}^{s_1 \times s_2}$ is of full column rank then there exists a* nontrivial necessary condition *for the fact that the system $\mathcal{F}$ has a nontrivial common root. Moreover, for any square submatrix D' of the same rank with D,* $\det(D') = 0$ *is such a necessary condition.*

PROOF. From the previous lemma, $\det(D^+D) = 0$ is such a necessary condition. Let (a, b) be a nontrivial common root of $\mathcal{F}$, D' be any nonsingular submatrix of D. Then $\text{rank}\varphi_a(D) = \text{rank}\varphi_a(D') < \text{rank}D$ and hence $\varphi_a(D')$ must be singular. Therefore $\varphi_a(\det(D')) = 0$ where $\det(D')$ is not identical to zero. □

In case the Dixon matrix is not of full column rank, we need to use the full-rank factorization technique in [Tra96]. In that, for any matrix D, there exist matrices F and G such that

- $D = FG$,
- F is of full-column rank,
- G is of full-row rank.

In that, we can choose F to be any matrix whose columns are a basis for the range space of D. Then the matrix G is uniquely determined. Alternately, we can choose G be any matrix whose rows are a basis for the row space of D. The matrix G can be obtained by, for example, performing elementary row operations on the matrix D. Then the matrix F is uniquely determined. Actually, the full-rank factorization is very useful for improving our method as we will see in the next lemmas.

Lemma 3.4 *If* $\text{rank}\varphi_a(D) < \text{rank}D$ *for a common root (a, b) of $\mathcal{F}$ then there exists a nontrivially necessary condition. Moreover, for any square submatrix D' of the same rank with D,* $\det(D') = 0$ *is such a necessary condition.*

PROOF. We can extend D' to G, a matrix whose rows are a basis for the row space of D. Let $D = FG$ be the corresponding full-rank factorization of D for an appropriate matrix F. Then $\text{rank}\varphi_a(G) < \text{rank}G$ and hence $\varphi_a(\det(D')) = 0$ □

Lemma 3.5 *Let $D = FG$ be a full-rank factorization of D. If* $\text{rank}\varphi(D) = \text{rank}D$ *for all common roots of $\mathcal{F}$ then $\varphi(D) = \varphi(F)\varphi(G)$ is a full-rank factorization of $\varphi(D)$.*

PROOF. Otherwise, $\text{rank}\varphi(D)$ would less than $\text{rank}D$. □

The following corollary shows that the result in [KSY94] is a special case of our results.

Corollary 3.2 *If there exists a submatrix $\bar{D}$ of size $s_1 \times (s_2 - 1)$ by deleting the i^{th} column of D such that* $\text{rank}\bar{D} < \text{rank}D$ *then there exists a nontrivial necessary condition for the fact that the system $\mathcal{F}$ has a nontrivial common root (a, b) where $v_i(b) \neq 0$. Moreover, for any square submatrix D' of the same rank with D,* $\det(D') = 0$ *is such a necessary condition.*

PROOF. Since $v_i(b) \neq 0$, we have $\varphi_a \text{rank}\bar{D} = \varphi_a \text{rank}D < \text{rank}D$. The assertion comes directly from Lemma 3.4. □

In some applications such as implicitization and Geometric Reasoning, we often know in advance that there exists a common root of the system $\mathcal{F}$ in the algebraic torus $(K \setminus \{0\})^n$. In this case, the following corollary is straightforward but found very useful.

Corollary 3.3 *Let $\mathcal{F}$ be a system of $n+1$ equations in n variables. If D is of full row rank then there exists a nontrivial necessary condition for the fact that the system $\mathcal{F}$ has a nontrivial common root (a,b) in the torus $(K\setminus\{0\})^n$. Moreover, for any square submatrix D' of the same rank as D, $\det(D')=0$ is such a necessary condition.*

Note: Lemma 3.3, Corollary 3.2 and Corollary 3.3 are free of any specialization.

From the above lemmas, we could find a nontrivial necessary condition if we could get a complete overview of the solutions of $Dx=0$. Fortunately, this can be achieved by computing the null space of D. Let $D=FG$ be a full-rank factorization of D. The null space of G is also the null space of D. As in Section 2, we know in advance that the most general solution of $Gx=0$ is $(I-G^{+}G)y$ where y is arbitrary. We can even find a basis for the null space of D as follows. Let the columns of G be rearranged, if necessary, so that the first r columns are linearly independent where r is the rank of D. A rearrangement of columns may be interpreted as postmultiplication by a suitable permutation matrix; thus,

$$GQ=[G_1\ \ G_2] \quad \text{or} \quad G=[G_1\ \ G_2]Q^T, \tag{9}$$

where Q is an $s_2\times s_2$ permutation matrix (hence $Q^{-1}=Q^T$) and G_1 consists of r linearly independent columns, so that G_1 is nonsingular. Corresponding to (9), let the vector x in K^{s_2} be partitioned

$$x=\begin{bmatrix} x_1 \\ x_2 \end{bmatrix}, \quad x_1\in K^r \tag{10}$$

In (9), the columns of G_2 are linear combinations of the columns of G_1, say,

$$G_2=G_1T \quad \text{or} \quad T=G_1^{-1}G_2\in K^{r\times(s_2-r)}, \tag{11}$$

where the matrix T is called the *multiplier* corresponding to the partition (9). Using (9), (10) and (11) we have

$$Gx=G_1[I_r\ \ T]Q^T\begin{bmatrix} x_1 \\ x_2 \end{bmatrix}=0, \tag{12}$$

whose general solution is

$$\begin{bmatrix} x_1 \\ x_2 \end{bmatrix}=Q\begin{bmatrix} -T \\ I_{s_2-r} \end{bmatrix}y, \tag{13}$$

where $y\in K^{s_2-r}$ is arbitrary. Note that the null space of G is completely determined by the multiplier T and the permutation matrix Q. In fact (13) shows that the columns of the $s_2\times(s_2-r)$ matrix

$$Q\begin{bmatrix} -T \\ I_{s_2-r} \end{bmatrix}$$

form a basis for the null space of G and therefore a basis for the null space of D. Moreover, since we have to perform the Gaussian elimination algorithm on D anyway (by using fraction-free methods or modular and multivariate sparse interpolation methods), with some more efforts we can diagonalize the matrix by performing a forward elimination. In this case, matrix T is easy to compute.

4 Main Algorithm

In this section, we elaborate the main algorithm.

Algorithm 4.1 (Dixon)

Inputs: A system of $n+1$ nonzero polynomials in n variables.
Output: A nontrivial necessary condition for the fact that the system $\mathcal{F}$ has a nontrivial common root.

1. *Setup the Dixon matrix D of $\mathcal{F}$;*
2. **If** *D is of full column rank* **then**
   (a) *Compute a square submatrix D' of the same rank with D by using, for example modular and multivariate sparse interpolation methods.*
   (b) **Return**($\det(D')$).
3. **If** *D is of full row rank and we know in advance that there exists a nontrivial common root in the algebraic torus $(K \setminus \{0\})^n$* **then**
   (a) *Compute a square submatrix D' of the same rank with D by using, for example modular and multivariate sparse interpolation methods.*
   (b) **Return**($\det(D')$).
4. *Compute a full-rank factorization $D = FG$.*
5. *Check the null space of G which is also the null space of D. As in Section 2, we know in advance that the most general solution of $Gx = 0$ is $(I - G^{+}G)y$ where y is arbitrary.*
6. **If** *there exists an independent column of D* **then**
   Return($\det(D')$). *This is a byproduct of Step 5.*
   Else
   Return(*"failure"*).

REMARK: I would like to emphasize that the growth of the entries of the Dixon matrix will crucially influence the efficiency of the algorithm. To overcome this difficulty, recently in [Tra97] the author presents an algorithm based on modular and multivariate sparse interpolation methods which works on the actual terms of the determinant polynomial. In the next section, we discuss some examples.

5 Experiments

We have implemented the algorithm using CASA, a system for constructive algebraic geometry developed at RISC-Linz and based upon the kernel of the Maple language (i.e. we have our own implementation of algorithms for computations such as Gröbner bases for an arbitrary admissible ordering, Puiseux expansions, resultants, etc. See [MWW93] and [TW97] for more details). All the experiments were carried out on a Sun Ultra-1 workstation under SunOS 5.5 with 128MB RAM.

Example 5.1 (Implicitization) *Given an algebraic surface in parametric form*

$$x_1 = \frac{y_1^2 - y_2^2}{y_2}, \quad x_2 = \frac{y_1^2 - y_2^2}{y_1}, \quad x_3 = \frac{1}{y_1 \cdot y_2}.$$

The objective is to find an implicit form for the surface. We first transform the system of equations into

$$\begin{aligned} q_1 &= x_1 \cdot y_2 - (y_1^2 - y_2^2), \\ q_2 &= x_2 \cdot y_1 - (y_1^2 - y_2^2), \\ q_3 &= x_3 \cdot y_2 - 1, \\ q_4 &= t \cdot y_1 \cdot y_2 - 1. \end{aligned}$$

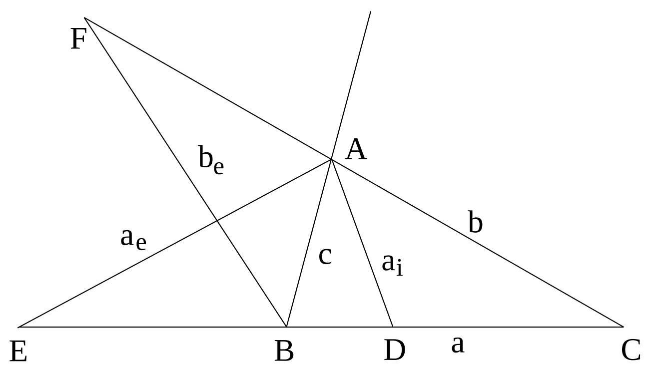
The objective now is to eliminate t, y_2 and y_1 without any constraints on them.

We first compute the Dixon matrix which turns out to be a matrix of size 4×4 and of full column rank. Therefore, by the Lemma 3.3 for any square submatrix D' of the same rank with D, $\det(D') = 0$ is a necessary condition. We perform a Gaussian elimination process, compute the determinant and remove the extra factors getting an implicit form for the surface

$$x_2^4 - x_3 \cdot x_2^3 \cdot x_1^3 - 2 \cdot x_2^2 \cdot x_1^2 + x_1^4.$$

The total time taken was 0.080 second. The total memory space requirements were 298KB.

Example 5.2 (Elimination) *Let ABC be a triangle, a, b and c be the length of the sides BC, AC and AB, a_i and a_e be the length of internal (AD) and external (AE) bisectors of angle A, and b_e be the length of the external angle bisector (BF) of angle B.*



The objective is to determine that given general value of the three angle bisectors, whether one can draw the triangle using only a compass and a ruler. It has been proved in [Her75] that this is possible if and only if we can express a in terms of a_i, a_e and b_e, and the expression is of degree 2^m in a for some integer m. The problem can be transformed into an algebraic form as follows.

$$\begin{aligned} q_1 &= a_i^2 \cdot (b+c)^2 - c \cdot b \cdot (c+b-a) \cdot (c+b+a), \\ q_2 &= a_e^2 \cdot (c-b)^2 - c \cdot b \cdot (a+b-c) \cdot (c-b+a), \\ q_3 &= b_e^2 \cdot (c-a)^2 - a \cdot c \cdot (a+b-c) \cdot (c+b-a). \end{aligned}$$

The objective now is to eliminate b and c without any constraints on them. This example has been used in [GW92] and [KSY94]. In that the authors report solving the problem took about 19 hours on their implementation in Lisp on a SUN 4 workstation and 501 seconds on their implementation in Maple on a SUN sparc 10 respectively.

We first compute the Dixon matrix which turns out to be a matrix of size 13×14 and of full row rank. Since we implicitly assume that b and c are not zero, by the Corollary 3.3 for any square submatrix D' of the same rank with D, $\det(D') = 0$ is a necessary condition. We perform a Gaussian elimination process, compute the determinant and remove the extra factors getting an expression in a, a_i, a_e and b_e of 330 terms and of degree 20 in a. That means given a_i, a_e and b_e, the triangle cannot be constructed using only a compass and a ruler. The total time taken was 57.80 second. The total memory space requirements were 318,337KB.

We present some more examples. In Table 1 we report the matrix sizes, the norm bounds, the term bounds, the computing times (in second) and the memory consumptions (in KB) of Dixon's resultant method. Note that, the extended Dixon's resultant method is far more efficient than other elimination methods such as the Gröbner Bases method and the successive resultant method. In fact, most of the examples cannot be carried out by the Gröbner Bases method or the successive resultant method.

Example 5.3 (Implicitization) *Find an implicit form for the surface given in parametric form*

$$x_1 = \frac{y_1^2 - y_2^2}{y_2}, \quad x_2 = \frac{y_1^2 - y_2^2}{y_1}, \quad x_3 = \frac{1}{y_1 - y_2}.$$

Example 5.4 (Implicitization) *Find an implicit form for the surface given in parametric form*

$$x_1 = \frac{y_2 + 2y_1^4 - 1}{y_2 - y_1 - 2}, \quad x_2 = \frac{y_1^2 y_2 + 1}{y_1}, \quad x_3 = \frac{1}{y_1 \cdot y_2}.$$

Example 5.5 (Elimination) *Eliminate u, v and w from the following system of equations.*

$$\begin{aligned} q_1 &= (x-u)^2 + (y-v)^2 - 1, \\ q_2 &= v^2 - u^3, \\ q_3 &= 2 \cdot v \cdot (x-u) + 3 \cdot u^2 \cdot (y-v), \\ q_4 &= (3w \cdot u^2 - 1) \cdot (2w \cdot v - 1) \end{aligned}$$

Example 5.6 (Elimination) *Determine if the following system has a common root.*

$$\begin{aligned} q_1 &:= x_2^2 + x_1^2 - 1, \\ q_2 &:= x_1 \cdot x_2 - 1, \\ q_3 &:= x_2^2 - x_1. \end{aligned}$$

Fig. 1. Some Experiences

Example	matrix size	norm bound	term bound	time (second)	mem (KB)
E.g. 1	4	10^4	10^3	0.08	298
E.g. 2	14	10^{13}	10^{13}	57.80	318300
E.g. 3	5	10^3	10^2	0.09	422
E.g. 4	13	10^6	10^4	0.32	1136
E.g. 5	16	10^{28}	10^6	54.22	341000
E.g. 6	5	10^5	10^1	0.03	161

6 Conclusion

By using techniques from pseudoinverse theory, we extended Dixon's resultant to an arbitrary system of $n+1$ nontrivial polynomials where the Dixon matrix can be singular or even nonsquare. Our method does not require any precondition. The method has been successfully used to solve the problem of implicitization and other non-trivial problems in the elimination theory.

Acknowledgments

The author would like to thank Prof. Franz Winkler for his encouragements and his support. He would also like to thank Prof. Deepak Kapur for useful discussions.

References

[BIG74] A. Ben-Israel and T. N. E. Greville. *Generalized Inverses: Theory and Applications.* John Wiley & sons, 1974.

[Bje51a] A. Bjerhammar. Applications of calculus of matrices to method of least squares; wth special reference to geodetic calculations. Technical Report 49, Kungl. Tekn. Högsk. Handl., 1951.

[Bje51b] A. Bjerhammar. Rectangular reciprocal matrices with special reference to geodetic calculations. *Bull. Géodésique*, pages 188–220, 1951.

[Bje68] A. Bjerhammar. A generalized matrix algebra. Technical Report 124, Kungl. Tekn. Högsk. Handl., 1968.

[BO71] T. L. Boullion and P. L. Odell. *Generalized Inverse Matrices.* Wiley–Interscience, New York, 1971.

[Can90] J. Canny. Generalized characteristic polynomials. *Journal of Symbolic Computation*, pages 241–250, 1990.

[Cay65] A. Cayley. On the theory of elimination. *Cambridge and Dublin Mathematical Journal*, III:210–270, 1865.

[Chi90] E. Chionh. *Base points, resultants, and the implicit representation of rational Surfaces.* PhD thesis, Dept. Comp. Sci., Uni. of Waterloo, 1990.

[Dix08] A.-L. Dixon. The eliminant of three quantics in two independent variables. In *Proc. London Mathematical Society*, volume 6, pages 468–478, 1908.

[Fre03] I. Fredholm. Sur une classe d'équations fonctionnelles. *Acta Math.*, 27:365–390, 1903.

[GW92] X. S. Gao and D. K. Wang. On the automatic derivation of a set of geometric formulae. In *Mathematics Mechanization Research Preprints*, number 8. Academia Sinica, 1992.

[Her75] I. N. Herstein. *Topics in Algebra.* John Wiley & Sons, second edition edition, 1975.

[KS95] D. Kapur and T. Saxena. Comparison of various multivariate resultant formulations. In *The proceeding of ISSAC'95*, 1995.

[KSY94] D. Kapur, T. Saxena, and L. Yang. Algebraic and geometric reasoning using Dixon resultants. In *The proceeding of ISSAC'94*, 1994.

[Moo20] E. H. Moore. On the reciprocal of the general algebraic matrix (abstract). *Bull. Amer. Math. Soc.*, 26:394–395, 1920.

[MWW93] M. Mnuk, B. Wall, and F. Winkler. Casa reference manual (version 2.1). Technical Report 93–63, RISC, 1993.

[Pen55] R. Penrose. A generalized inverse for matrices. *Proc. Cambridge Philos. Soc.*, 51:406–413, 1955.

[Tra96] Quoc-Nam Tran. *A Hybrid Symbolic-Numerical Approach in Computer Aided Geometric Design (CAGD) and Visualization.* PhD thesis, Research Institute for Symbolic Computation (RISC–Linz), University of Linz, Austria, 1996. dissertation.ps.gz.

[Tra97] Quoc-Nam Tran. Extending dixon's resultant using pseudoinverse matrices. Preprint, 1997.

[TW97] Quoc-Nam Tran and Franz Winkler. Casa reference manual (version 2.3). Technical report, RISC, 1997. Preprint.