Algebraic and Geometric Reasoning using Dixon Resultants*

Deepak Kapur and Tushar Saxena
Institute for Programming and Logics
Department of Computer Science
State University of New York at Albany
Albany, NY 12222
{kapur, saxena}@cs.albany.edu

Lu Yang
Centre for Mathematical Sciences
Chengdu Institute of Computer Applications
Academia Sinica
610041 Chengdu, China

Abstract

Dixon's method for computing multivariate resultants by simultaneously eliminating many variables is reviewed. The method is found to be quite restrictive because often the Dixon matrix is singular, and the Dixon resultant vanishes identically yielding no information about solutions for many algebraic and geometry problems. We extend Dixon's method for the case when the Dixon matrix is singular, but satisfies a condition. An efficient algorithm is developed based on the proposed extension for extracting conditions for the existence of affine solutions of a finite set of polynomials. Using this algorithm, numerous geometric and algebraic identities are derived for examples which appear intractable with other techniques of triangulation such as the successive resultant method, the Gröbner basis method, Macaulay resultants and Characteristic set method. Experimental results suggest that the resultant of a set of polynomials which are symmetric in the variables is relatively easier to compute using the extended Dixon's method.

1 Introduction

There exist many different techniques for solving a system of algebraic polynomial equations. Resultant computations are still perhaps the most popular way to get information about solutions of polynomial equations. Sylvester resultant method is the most studied and used technique for determining a common solution of two polynomial equations in one variable. Implementations of Sylvester resultants are supported in all computer algebra systems including Mathematica, Maple, Reduce and Macsyma.

Successive Sylvester resultant computations can be used to solve a system of polynomial equations in many variables by eliminating variables one at a time. However, the performance of successive resultant techniques is very sensitive to the ordering of variables. Human intervention is required to determine the most efficient ordering and so they are not automatic methods. Successive elimination methods are also very inefficient, in fact, it is more efficient to directly compute a condition for the existence of common zeros without eliminating variables one at a time. Macaulay's multivariate resultant method is one such method, and it has recently gotten popularized [3]. Unfortunately, efficient implementations of Macaulay's method are not available in any computer algebra method. Other alternatives such as Grobner basis [2] and Characteristic set methods [14][6] don't seem to work well either.

Dixon's method is an efficient method to simultaneously eliminate variables from a system of nonhomogeneous polynomial equations, but unfortunately, it does not work for most algebraic and geometric problems. The main result reported in Dixon's original paper [7] was a method to obtain the resultant of a set of three *generic bidegree* polynomials. This is a generalization of Cayley's formulation [4] of Bezout's efficient method for computing the resultant of two univariate polynomials. Dixon wrote that his method generalized to any $n+1$ *generic ndegree*[1] polynomials in n variables. That is, his method gives the resultant of $n+1$ generic ndegree polynomials. For arbitrary set of $n+1$ nonhomogeneous polynomials in n variables (i.e. not necessarily generic and ndegree), his method gives a necessary condition (henceforth called the *Dixon resultant*) for the existence of a common affine zero.

For most problems which arise in geometry, the matrix set up in Dixon's method, the *Dixon matrix*, becomes singular. As a consequence, the Dixon resultant vanishes identically, without providing any information about the common solutions of equations. This is perhaps the reason that Dixon's method has not been widely used, even though it is quite efficient. Chionh in [5] suggested using perturbation of certain coefficients to obtain nonzero conditions, but this is a nonautomatic method and requires human expertise. Canny in [3] defined the Generalized Characteristic Polynomial for Macaulay resultants which is a systematic way of perturbing a polynomial system so that one gets nonzero conditions for the existence of affine solutions. This can also be achieved for Dixon resultants by mechanically perturbing the polynomials to be ndegree, but this leads to a larger size Dixon matrix with larger entries too. Computing the determinant of such a matrix is cumbersome and leads to

*Supported in part by a grant from United States Air Force Office of Scientific Research AFOSR-91-0361.

[1] $n+1$ nonhomogeneous polynomials $p_1, \cdots, p_{n+1}$ in $x_1, \cdots, x_n$ are said to be *generic ndegree* if there exist nonnegative integers $m_1, \cdots, m_n$ such that each

$$p_j = \Sigma_{i_1=0}^{m_1} \cdots \Sigma_{i_n=0}^{m_n} a_{j,i_1,\ldots,i_n} x_1^{i_1} \cdots x_n^{i_n} \ for \ all \ 1 \le j \le n+1.$$

inefficiency.

In this paper, we overcome these restrictions by extracting nonzero conditions directly from the Dixon matrix of the system of polynomials. The proposed method does not involve any perturbation, instead we identify and prove a condition on singular Dixon matrices under which, *nonzero* necessary conditions for the existence of a common solution for a system of equations can be extracted. Based on this result, an algorithm for computing the nonzero necessary conditions for the existence of affine common zeros of a system of polynomials directly from its Dixon matrix is developed. It is found that most of the problems for which Dixon's original method was inapplicable, can be solved using our extension of it. Moreover, unlike perturbation techniques [3], the proposed method does not introduce any new variable and terms into the system of polynomials. Because of this, computation is not made any more difficult. Finally, unlike successive elimination techniques and perturbation techniques of [5], our method is fully automatic and does not require any human intervention.

We successfully proved many nontrivial algebraic and geometric identities and theorems within a few minutes (the maximum time for any example was less than 9 minutes) using a naive implementation of our algorithm in MAPLE. We mention five such examples in this paper. Two of these five examples seem intractable by implementations of most other techniques of elimination. The other three examples also take too long to compute using other techniques. Implementations of both, Sylvester's resultant and Bezout's resultant in MAPLE were tried to perform successive resultant computations, but they ran up to a day before running out of memory on all examples. We also tried to compute the lexicographic Gröbner basis on MACAULAY (Bayer and Stillman [1]) as well as MAPLE, but they too ran out of memory after running upto a day. Under block orderings, MACAULAY could not compute the resultant of two examples. For the other three that it could solve, it took substantially longer time than our algorithm. Macaulay resultants (GCP if the Macaulay matrix is singular [3]) were also implemented and tried on MAPLE, but they failed for three examples and took longer time on the other two. Finally, perturbation techniques similar to the GCP of Canny were tried to obtain nonsingular Dixon matrices, but perturbation of polynomials resulted in a blowup of the order of the Dixon matrix due to additional terms and consequently, the determinant computation could not be successfully performed on MAPLE. These preliminary findings, though far from complete, are very encouraging and suggest that further studies are needed to examine methods based on Dixon resultants.

In the next section, we review Cayley's formulation of Bezout's method for two univariate polynomials, and Dixon's method for any $n+1$, generic ndegree polynomials in n variables. Section 2.3 outlines the shortcomings of Dixon's method and the reason for its unapplicability to most geometric problems. In section 3, an algorithm is developed which overcomes this restriction and extracts a nonzero necessary condition for the existence of affine zeros from the Dixon matrix, *even* when it is singular. In section 4, two detailed examples are presented to illustrate the application of our method. In section 5, the advantages of this method over other techniques are discussed, and finally, in section 6, some empirical results are reported.

2 Review of Dixon Resultants

In the next subsection, we describe Cayley's method to determine the resultant for two univariate polynomials. In the subsection following that, we describe an extension by Dixon which gives a general method to determine resultant of $n+1$ *generic ndegree* polynomials in n variables. A detailed exposition of Dixon's method can be found in [12].

2.1 Cayley's Method for Two Polynomials in One Variable

Even though the analysis in this section was developed by Cayley in [4], we will mostly use Dixon's name while developing the notation so that a uniform notation can be carried to the generalization that is presented in the next subsection.

Consider a set $\mathcal{F}$ of two univariate polynomials $\{p_1(x_1), p_2(x_1)\}$. Let $d_{max} = max(degree(p_1, x_1), degree(p_2, x_1))$, where by $degree(p_1, x_1)$, we mean the maximum degree of x_1 in p_1. Consider the polynomial

$$\Delta(x_1, \alpha_1) = \begin{vmatrix} p_1(x_1) & p_2(x_1) \\ p_1(\alpha_1) & p_2(\alpha_1) \end{vmatrix},$$

where α_1 is a new variable and $p_1(\alpha_1)$ stands for uniformly replacing x_1 by α_1 in p_1. Making $x_1 = \alpha_1$ would make $\Delta = 0$ which means that $x_1 - \alpha_1$ divides Δ. Let,

$$\delta(x_1, \alpha_1) = \frac{\Delta(x_1, \alpha_1)}{(x_1 - \alpha_1)} = \frac{p_1(x_1)p_2(\alpha_1) - p_2(x_1)p_1(\alpha_1)}{(x_1 - \alpha_1)}$$

We call δ the **Dixon Polynomial**. The polynomial δ is a $d_{max} - 1$ degree polynomial in α_1 and is symmetric in x_1 and α_1. Every common zero of $p_1(x_1)$ and $p_2(x_1)$ is a zero of $\delta(x_1, \alpha_1)$ no matter what value α_1 has; thus at a common zero of p_1 and p_2, the coefficient of every power product of α_1 in $\delta(x_1, \alpha_1)$ must be 0. This gives a set (say $\mathcal{E}'$) of d_{max} equations corresponding to the coefficients of all the power products of α_1 (viz. α_1^i for each $0 \le i \le d_{max} - 1$), each of which is a $d_{max} - 1$ degree polynomial in x_1. So, if D is the $d_{max} \times d_{max}$ coefficient matrix of $\mathcal{E}'$, then

$$\mathcal{E}' \equiv D \begin{pmatrix} 1 \\ x_1 \\ x_1^2 \\ \vdots \\ x_1^{d_{max}-1} \end{pmatrix} = \begin{pmatrix} 0 \\ 0 \\ 0 \\ \vdots \\ 0 \end{pmatrix}.$$

If each power product of x_1 (including $x_1^0 = 1$) is viewed as a new variable, v_i, then we get a set $\mathcal{E}$ of d_{max} homogeneous linear equations in d_{max} variables:

$$\mathcal{E} \equiv D \begin{pmatrix} v_1 \\ v_2 \\ v_3 \\ \vdots \\ v_{d_{max}} \end{pmatrix} = \begin{pmatrix} 0 \\ 0 \\ 0 \\ \vdots \\ 0 \end{pmatrix}.$$

If a common affine zero exists for $\mathcal{F}$ (say $x_1 = c_1$), then this is a solution for $\mathcal{E}'$ also. This results in a nontrivial solution for $\mathcal{E}$ viz., $v_1 = 1, v_2 = c_1, v_3 = c_1^2, \cdots, v_{d_{max}} = c_1^{d_{max}-1}$. Hence 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 and p_2 for them to have a common zero.

It was proved by Cayley in [4] that vanishing of the determinant of D, the Dixon resultant of $\mathcal{F}$, is a necessary and sufficient condition for $\mathcal{F}$ to have a nontrivial common projective zero. In fact, Bezout matrix coincides with the Dixon matrix for the univariate case.

2.2 Dixon's Generalization to Two and More Variables

Dixon explicitly generalized Cayley's method presented in the previous subsection to the two variable case in [7], but it can be easily generalized to any number of variables (and Dixon alluded to this), so we present this generalization under Dixon's name too.

Let $\mathcal{F} = \{p_1(x_1, \cdots, x_n), \cdots, p_{n+1}(x_1, \cdots, x_n)\}$ be the set of $n+1$ *generic ndegree* polynomials in n variables. Let,

$$d_{max_i} = max(degree(p_1, x_i), \cdots, degree(p_{n+1}, x_i)),$$

for all $1 \leq i \leq n$.

We form an $(n+1) \times (n+1)$ determinant similar to the determinant in the last section. Let this determinant be defined as follows:

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

where $\alpha_1, \cdots, \alpha_n$ are new variables and $p_i(\alpha_1, \cdots, \alpha_k, x_{k+1}, \cdots, x_n)$ stands for uniformly replacing x_j by α_j for $1 \leq j \leq k$ in p_i.

Each of $x_i = \alpha_i$, for all $1 \leq i \leq n$, is a zero of Δ, so they can be removed by dividing Δ by $\Pi_{i=1}^{n}(x_i - \alpha_i)$. Let

$$\delta(x_1, \cdots, x_n, \alpha_1, \cdots, \alpha_2) = \frac{\Delta(x_1, \cdots, x_n, \alpha_1, \cdots, \alpha_n)}{(x_1 - \alpha_1) \cdots (x_n - \alpha_n)}.$$

The polynomial δ, which is the **Dixon polynomial**, is of degree $((n+1-i) \times d_{max_i}) - 1$ in α_i and $(i \times d_{max_i}) - 1$ in x_i for all $1 \leq i \leq n$.

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

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

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

$$\Pi_{i=1}^{n} i \times d_{max_i} = n! \times \Pi_{i=1}^{n} d_{max_i} = s$$

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

$$\mathcal{E}' \equiv D \begin{pmatrix} 1 \\ x_1 \\ x_2 \\ \vdots \\ x_n \\ x_1^2 \\ x_1 x_2 \\ x_2^2 \\ \vdots \\ x_n^2 \\ \vdots \\ \Pi_{i=1}^{n} x_i^{i \times d_{max_i} - 1} \end{pmatrix} = \begin{pmatrix} 0 \\ 0 \\ 0 \\ \vdots \\ 0 \end{pmatrix}.$$

If each power product of $x_1, \cdots, x_n$ (including $x_1^0 \cdots x_i^0 = 1$) is viewed as a new variable, v_i, then we get a set $\mathcal{E}$ of s homogeneous linear equations in s variables:

$$\mathcal{E} \equiv D \begin{pmatrix} v_1 \\ v_2 \\ v_3 \\ \vdots \\ v_s \end{pmatrix} = \begin{pmatrix} 0 \\ 0 \\ 0 \\ \vdots \\ 0 \end{pmatrix}.$$

If a common affine zero exists for $\mathcal{F}$ (say $x_1 = c_1, \cdots, x_n = c_n$), then this is a solution for $\mathcal{E}'$ also. This results in a nontrivial solution for $\mathcal{E}$ viz., $v_1 = 1, v_2 = c_1, v_3 = c_2, \cdots, v_s = c_1^{d_{max_1}-1} c_2^{2d_{max_2}-1} \cdots c_n^{nd_{max_n}-1}$. 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 $p_1, \cdots, p_{n+1}$ for them to have a common zero. As before, D the **Dixon matrix** and its determinant the **Dixon resultant**.

It was proved by Dixon in [7] that for $n+1$ *generic ndegree* polynomials in n variables, vanishing of the Dixon 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. Moreover, $det(D)$ is not identically zero. Recall that *generic ndegree* means that there exists a set of n integers, $k_1, \cdots, k_n$ such that all polynomials, $p_1, \cdots, p_{n+1}$, are of the kind:

$$p_j = \Sigma_{i_1=1}^{k_1} \cdots \Sigma_{i_n=1}^{k_n} a_{j,i_1,\cdots,i_n} x_1^{i_1} \cdots x_n^{i_n}, \ 1 \leq j \leq n+1.$$

where each a is a distinct indeterminate. So for such a set of polynomials, the Dixon resultant is a polynomial in all of a's that is not identically zero but vanishes for those particular values of a's (from the algebraic closure of the field of rational numbers, say) for which $\mathcal{F}$ has a common affine zero. Next we discuss the case when the polynomials are not generic and ndegree.

2.3 Case of Arbitrary Polynomials

Consider the following three polynomials in two variables.

$$\begin{aligned} p_1 &= x^2 + axy - y + b \\ p_2 &= xy + (a+b)y - c \\ p_3 &= bx + y + 2ac \end{aligned}$$

Here a, b and c are parameters. This set of polynomials is not generic (because for example, the coefficient of x^2 in p_1, 1, is not an independent parameter, and the coefficient of y in p_2 is not independent since it depends on the coefficients of p_1). This set is also not ndegree (because for example, even though the maximum degree of x and y is two and one respectively, there is no term x^2y in any of the polynomials. So what must one do in this case? The analysis in the previous section was developed for generic ndegree polynomials, so it would not work directly for this set of polynomials. It is possible to write down the generic polynomials of ndegree (2,1) in x and y:

$$\begin{aligned} p_1' &= a_1x^2y + a_2x^2 + a_3xy + a_4x + a_5y + a_6 \\ p_2' &= a_7x^2y + a_8x^2 + a_9xy + a_{10}x + a_{11}y + a_{12} \\ p_3' &= a_{13}x^2y + a_{14}x^2 + a_{15}xy + a_{16}x + a_{17}y + a_{18} \end{aligned}$$

Now construct its 4×4 Dixon matrix D. Compute its determinant $det(D)$ and substitute the values $a_1 = 0, a_2 = 1, a_3 = a, a_4 = 0, a_5 = -1, \cdots, a_{18} = 2ac$ into $det(D)$. This would give a polynomial in a, b and c (call it p_{dix}). The polynomial p_{dix} vanishes at all those values of a, b and c at which p_1, p_2 and p_3 has an affine zero. This same approach could be followed for any arbitrary set of polynomials $\mathcal{F}$ as follows:

1. Construct generic ndegree polynomials $\mathcal{F}'$ which when specialized in a certain way (say a mapping ψ from the parameters of $\mathcal{F}'$ to the parameters of $\mathcal{F}$), becomes $\mathcal{F}$.
2. Find the Dixon resultant of $\mathcal{F}'$.
3. Specialize the parameters in this Dixon resultant polynomial using ψ to get the polynomial p_{dix}.

When this approach is followed, often p_{dix} becomes identically zero. This is something which was avoided while working with generic ndegree polynomials. More importantly, this approach also results in inefficiency because $\mathcal{F}'$ has larger polynomials (substantially in some cases) than $\mathcal{F}$.

We have adopted a different approach in this paper. Remove the restriction of generic ndegree from the construction of the previous section. So now, we are given a set of any arbitrary polynomials. Construct Δ as in the last subsection. Divide its determinant by $(x_1-\alpha_1)\cdots(x_n-\alpha_n)$ to get the Dixon polynomial δ. This polynomial now may have a degree less than or equal to $((n+1-i)\times d_{max_i})-1$ in each α_i and less than or equal to $(i \times d_{max_i})-1$ in each x_i for all $1 \le i \le n$. Construct the set of linear equations $\mathcal{E}$ 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 $\mathcal{E}$ to be the **Dixon matrix** D *which may be an* $s_1 \times s_2$ *matrix where* $s_1 \le s$ *and* $s_2 \le s$.

Notice though that the determinant of the Dixon matrix may not give us a necessary condition for the existence of affine zeros of $\mathcal{F}$ because $\mathcal{E}$ may not have a nontrivial solution even when $\mathcal{F}$ has a common affine zero. This is because the Dixon matrix may not contain the column corresponding to the monomial $x_1^0 \cdots x_n^0$, and hence if $\mathcal{F}$ has only $x_1 = 0, \cdots, x_n = 0$ as the common solution, then $\mathcal{E}$ only has a trivial solution. Even if the Dixon matrix *does* contain the column corresponding to $x_1^0 \cdots x_n^0$, it could be singular. Worst of all, the Dixon matrix could be a rectangular matrix (because s_1 and s_2 may not be same), in which case, there does not even exist the possibility of computing the determinant. We deal with all of these possibilities together in the next section.

3 Dealing with Singular Dixon Matrices

First let us formalize the notion for this new framework of arbitrary polynomials. Let there be m parameters $a_1, \cdots, a_m$ and $\mathcal{P} = \mathcal{Q}[a_1, \cdots, a_m]$, the ring of all polynomials in them ($\mathcal{Q}$ is the field of rational numbers). The problem being solved is, given a set $\mathcal{F}$ of $n+1$ polynomials from $\mathcal{P}[x_1, \cdots, x_n]$, give a polynomial from $\mathcal{P}$ that is not identically zero, but which must vanish for all those particular values of the parameters $a_1, \cdots, a_m$ from $\bar{\mathcal{Q}}^m$ ($\bar{\mathcal{Q}}$ is the algebraic closure of $\mathcal{Q}$) for which $\mathcal{F}$ has a common affine zero in $\bar{\mathcal{Q}}^n$. We will call any such polynomial a **Projection operator,** ϱ.

Consider a slightly different problem from the one posed in the previous paragraph as follows: A set of constraints $\mathcal{C}$ on the variables $x_1, \cdots, x_n$ of the form $x_{i_1} \neq 0 \wedge \cdots \wedge x_{i_k} \neq 0$ are also given, and we are looking for a polynomial in $\mathcal{P}$ that is not identically zero, but vanishes on all those particular values of parameters from $\bar{\mathcal{Q}}^m$ for which $\mathcal{F}$ has a common affine zero in $\bar{\mathcal{Q}}^n$ *which also satisfies the set of constraints* $\mathcal{C}$. We will call this the **Projection operator modulo $\mathcal{C}$,** $\varrho_{\mathcal{C}}$.

If one is looking for a necessary condition, ϱ for the existence of affine solutions, then $\mathcal{C}$ can be ϕ, and $\varrho = \varrho_\phi$. But the problem of computing $\varrho_{\mathcal{C}}$ in itself is important because in areas such as geometric theorem proving, one occasionally encounters constraints under which solutions are sought. So, we will now give a method which, given the set of $n+1$ polynomials $\mathcal{F} \subset \mathcal{P}[x_1, \cdots, x_n]$ and the set of constraints $\mathcal{C}$ on the variables, gives the required necessary condition on the parameters which is not identically zero *provided* the Dixon matrix satisfies a certain condition.

Recall that the Dixon matrix is of dimension $s_1 \times s_2$ where $s_1 \le n!\Pi_{i=1}^n d_{max_i}$ rows (corresponding to the coefficients of all power products of αs in the Dixon polynomial) and $s_2 \le n!\Pi_{i=1}^n d_{max_i}$ columns (corresponding to all power products of x_is in the Dixon polynomial). Let the columns of the Dixon matrix be denoted by m_i (m_1 being the first column of the Dixon matrix and so on). For any column indexed by m_i, let $monom(m_i)$ denote the monomial (or power product in $x_1, \cdots, x_n$) corresponding to that column. Also, given a set of constraints $\mathcal{C}$, let $nvcol(\mathcal{C})$ denote the set of all columns m_i such that $\mathcal{C} \Rightarrow monom(m_i) \neq 0$.

First let us establish a lemma about polynomials in which all coefficients are from the algebraic closure of the field of rational numbers $\bar{\mathcal{Q}}$. Given a set of $n+1$ polynomials $\mathcal{G} \subset \bar{\mathcal{Q}}[x_1, \cdots, x_n]$, let N be the $s_1 \times s_2$ Dixon matrix of $\mathcal{G}$. Also let,

$$\mathcal{N}_1 = \{X | X \textit{ is an } s_1 \times (s_2-1) \textit{ submatrix of } N \textit{ obtained by deleting a column which belongs to } nvcol(\mathcal{C}) \textit{ from } N\}$$

Notice that if $\mathcal{C}$ is $x_1 \neq 0 \wedge \cdots \wedge x_n \neq 0$ then $\mathcal{N}_1$ contains all the $s_1 \times (s_2-1)$ submatrices of N. Now we prove a lemma relating the rank of the Dixon matrix N with the ranks of all the matrices which are elements of $\mathcal{N}_1$ as follows.

Lemma 3.1 *If $\mathcal{G}$ has a common affine zero which satisfies $\mathcal{C}$, then*

$$\forall X \epsilon \mathcal{N}_1, rank(X) = rank(N)$$

Proof : Let $x_1 = c_1, \cdots, x_n = c_n$ be the common affine zero of $\mathcal{F}$ which satisfies $\mathcal{C}$. Since each of the s_2 variables in the set of s_1 homogeneous linear equations, $\mathcal{E}$, stands for a monomial in $x_1, \cdots, x_n$, this solution for $\mathcal{F}$ generates a solution for $\mathcal{E}$, say $v_1 = C_1, v_2 = C_2, \cdots, v_{s_2} = C_{s_2}$. So, if $m_1, \cdots, m_{s_2}$ are the $s_1 \times 1$ column vectors of the Dixon matrix N then,

$$C_1 m_1 + C_2 m_2 + \cdots + C_i m_i + \cdots + C_{s_2} m_{s_2} = 0.$$

Let X be any element of $\mathcal{N}_1$. Then X was obtained from N by deleting some column of N (say m_i) which belonged to $nvcol(\mathcal{C})$. But this means that $monom(m_i)$ cannot vanish on any solution which satisfies $\mathcal{C}$, or in other words, $C_i \neq 0$. So dividing the above equation by C_i and rearranging terms on both sides, we get the equation:

$$m_i = \frac{-C_1}{C_i} m_1 + \frac{-C_2}{C_i} m_2 + \cdots + \frac{-C_{i-1}}{C_i} m_{i-1} + \frac{-C_{i+1}}{C_i} m_{i+1} + \cdots + \frac{-C_{s_2}}{C_i} m_{s_2}$$

In other words, the column vector m_i is a linear combination of the column vectors $m_1, \cdots, m_{i-1}, m_{i+1}, \cdots, m_{s_2}$. Hence, the dimension of the column vector space spanned by $m_1, \cdots, m_{i-1}, m_{i+1}, \cdots, m_{s_2}$ is the same as the dimension of the column vector space spanned by $m_1, \cdots, m_{s_2}$. As a consequence, the rank of the matrix formed by the columns $m_1, \cdots, m_{i-1}, m_{i+1}, \cdots, m_{s_2}$ (viz., X) must be the same as that of the matrix formed by all the column vectors (viz., N), i.e., $rank(X) = rank(N)$. □

Now let's get back to the case when the polynomials have coefficients from $\mathcal{P}$. Let $\mathcal{F} \subset \mathcal{P}[x_1, \cdots, x_n]$ be a set of $n+1$ polynomials and D the Dixon matrix of $\mathcal{F}$ (with entries now from $\mathcal{P}$). Let $r = rank(D)$. Also, let $\mathcal{D}_1$ be the set of all $s_1 \times (s_2 - 1)$ matrices obtained from D by deleting a column which is an element of $nvcol(\mathcal{C})$ (just as $\mathcal{N}_1$ was obtained from N previously).

Let $\phi : \{a_1, \cdots, a_m\} \mapsto \bar{\mathcal{Q}}$ be a mapping which gives values to the parameters from the algebraic closure of $\mathcal{Q}$. By abuse of notation $\phi(\mathcal{F})$,$\phi(D)$, and $\phi(\mathcal{D}_1)$ are the results of substituting those values for the parameters in $\mathcal{F}$, D and $\mathcal{D}_1$ respectively and removing all zero rows and columns from D and elements of $\mathcal{D}_1$. Finally, let

$$\mathcal{R} = \{Y | Y \ is \ an \ r \times r \ nonsingular \ submatrix \ of \ D\}$$

Note that $\mathcal{R}$ is never an empty set (because $rank(D) = r$) and moreover, for all $Y \epsilon \mathcal{R}$, $det(Y) \not\equiv 0$ (because elements of $\mathcal{R}$ are nonsingular). Now we have the following theorem.

Theorem 3.2 *If $\exists X \epsilon \mathcal{D}_1$ s.t. $rank(X) < rank(D)$ then for all $Y \epsilon \mathcal{R}$, $\phi(det(Y))$ vanishes if $\phi(\mathcal{F})$ has a common affine zero which satisfies $\mathcal{C}$.*

Proof : Let $\mathcal{G} = \phi(\mathcal{F})$, then it is easy to see that $N = \phi(D)$ is the Dixon matrix of $\mathcal{G}$ and $\mathcal{N}_1 = \phi(\mathcal{D}_1) - \{N\}$ is the set of all submatrices of N obtained by deleting a column which is an element of $nvcol(\mathcal{C})$ from N. So by the previous lemma, if $\mathcal{G} = \phi(\mathcal{F})$ has a common affine zero which satisfies $\mathcal{C}$ then

$$\forall X \epsilon \mathcal{D}_1, rank(\phi(D)) = rank(\phi(X)). \tag{1}$$

Also, the rank of any matrix with entries from $\mathcal{P}$ cannot increase when values from $\bar{\mathcal{Q}}$ are substituted for the parameters, so

$$\forall X \epsilon \mathcal{D}_1, rank(\phi(X)) \leq rank(X). \tag{2}$$

If there is an $X \epsilon \mathcal{D}_1$ which satisfies the premise of the lemma then

$$rank(X) < rank(D) = r, \tag{3}$$

and from equations (1), (2) and (3) we get,

$$rank(\phi(D)) = rank(\phi(X)) \leq rank(X) < rank(D) = r$$

$$i.e., \ rank(\phi(D)) < r.$$

This means that once the values of parameters are substituted in D, its rank becomes less than r i.e., all the $r \times r$ submatrices of D (which are exactly the elements of $\mathcal{R}$) become singular, which means that their determinants vanish at these values of the parameters. □

Theorem 3.2 suggests the following algorithm to obtain $\varrho_{\mathcal{C}}$. Check if $\exists X \epsilon \mathcal{D}_1 \ s.t. \ rank(X) < rank(D)$. If this is true, then the determinant of any element of $\mathcal{R}$ gives the required polynomial. But we need to be able to efficiently perform the check, obtain an element of $\mathcal{R}$ and finally compute its determinant. To perform the check, the following lemma is of help.

Let $\bar{w} = (w_1, \cdots, w_{s_2})^T$ be the $s_2 \times 1$ vector which satisfies the matrix equation $D\bar{w} = \bar{0}$, then

Lemma 3.3 *$\exists X \epsilon \mathcal{D}_1$ s.t. $rank(X) < rank(D)$ if and only if there exists some $0 \leq i \leq s_2$ such that the $w_i = 0$ and $\mathcal{C} \Rightarrow monom(m_i) \neq 0$.*

Proof : First of all, let us write the expanded version of the matrix equation $D\bar{w} = \bar{0}$:

$$m_1 w_1 + m_2 w_2 + \cdots + m_{s_2} w_{s_2} = \bar{0}$$

Only if: Let $X \epsilon \mathcal{D}_1 \ s.t. \ rank(X) < rank(D)$. Let m_i be that row of D whose deletion resulted in X. Then, by definition of $\mathcal{D}_1$, $\mathcal{C} \Rightarrow monom(m_i) \neq 0$. Also, $rank(X) < rank(D)$, implies that m_i is linearly independent. This means that $w_i = 0$ because otherwise, the above equation can be divided by w_i and then by rearranging terms, m_i can be expressed as a linear combination of other columns, hence implying that m_i is linearly dependent and resulting in a contradiction.

If: Assume that there exists some $0 \leq i \leq s_2$ such that $w_i = 0$ and $\mathcal{C} \Rightarrow monom(m_i) \neq 0$. The latter assumption means that the matrix obtained by deleting m_i from D (say X) must be in $\mathcal{D}_1$. But also since $w_i = 0$, it follows that column m_i is linearly independent of other columns of D, hence implying that $rank(X) < rank(D)$. □

Finally, in accordance with theorem 3.2, we would like to get the determinant of any element in $\mathcal{R}$ which will give the required necessary condition on the parameters. This is achieved due to the following lemma. Let D_{row} have the following properties:

1. D_{row} is row-reduced, i.e., each column of D_{row} which contains the leading non-zero entry of some row has all its other entries 0.
2. D_{row} is row-equivalent to D, i.e., D_{row} can be obtained from D by a finite sequence of the following two steps:
   (a) **Elimination step:** Replacement of i^{th} row of D by the i^{th} row plus d times j^{th} row, d is any rational function in the parameters, and $i \neq j$.
   (b) **Pivoting step:** Interchange two rows (say the i^{th} and the j^{th} rows) of D.

Notice that D_{row} can be constructed from D by simple Gaussian elimination and many computer algebra systems already support such an operation ($D_{row} = gausselim(D)$ in MAPLE).

Lemma 3.4 *There exists some $Y \epsilon \mathcal{R}$ such that the product of all the pivot elements of D_{row} is equal to $det(Y)$.*

Proof: Assign to the j^{th} row of D, a label l_j. Now obtain the matrix D_{row} which is row-equivalent to D by successively applying a permutation of the above two steps, except that whenever the pivoting step is applied, interchange the labels of the rows too. In the elimination step, the labels remain the same. Since the rank of D is r, D_{row} will have r pivots. Let $m_{i_1}, \cdots, m_{i_r}$ be the pivot columns, and $l_{j_1}, \cdots, l_{j_r}$ the labels of pivot rows of D_{row}. Then it is easy to show that the product of all pivot elements of D_{row} is the determinant of the $r \times r$ submatrix (say Y) of D obtained by the rows labelled by $l_{j_1}, \cdots, l_{j_r}$ and the columns indexed by $m_{i_1}, \cdots, m_{i_r}$ in D. Since the product of pivot elements is not identically zero, Y is not singular so it is in $\mathcal{R}$ by definition. □

Based on theorem 3.2 and lemmas 3.3 and 3.4, we can get the following algorithm. Assume that the set of $n+1$ polynomials $\mathcal{F} \subset \mathcal{P}[x_1, \cdots, x_n]$ and the constraints $\mathcal{C}$ are given. The following algorithm checks if the precondition in theorem 3.2 is true in which case it returns ϱ_C. If the precondition of theorem 3.2 is not true, then this heuristic fails.

1. Set up the $s_1 \times s_2$ Dixon matrix (D) of $\mathcal{F}$.
2. Solve the matrix equation $D\bar{w} = \bar{0}$. Let $\bar{w} = (w_1, \cdots, w_{s_2})$ denote the solution.
3. Find out if there exists an w_i in $\bar{w}$ such that $w_i = 0$ and also $\mathcal{C} \Rightarrow monom(m_i) = 0$. If such an w_i exists then
   (a) Compute D_{row}.
   (b) Return the product of all the pivots of D_{row}.
4. Else, this heuristic fails.

Theorem 3.5 *If the Dixon matrix of a set of polynomials satisfies the precondition of theorem 3.2, then the above algorithm computes ϱ_C.*

Proof : Follows directly from theorem 3.2 and lemmas 3.3 and 3.4. □

There are two points to be noted here. First, though there exist examples where the condition in step (3) is not true, they are rare. In all the examples from geometry that we tried, D was singular many times, but the condition of step (3) always held. Secondly, once the polynomial in the parameters is obtained from the algorithm, it can usually be factored. Since the vanishing of this polynomial is only a necessary condition, occasionally it will be the case that some of the factors do not vanish on any of the solutions. These factors can be removed to obtain a smaller ϱ_C. This is done manually as suggested by specific algebraic or geometric problem. Our experience has been that it is usually easy to identify such extraneous factors, but automatic method to accomplish this need to be further explored. In the next section, we discuss some examples.

4 Geometric Reasoning using Dixon resultants

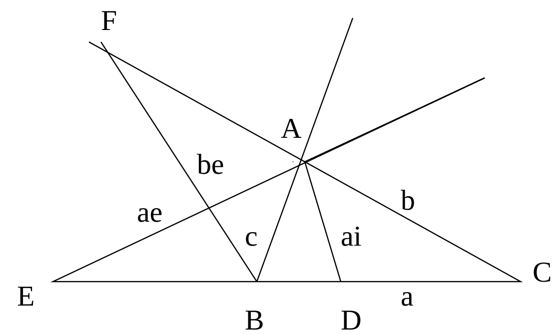
We implemented the proposed algorithm on a SPARCstation 10 in MAPLE using the primitive operations such as *linsolve, gausselim*, etc., available in its linear algebra package. No attempt was made to optimize Maple code. We believe the algorithm can be made much faster using interpolation techniques for computing determinants as reported by Canny and Manocha in [13].

Two geometric identities derived using the algorithm are presented in this section. A few more identities will be presented without details in a later section.

Example 1: Side-Bisector Relation

Let ABC be a triangle as in the following figure, a, b and c the length of the sides BC, AC and AB, a_i and a_e the length of internal (AD) and external (AE) bisectors of angle A, and b_e the length of the external angle bisector (BF) of angle B. The objective is to express a in terms of a_i, a_e and b_e.

This problem was first posed by Heymann in [11]. He wanted to determine if, given *general* values of the three angle bisectors, can one draw the triangle using only a compass and a ruler? This is possible if and only if the expression involving a, a_i, a_e and b_e is of degree 2^m in a for some integral m (see corollary 2 of Theorem 5.4.1 in [10]). This example was again posed in [8] where they solved this problem, and we have presented it exactly in the same way.



It is a standard result of Euclidean geometry that:

$$a_i^2 = \frac{cb(c+b-a)(c+b+a)}{(b+c)^2}$$
$$a_e^2 = \frac{cb(a+b-c)(c-b+a)}{(c-b)^2}$$
$$b_e^2 = \frac{ac(a+b-c)(c+b-a)}{(c-a)^2}$$

Hence, it is easy to express the length of the bisectors in terms of the length of the sides. The challenge is to express the length of the sides in terms of the length of these three bisectors. In principle, this is a simple elimination problem, i.e., if we can eliminate the variables b and c from any two equations, then we can plug the expressions for these two variables into the third equation to obtain an expression for a in terms of only a_i, a_e and b_e.

This can be achieved by computing the resultant of these three polynomials w.r.t. the two variables, b and c. To this effect, let us first represent these equations as polynomials by transforming them to the following:

$$q_1 = a_i^2(b+c)^2 - cb(c+b-a)(c+b+a)$$
$$q_2 = a_e^2(c-b)^2 - cb(a+b-c)(c-b+a)$$
$$q_3 = b_e^2(c-a)^2 - ac(a+b-c)(c+b-a)$$

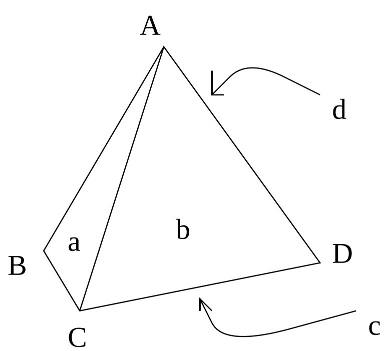
The objective is to eliminate b and c without any constraints on them. We discuss the trace of our algorithm on this set in the next paragraph.

First the Dixon polynomial was computed, followed by the Dixon matrix (M) of this set which turned out to be 13×14. Solving the matrix equation $M\bar{x} = \bar{0}$ resulted in a vector whose component corresponding to the monomial $b^0c^0 = 1$ is 0 and this took 198 seconds. Hence the condition in step (2) of the algorithm is true. Now Gaussian elimination was performed. Gaussian elimination took 284 seconds. The product of the pivot elements of the matrix after Gaussian elimination is the necessary condition that a, a_i, a_e and b_e must satisfy for any triangle ABC. The total computation took 501 seconds. After removing extraneous factors, the smallest necessary condition was obtained and it contains 330 terms. The result is the same as the one reported in [8]. Since the expression is of degree 20 in a, the triangle cannot be constructed using a compass and a ruler, given a_i, a_e and b_e (see corollary 2 of Theorem 5.4.1 in [10]).

Gao and Wang in [8] reported that they solved the problem by successively eliminating b and c using a combination of pseudo division and Sylvester's resultant computation. They took about 19 hours on their implementation in lisp on a SUN 4 workstation. Our algorithm took less than 17 minutes on a SUN 4, and less than 9 minutes on a SPARC-station 10. □

Example 2: Maximum Volume of a Tetrahedron

Consider a tetrahedron as below:



The objective is to determine the maximum volume that a tetrahedron can have, given that the squares of the areas of the four faces, ABC, ACD, BCD and ABD are a, b, c and d respectively.

It has been established in [9] that if there exist parameters x, y, z and w which satisfy the following four equations:

$$\begin{aligned} yz + zw + wy - a &= 0 \\ zx + xw + wz - b &= 0 \\ wx + xy + yw - c &= 0 \\ xy + yz + zx - d &= 0 \end{aligned}$$

then the tetrahedron is an orthocentric tetrahedron and hence the one with the maximum volume for these surface areas. Moreover, the square of the volume (T) of this tetrahedron is:

$$T = \frac{2}{9}(xyz + yzw + zwx + wxy)$$

How does one get the value of T purely in terms of a, b, c and d? This problem translates to eliminating x, y, z and w from the above mentioned five equations, i.e., computing the resultant of the following five polynomials:

$$\begin{aligned} q_1 &= yz + zw + wy - a \\ q_2 &= zx + xw + wz - b \\ q_3 &= wx + xy + yw - c \\ q_4 &= xy + yz + zx - d \\ q_5 &= 2(xyz + yzw + zwx + wxy) - 9T \end{aligned}$$

with respect to x, y, z and w. It turns out that any of these variables x, y, z and w being zero is a degeneracy, hence we can work under the constraint $\mathcal{C} = x \neq 0 \wedge y \neq 0 \wedge z \wedge w \neq 0$. We now discuss the trace of the program.

The Dixon matrix was set up and it was found to be of dimension 16×16. Solving the equation $M\bar{x} = \bar{0}$ took 10 seconds and it was found that the component of $\bar{x}$ corresponding to the monomial x was 0. So the condition of step 3 is satisfied and Gaussian elimination was performed which took 84 seconds. The necessary condition for the existence of an affine zero satisfying $\mathcal{C}$ was computed as the product of the pivots entries of the matrix which was obtained after Gaussian elimination. It was found that there was no extraneous factor in the necessary condition, hence it is the smallest necessary condition and it contained 434 terms. The total time taken was 110 seconds. □

5 Advantages of Dixon resultants

In our experimentation with this technique, we found that this technique is faster than successive Sylvester resultant computation for computing resultants for both cases - (1) two polynomials and single variable to be eliminated, and (2) more than two polynomials with more than one variable to be eliminated. In the case when the Dixon matrix is singular, this technique was also found to be faster than perturbation techniques. The reasons are as follows.

1. **Smaller Determinant :** In the one variable case, it was seen in section 2.1 that the resultant is the determinant of an $max(m, n) \times max(m, n)$ Dixon matrix (where m and n are the degrees of the two polynomials). This is a much smaller matrix than the $(m+n)\times(m+n)$ matrix that one gets using Sylvester's formulation. The smaller matrix leads to **reduced time**.

2. **Uniform Approach :** When more than one variables need to be eliminated, Dixon's method works better because all polynomials and variables are treated uniformly. The method particularly works best in the case of problems (or polynomials) which are symmetric in the variables which need to be eliminated. One possible reason is that Dixon's method adopts a symmetric approach and a single Dixon matrix is set up for all the polynomials together. This is in contrast to the successive resultant computation techniques which eliminate variables one by one, and compute numerous intermediate resultants before successively computing the resultant of the whole set. This ordering among the variables breaks the symmetry of the problem because of which huge intermediate polynomials before the (relatively small) resultant is obtained. This usually turns out to be costly. Dixon's method avoids such intermediate polynomials and hence is **much faster** and also **saves space**. This is the main reason that our algorithm is able to prove theorems and derive identities substantially faster than other methods.

3. **Faster than Perturbation :** Perturbation usually is an expensive operation as opposed to the approach presented in this paper. Dixon resultants (and in fact most resultant formulations in general) are sensitive to the number of variables, the degree of each polynomial and also the distribution of variables (a variable occurring in a few polynomials is better than it occurring in a lot of polynomials). If the Dixon matrix is singular, then a generic perturbation (a la [3]) can also be performed to obtain a projection operator which is not identically zero. However, the perturbation variable is introduced at the highest possible degree, in every polynomial. Our method, on the other hand, avoids perturbation in many cases. In the proposed method, the projection operator is extracted from the Dixon matrix of the original system, **without any extra computation**. This results in a method which is more efficient than perturbations and this is reflected in all the examples of this paper.

4. **Automatic Method :** Methods based on variable orderings or which employ successive elimination also suffer with human intervention. A good ordering must be specified or the order in which elimination is performed must be given. This seems unavoidable since the time taken by successive elimination techniques is sensitive to the variable ordering used. Dixon's method, on the other hand, does not eliminate the variables in any particular order. Instead, this method directly computes the resultant; thus being **fully automatic**. In fact, never once did we have to interfere during the proofs of the geometry theorems and algebraic identities mentioned in this paper.

6 Empirical results

We present more examples, and give the following characteristics about each example in table 1:

(a) Sing : Whether the Dixon matrix was singular (s) or nonsingular (n).

(b) Terms : Number of terms in the smallest necessary condition for affine zeros.

(c) Dix : Time taken by an implementation of our algorithm in MAPLE on a SPARCstation 10.

(d) Gröb : Time taken by MACAULAY system [1] on a SPARCstation 10 for computing the Gröbner basis using block ordering where the first block contains all the variables and the second all the parameters. Variables among the same block are degree ordered, and across the blocks, they are lexicographically ordered.

(e) Mac/GCP : Time taken to compute the Macaulay resultant (GCP when Macaulay matrix is singular [3]) using MAPLE on a SPARCstation 10.

A (*) indicates that either the program went on for more than a day, or it ran out of space.

We tried successive resultant computation using the pre-existing implementations of Sylvester's resultants and Bezout's resultants in MAPLE for various variable orderings, but the computation ran out of memory for every example. We also tried to compute the lexicographic Gröbner bases to obtain the conditions for the common zeros. Neither MAPLE, nor MACAULAY (Bayer and Stillman [1]) could compute the Gröbner basis of any of the examples in this paper. For lexicographic ordering among all the variables, MACAULAY ran for upto a day on some of these examples and then ran out of memory. When block ordering (as described in (d) above) was used, MACAULAY successfully terminated after a long time on three of the examples (2, 4 and 5), but the remaining two (1 and 3) went on for a day and still did not terminate.

Example	Sing	Terms	Dix	Gröb	Mac/GCP
1	s	330	501s	*	*
2	s	434	110s	585s	*
3	n	2424	9.6s	*	39s
4	s	990	154s	15429s	*
5	n	781	2.4s	2207s	850s

Table 1: Comparision of Various Methods

Attempt was made to compute the Macaulay resultant (GCP in case the Macaulay matrix was singular [3]) on MAPLE, but none of the GCP computations (examples 1, 2 and 4) ever finished. In all those cases, the computation ran for upto a day before running out of space. The resultant for the examples in which GCP was not required (examples 3 and 5) successfully terminated, but took longer time than our method.

For all the examples in which the Dixon matrix is singular, perturbation techniques ([3]) were also tried. One can perturb the polynomials so that they become ndegree and the Dixon matrix is no longer singular. This technique also failed for each example in this paper because the Dixon matrix blows up after perturbation, hence resulting in a substantially larger Dixon matrix with larger polynomial entries. The determinant computation for these matrices ran out of space for all the examples on MAPLE.

For each example, the variables are one or more of x, y and z. The polynomials whose resultant needs to be computed are two or more of q_i's. The numbering of the examples follows after the previous two examples.

Example 3: Expression for the distance of the intersection of two general conics from the origin.

$$\begin{aligned} q_1 &= a_1x^2 + a_2xy + a_3y^2 + a_4x + a_5y + a_6 \\ q_2 &= b_1x^2 + b_2xy + b_3y^2 + b_4x + b_5y + b_6 \\ q_3 &= x^2 + y^2 - T \end{aligned}$$

Example 4: Conditions for perpendicular intersection of a general conic and a general circle.

$$\begin{aligned} q_1 &= a_1x^2 + a_2xy + a_3y^2 + a_4x + a_5y + a_6 \\ q_2 &= x^2 + y^2 + b_1x + b_2y + b_3 \\ q_3 &= \frac{dq_1}{dx}\frac{dq_2}{dx} + \frac{dq_1}{dy}\frac{dq_2}{dy} \end{aligned}$$

Example 5: Conditions for the following four equations to have a common solution.

$$\begin{aligned} q_1 &= a_1x + a_2y + a_3z + a_4 \\ q_2 &= b_1xy + b_2yz + b_3zx + b_4 \end{aligned}$$

$$q_3 = c_1xyz + c_2$$
$$q_4 = d_1xyz + d_2x + d_3y + d_4z$$

7 Conclusion

Dixon's method for simultaneously eliminating several variables is discussed. The method is extended for the case when Dixon matrix is singular. An algorithm is given to extract a nonzero projection operator for a subclass of the systems of multivariate polynomials from its singular Dixon matrix. This algorithm avoids perturbation.

A great deal of work needs to be done for further investigating Dixon's method. In particular, there is a need to further understand Dixon's method in the general case. Determinant computations of matrices with polynomial entries are a major bottleneck in the method. Fast methods based on interpolation, similar to those in [13], need to be investigated in order to make Dixon's method more widely applicable.

Acknowledgements : We would like to thank Lakshman Y. N. for useful discussions.

References

[1] Bayer D., Stillman M., *Macaulay User's Manual*, Cornell University, Ithaca, NY.

[2] B. Buchberger, Gröbner bases: An Algorithmic method in Polynomial Ideal theory, in *Multidimensional Systems Theory,* N.K. Bose, ed., D. Reidel Publishing Co., 184-232, 1985.

[3] Canny J., Generalized Characteristic Polynomials, *Journal of Symbolic Computation*, 1990, 241-250.

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

[5] Chionh E., *Base points, resultants, and the implicit representation of rational Surfaces.* Ph. D. thesis, Dept. Comp. Sci., Univ. of Waterloo, 1990.

[6] Chou, S.-C., Gao, X.-S., Methods for Mechanical Geometry Formula Deriving. *Proc. ISSAC 1990*, ACM press, 1990.

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

[8] Gao, X.-S., Wang, D.-K., On the Automatic Derivation of a Set of Geometric Formulae. *Mathematics Mechanization Research Preprints* No. 8, 1992, 26-37, Academia Sinica, China.

[9] Gerber, L., The Orthocentric Simplex as an Extreme Simplex. *Pacific Journal of Mathematics* 56, 1975, 97-111.

[10] Herstein, I. N., *Topics in Algebra - Second Edition* John Wiley & Sons, Inc., USA, 1975.

[11] Heymann, W. Problem der Winkehalbierenden *Ztschr. f. Math. and Phys.* No. 35, 1890.

[12] Kapur, D., Lakshman, Y., Elimination Methods: an Introduction. *Symbolic and Numerical Computation for Artificial Intelligence* Donald, Kapur and Mundy (eds.), Academic Press, 1992.

[13] Manocha D., Canny J.F., Multipolynomial Resultant Algorithms. Tech. Report, *University of California, Berkeley*, 1991.

[14] W. Wu, On the decision Problem and the mechanization of theorem proving in elementary geometry, *Scientia Sinica,* 21, (1978), 150-172. Also in Bledsoe and Loveland, eds., *Theorem Proving: After 25 years, Contemporary Mathematics,* 29, 213-234. 1984.