THE UNIVERSITY *of* EDINBURGH

A COMPUTATIONAL CLASSIFICATION OF MULTIVARIATE POLYNOMIALS USING SYMMETRIES AND REDUCTIONS.

CARL STURTIVANT.

Ph.D.

University of Edinburgh

1983

ABSTRACT

An examination of some properties that interrelate the computational complexities of evaluating multivariate polynomial functions is presented. The kind of relationship between polynomial functions that is studied takes the form of linear transformations of the arguments and results of a polynomial function that transform it into another such function. Such transformations are a generalisation of projection (a form of reduction in algebraic complexity first introduced by Valiant, whereby variables and constants are substituted for the arguments of a polynomial function in order to transform it into another polynomial function). In particular, two restricted forms of this generalised projection are considered: firstly, those that relate a polynomial function to itself, and secondly, those that are invertable. Call these symmetries and similarities, respectively.

The structure of the set of symmetries of a polynomial function is explored, and the computationally useful members of the set identified; a technique for finding all such symmetries is presented. It is shown that polynomials related by similarity have "isomorphic" sets of symmetries, and this condition may be used as a criterion for similarity. Similarity of polynomial functions is shown to be an equivalence relation, and "similar polynomials" can be seen to possess closely comparable complexities. A fast probabilistic algorithm for finding the symmetries of a polynomial function is given.

The symmetries of the determinant and of the permanent (which differs from the determinant only in that all of its monomials have coefficients of +1), and those of some other polynomials, are explicitly found using the above theory. Fast algorithms using linear algebra for evaluating the determinant are known, whereas evaluating the permanent is known to be a #p-complete problem, and is apparently intractable; the reasons for this are exposed. As an easy corollary it is shown that the permanent is not preserved by any bilinear product of matrices, in contrast to the determinant which is preserved by matrix multiplication. The result of Marcus and Minc, that the determinant cannot be transformed into the permanent by substitution of linear combinations of variables for its arguments (i.e. the permanent and determinant are not similar), also follows as an easy corollary. The relationship between symmetries and ease of evaluation is discussed.

Acknowledgements

It is a pleasure to thank Gordon Brebner, Mark Jerrum and Les Valiant for many interesting discussions.
Thanks also to Eleanor Kerse, Dorothy McKie and Kate Duncan for seeing through a difficult typing job.

CONTENTS.

INTRODUCTION

(i) Preliminary Remarks

Computational complexity is ultimately concerned with finding the minimum number of steps (or amount of any other resource) required to solve a problem, and with finding methods that achieve this minimum labour. Classical "machine" complexity uses a precisely defined machine model of computation (mostly the k-tape Turing machine) to give precise definitions of "problem", "solution", "number of steps", etc. [10]. Detailed statements about the run-time behaviour of algorithms then depend upon the particular machine model. Fortunately, more general statements may well be invariant over a wide choice of "reasonable" machine models, and it is such statements that are of interest in complexity theory.

A very important example is the class P of all problems for which there exists a polynomial time algorithm (i.e. an algorithm whose run-time is bounded above by a polynomial in the size of the input [10 p.6]). P is recognised to be independent of the model of computation (in much the same way as the class of partial recursive functions is). Empirically, this is the class of problems that have practicable solutions [2 p.2], and we will refer to polynomial-time algorithms as "fast" algorithms from now on. Membership or non-membership of P is thus a key problem area in complexity, and a substantial amount of work has been devoted both to attempting to answer this question for specific problems, and to developing more general techniques which may be subsequently applied to particular problems. Unfortunately, current techniques for proving that certain problems intrinsically require a certain amount of labour to solve them are crude. Thus it has proved possible to show that

some extremely hard natural problems are not in P, but for wide classes of natural problems the question remains unanswered. Two prime examples of this are the classes NP and #P, [7,10,37], which consist essentially of problems concerning respectively the existence of and number of solutions to (usually combinatorial) problems, where there is a fast algorithm to check whether or not a purported solution is in fact a solution. It is very natural to pose a problem in the form "does there exist an 'x' in structure 's'?" or "how many 'x's are there in structure 's'?", and for many problems people need to solve in practice, there is a fast algorithm to check whether or not any putative 'x' is an 'x'. The question of whether there are any problems in #P or NP that are not in P is still unresolved despite substantial efforts [10 p.181].

By comparison with the rudimentary nature of known techniques for pursuing the absolute complexity of problems, much is known about their relative complexities, in the sense that there are many results (and techniques for obtaining such) of the form, "if problem A is in P then problem B is in P". Such results are obtained by means of reductions [10,37]. A polynomial reduction (or polynomial Turing reduction) from problem B to problem A is a polynomial time algorithm for problem B involving a polynomial number of "subroutine calls" of a notional polynomial time algorithm for problem A. If such a reduction exists then the existence of a polynomial-time algorithm for problem A explicitly implies the existence of a polynomial-time algorithm for problem B. Other notions of reducibility are in common use e.g. polynomial transformations [9,10].

An important notion in relative complexity is that of complete problems [9,10]. In particular, a problem A $\in$ NP is NP-complete if

any problem in NP is polynomially reducible to A. (NP-complete problems thus constitute the "hardest" problems in NP, and if A is NP-complete, $A \in P$ implies $NP \subseteq P$). #P-completeness is similarly defined [37]. Surprisingly, large numbers of natural NP-complete problems have been found: in fact almost all natural problems in NP have either been shown to be in P or to be NP-complete [10 p.154]. In fact it would seem likely that NP and #P complete problems are intractable and do not have fast algorithms [37 p.5], indeed they seem to require strictly exponential time (i.e. time $O(2^{\alpha n^{\varepsilon}})$ for some $\alpha, \varepsilon > 0$ where n is the size of the input) empirically.

In contrast to machine complexity, algebraic complexity abandons the idea of machine models of computation, and concerns itself instead with choosing an appropriate algebra for a problem and investigating how efficiently that problem can be solved within that algebra.

Despite the relatively "clean" way of expressing a problem afforded by algebraic complexity, known algebraic techniques for proving a problem intrinsically hard are almost as crude as those used in machine complexity. Relative complexity, however, is dealt with in a more satisfactory manner: different notions of reduction from those employed in machine complexity are used - usually projection (introduced by Valiant [34] and subsequently used in [12,28,35]). Projection consists of a series of substitutions that convert one expression into another. As this is such a simple notion of reduction we may hope that some negative complexity results will be proved in the future.

In algebraic complexity a problem is expressed within some algebra as follows. Firstly, the algebra is chosen so that its operations are sufficient to solve the problem. Clearly, an

expression in an algebra (e.g. a polynomial over some field) has a fixed number of arguments (indeterminates), whereas the problem may have inputs of an infinite number of different sizes. Thus, the problem is represented by an infinite family of expressions within the algebra, one for each input size, indexed by the number of arguments. For example the travelling salesman problem [10 p.211] could be represented algebraically in the semi-ring $(\mathbb{Q}^+, \min, +)$ (with minimisation as formal addition, and addition as formal multiplication) as the family of expressions

$$TSP = \{TSP_{n2} \mid n \in \mathbb{N},\ TSP_{n2} = \sum_{\sigma \in H_n} \prod_{i=1}^{n} x_{i\sigma(i)}\}$$

where H_n is the set of all permutations of $(1,2,\ldots,n)$ that consist of a single cycle (i.e. Hamiltonian circuits), and $[x_{ij}]$ is the $n \times n$ matrix of edge weights in the n-vertex graph. (Clearly, in this algebra $w_\sigma \stackrel{\Delta}{=} \prod_{i=1}^{n} x_{i\ \sigma(i)}$ is the sum of the weights on the Hamiltonian circuit σ, and $\sum_{\sigma \in H_n} w_\sigma$ is the minimum such sum.)

There are two important measures of the computational complexity of such an algebraic problem. For each expression (i.e. for each number of arguments) in the family constituting the problem, there is one (or more) minimum sized formula(e) in the algebra that represent(s) the same function. (By the size of a formula is meant the number of operations from the algebra in it.) The "formula size" of the problem is then defined as that function that gives the size of the minimal formula(e) as a function of the number of arguments.

A more basic measure of complexity is "circuit size". Clearly, a formula may be regarded as a tree (directed) with the vertices labelled with the operations in the algebra they represent. Thus formula size does not take account of the fact that during a

computation an intermediate result may be subsequently used in more than one place. (The only way to achieve this in a formula is to have copies of the same sub-formula in all the places where that sub-result is desired, and thus the computational cost is counted repeatedly.) The proper generalisation of formula that eliminates this drawback is the "circuit" or "straight line program" [35 p.2]. A circuit is a formula in which the requirement that there be only one outgoing edge from each vertex (i.e. one use of a sub-expression) is dropped. Thus a circuit is a directed acyclic graph (DAG), with the vertices labelled with the operations in the algebra they represent. Circuit size is then defined in much the same way as formula size: the "circuit size" of a problem is that function that gives the size of the minimal (in terms of number of operations in the algebra) circuit for the problem as a function of the number of arguments.

Now consider algebraic complexity over a field F (or ring R). A problem is now a family of polynomial functions with coefficients in F, for example the determinant family DET = $\{DET_1, DET_4, DET_9 \ldots\}$ where DET_m is the determinant polynomial for a $\sqrt{m} \times \sqrt{m}$ matrix. (Note that, in general, circuit and formula size may depend on F.) Hyafil [11] has shown that there is a constant α such that any polynomial of circuit size C and degree d has a formula of size less than or equal to $C^{\alpha \lg d}$. Since it is natural to consider only families of polynomials whose degrees are p-bounded (i.e. bounded above by a polynomial in the number of arguments), this means that the gap between formula size and circuit size is at most "quasi-polynomial", and is much less than the empirically exponential gaps that are currently unproven, between other important classes (for example P

and NP).

The analogue of P in algebraic complexity is p-circuit (pC). This is defined as all families of polynomials of p-bounded degree with p-bounded circuit size; p-formula (pF) is defined analogously [34]. Since a formula is a circuit, pF $\subseteq$ pC.

The analogue of the classes NP and #P in algebraic complexity is the class p-definable (pD) [34]. This is defined as all families of polynomials of p-bounded degree, where essentially the coefficient of any monomial is easily computed (i.e. there is a small formula) as a function of the powers of each variable in the monomial. The relationship of this class to P and NP will be examined later.

If $f_i(x_1,\ldots,x_i)$ and $g_j(y_1,\ldots,y_j)$ are polynomials over the field F (or ring R) then f_i is a <u>projection</u> of g_j if there is a substitution σ: $\{y_1,\ldots,y_j\}$ -> F $\cup$ $\{x_1,\ldots,x_i\}$ such that $f_i(x_1,\ldots,x_i) = g_j(\sigma(y_1),\ldots,\sigma(y_j))$. A family $f = \{f_i\}$ is a projection of a family $g = \{g_j\}$ if for all $f_i \in f$ there is a $g_j \in g$ such that f_i is a projection of g_j. The fact that one family is a projection of another is not of immediate computational relevance, since it may relate members of one family to members of the other with relatively enormous numbers of arguments. To be of relevance as a notion of reducibility analogous to polynomial reducibility in machine complexity, we need to impose a bound on this growth. The result is p-projection [28,35], which is defined as follows: a family f is a p-projection of a family g if there exists a constant k such that all $f_i \in f$, f_i is the projection of some $g_j \in g$ with $j \leq i^k$. Clearly, if g is in pC and f is a p-projection of g then f is in pC.

Compared to polynomial reduction in machine complexity, p-projection is a very strict form of reduction, of such simplicity that one would expect very few families to be interrelated. Thus, whilst any two problems in P are trivially related by polynomial reductions, the same is not true for pC via p-projections, so that the class of complete problems for pC via p-projection is not trivially equal to the whole of pC.

A family f is said to be universal for a problem class A (via p-projection) if any family in A is a p-projection of f. If, in addition $f \in A$, then f is complete for A via p-projection. Valiant [34] has shown that the determinant family DET is universal for pF; in fact that every polynomial of formula size s is the projection of an $(s+2) \times (s+2)$ determinant. The $n \times n$ determinant has a known formula of size $2^{O(\lg^2 n)}$ [11] and is not thought to be in pF. Nevertheless, the above result is essentially a universal means of expressing a sub-exponential formula for a polynomial, where such exists. (The best known lower bound for the formula size of the $n \times n$ determinant is $O(n^3)$ [14].)

The following short argument is due to Valiant [34]. Define a "quasi-polynomial" in n to be two to the power of any polynomial in log n. The determinant then has at most quasi-polynomial formula size. Using the result of Hyafil (given earlier) relating circuit and formula size by a quasi-polynomial bound and that of Valiant that any formula can be expressed as a determinant of virtually the same size, we can see that, firstly, the class of problems with qp-bounded (quasi-polynomial bounded) formula size is the same as the class of problems with qp-bounded circuit size; and thus, secondly, that a polynomial is in "qp-circuit" iff it is a

"qp-projection" of the determinant. We therefore have the result that the determinant is a universal device for sub-exponential computations in algebraic complexity. Whether or not the determinant is complete for pC via p-projection is unknown.

A problem that is complete for pD is said to be algebraically complete. Since pD is loosely the algebraic analogue of NP and #P, this is empirically an assertion of intractability i.e. of exponential circuit and formula size. The permanent family PER = $\{PER_1, PER_4, PER_9 \ldots\}$ is algebraically complete (over fields of characteristic $\neq 2$) [34]. (Note that the permanent has the same set of monomials as the determinant, but that the coefficient of each monomial is always +1 in the permanent.) Evaluating the permanent (over fields of characteristic $\neq 2$) has also been shown to be #P-complete in machine complexity [33]. (The travelling salesman problem (TSP), mentioned earlier, is also algebraically complete over the relevant semi-ring, thus explaining its apparent intractability. Indeed, there are many known algebraically complete problems [12,36].)

The smallest known formula for the n×n permanent is of size $0(n^2 2^n)$ [21,24]; thus the n×n permanent is the projection of the m×m determinant where $m \leq 0(n^2 2^n)$. The smallest known circuit for the permanent is of size $0(n2^n)$ [23, Appendix 2]. Empirically, it is considered likely that the circuit size of the permanent is exponential, since the contrary would imply faster algorithms for NP and #P complete problems in machine complexity, and smaller circuits for p-definable polynomials in algebraic complexity. (Clearly, if the permanent is a "qp-projection" of the determinant then every "qp-definable" polynomial would have qp-bounded formula

size.)

Algebraic complexity has been pursued in many algebraic systems e.g. boolean algebra [28], various semi-rings [13], various fields [5,34,35]. Of these, a complexity theory based on boolean algebra has the property of modelling discrete computations simply and accurately, and is thus the algebraic analogue of Turing machine complexity. It is not surprising therefore, that in this domain of computation the algebraically complete problems are essentially the NP-complete problems of machine complexity [28].

Boolean complexity is essentially equivalent to complexity over the finite field GF(2), since any boolean operation may be expressed as a small constant number of arithmetic operations in GF(2) and vice-versa. Formula and circuit size are therefore altered by at most a constant factor when translating between the two algebras, and the complexity of an algorithm expressed in one of the two systems is thus invariant upon changing to the other. The methods used in boolean complexity are usually combinatorial in nature and often very "syntactic". These methods have not yet proved powerful enough to answer the really difficult questions in complexity, such as "pD=?pC". The boolean/GF(2) case does seem to illustrate the inconvenient properties possessed by finite fields in this context.

Studies of semi-rings by Jerrum and Snir [13] have illustrated the fact that circuits in such "loose" algebraic systems fail to capture the subtleties of efficient computation which are possible in more restricted structures such as rings and fields. (They define a semi-ring to be a domain of computation in which there are two operations '+' and 'x', both associative and commutative, with

'x' distributing over '+'.) In particular, they exhibit several semi-rings within which the optimal circuits for computing various natural functions are the obvious ones. Such optimal circuits are, however, sub-optimal in some of the more restricted domains of computation. An example is matrix multiplication, which they show to have optimal circuit size $O(n^3)$ in some semi-rings, but the best result known to date is $o(n^{5/2})$ over any field [8].

Jerrum and Snir also exhibit semi-rings in which the permanent requires $O(n2^n)$ arithmetic operations, and in which the spanning tree polynomial [13 p.891] requires $(4/3)^{O(n)}$ operations. An interesting result due to Kirchoff [9 p.34,19] is that, over a field, the n×n spanning tree polynomial can be written as an n×n determinant whose entries are linear combinations of the indeterminates: thus showing the spanning tree polynomial to be in pC over a field. This remarkable fact illustrates the subtlety of computation over fields, to which we will therefore confine ourselves. No similar speedup is known for the permanent when going over to fields (of characteristic $\neq 2$). Indeed, the existence of such a speedup seems most unlikely since it would show pC = pD over fields (a close algebraic analogue of "P = #P"!).

Given the universal nature of the determinant for computing functions with small formula size, and the close connection between formula size and circuit size, it is reasonable to conjecture that the determinant (and linear algebra in general) in some sense constitutes a universal technique for fast algebraic algorithms. This conjecture is further strengthened by the ubiquity of linear algebra as a computational tool for all kinds of algebraic problems. Thus it is instructive that linear combinations of the indeter-

minates are used to write the spanning tree polynomial as a determinant, and this suggests that projection should be generalised to allow substitutions of such linear combinations. Such a generalisation of projection is further motivated by the observation that whilst the determinant is not known to be complete for pC via p-projection, it may well become so via some slightly more general notion of projection. Additionally, in boolean complexity projections have always been defined more generally to include substitutions of the negations of variables [28]. Re-interpreting this in the field GF(2) gives projections in which any constant may be added onto any substituted variable. Skyum [27] has shown that allowing such negations can make a drastic difference to the power of projections in boolean complexity by exhibiting a family of functions whose projective power is increased exponentially by such a generalisation of the notion of projection.

Bearing in mind the above remarks we therefore define a generalised projection from a polynomial $f(\underline{y})$ to a polynomial $g(\underline{x})$ (where $\underline{x} = \begin{pmatrix} x_1 \\ \vdots \\ x_n \end{pmatrix}$ and $\underline{y} = \begin{pmatrix} y_1 \\ \vdots \\ y_m \end{pmatrix}$ and $m \geq n$) over a field F to be an $m \times n$ matrix of constants L and an $n \times 1$ matrix of constants $\underline{l}$ such that $f(L\underline{x} + \underline{l}) = g(\underline{x})$ as functions. The existence or otherwise of such a generalised projection from f to g may be regarded as the non-emptiness (or otherwise) of the set of simultaneous solutions of the set of polynomial equations $\{\forall \underline{a} \in F^n \; f(L\underline{a} + \underline{l}) - g(\underline{a}) = 0\}$, where the entries of the matrices L and $\underline{l}$ are regarded as the variables to be solved for. Thus the existence or otherwise of generalised projections falls within the province of algebraic geometry. (The set of simultaneous roots of some set of polynomials is called an algebraic set. Algebraic geometry is concerned with

algebraic sets and their relation to polynomials [15,22].)

A good example of a general result in complexity that uses algebraic geometry is Strassen's "degree bound" [5,30], which gives a lower bound on the number of multiplications required to evaluate one or several polynomials in terms of their degree. The degree of several polynomials together can only be defined successfully in the context of algebraic geometry: discussion of this point is given in [5 p.116].

It is not surprising that algebraic geometry should be highly relevant to algebraic complexity since both are concerned with multivariate polynomials. Indeed it appears likely that the conceptual depth of algebraic geometry is necessary, especially in consideration of negative results in algebraic complexity: Strassen's degree bound is an example. It is no accident that the more combinatorial approaches to algebraic complexity, which have achieved great success in constructing algorithms and reductions, have had far less success in producing significant absolute negative results.

Unfortunately, problems in algebraic geometry over finite fields or over the rationals (or any algebraic extension thereof) can be virtually intractable, owing to their strong number theoretic flavour. Also, algebraic geometry is usually pursued over algebraically complete fields to avoid problems of "missing roots".

In the context of complexity it is natural to choose a field that contains the natural numbers so that various combinatorial "counting" polynomials (such as the permanent and the spanning tree polynomial) are included in the class of p-definable functions whilst

retaining their usual combinatorial interpretation. In this way pD becomes a direct algebraic analogue of #P (in fact pD becomes the restriction of #P when only algebraic algorithms are allowed). This enables direct concrete interpretation of any algebraic results. In the light of the remarks of the previous paragraph, we therefore need to choose an algebraically complete field of characteristic zero. Circuit and formula sizes are the same (for a given polynomial) over the algebraic closure of the rationals as over the complex numbers [appendix 1]. For simplicity, it seems natural, therefore, to choose the complex numbers.

(ii) Specific Discussion

There are many instances in complexity of a pair of problems with very similar problem specifications, one of which is in P, the other being NP or #P complete. Examples of such pairs can be found in [10 p.79]. In each case, the member of the pair for which there exists a fast algorithm possesses some structural features which are absent in the case of the other member. It is these structural features that enable algorithms more efficient than an exhaustive exponential search to be found. Indeed, the apparent absence of any such helpful features in the case of all known NP and #P complete problems is strong empirical evidence of their intractability. Unfortunately, it follows that any future proof that purports to distinguish NP and #P complete problems from tractable problems will have to be able to discriminate between problems with very similar specifications. One difficulty in doing this is to decide precisely what structural features of a problem could potentially give rise to a fast algorithm, and then proving that such do not exist.

An algebraic example of this phenomenon is the pair of polynomial families, the determinant and the permanent. (Let X be an n×n matrix. The permanent of X is defined as

$$\text{per } X = \sum_{\sigma \in S_n} \prod_{j=1}^{n} X_{j\sigma(j)}$$

where S_n is the group of all permutations of (1,2,...,n). Note that the permanent is the same as the determinant except that all terms have positive sign.) As mentioned previously, the permanent is algebraically complete [34,35], and computing it is a #P-complete problem [33], whereas fast algorithms are known for computing the

determinant [5]. These fast algorithms rely upon structural features (or "symmetries") of the determinant that the permanent apparently does not possess. A good example is Gaussian elimination which relies upon the identity $det(AB) = det(A).det(B)$ to transform a given matrix (whose determinant is required) into a triangular matrix, whilst changing the determinant only by a known factor. It is thus natural to ask whether the permanent possesses any such symmetries, and if so how they can be used to assist faster computation.

It has already been mentioned that the spanning tree polynomial may be written as a determinant in which linear combinations of the indeterminates have been substituted. A similar relationship exists between the wrapped convolution and the pairwise product. Let $\underline{x}$, $\underline{y}$ and $\underline{z}$ be $n \times 1$ column matrices. $\underline{z}$ is the Hadamard product (or pairwise product) of $\underline{x}$ and $\underline{y}$ if $\forall j \; z_j = x_j y_j$. Also $\underline{z}$ is the wrapped convolution of $\underline{x}$ and $\underline{y}$ if $\forall j \; z_j = \sum_{i=0}^{n-1} x_i y_{j-i \bmod n}$. An efficient technique for evaluating a wrapped convolution [2 p.254] relies upon transforming the convolution into the Hadamard product by means of the discrete Fourier transform. In both the case of the Spanning Tree polynomial and Wrapped Convolution, the polynomial in question is transformed into a polynomial that is easy to evaluate, by means of an invertable linear transformation of the indeterminates. Such a transformation is clearly a very restricted form of generalised projection, where the number of indeterminates in the polynomials being so related is constrained to be equal. Call polynomials related by such a transformation "similar polynomials". Whether the permanent is "similar" to something easy to compute is clearly an open question.

The essential structural feature of the determinant that makes it easy to evaluate is its preservation of matrix multiplication. Thus, given that the permanent is a multilinear polynomial of a very similar form to the determinant, one could regard the preservation of a bilinear "product" of a pair of matrices as a "symmetry", and ask whether the permanent possesses such a product and how such a product (if it exists) may assist fast evaluation (i.e. is there an operation "o" with $per(AoB) = per(A).per(B)$ and $(AoB)_{ij} = \sum_{\alpha\beta\gamma\delta} \psi_{ij\alpha\beta\gamma\delta} A_{\alpha\beta} B_{\gamma\delta}$ where the $\psi_{ij\alpha\beta\gamma\delta}$ are constants).

Alternatively, the determinant's preservation of matrix multiplication can be regarded as an example of a phenomenon of a more general kind; namely the existence of linear combinations of the indeterminates of a polynomial that when substituted in the place of the original indeterminates yields the original polynomial (aside from a constant factor). Thus in $det(AX) = det(A).det(X)$ the constants in A are to be regarded as giving rise to a linear transformation of the inputs X, that preserves the determinant of X for all X aside from a constant factor (det(A)). The fact that in the case of the determinant the inputs are in the form of a matrix and the linear combinations are then generated by matrix multiplication is to be regarded as irrelevant to the general definition of the phenomenon. This is essentially the viewpoint used by the Gaussian elimination algorithm for evaluating determinants.

Suppose a polynomial $P(\underline{x})$, (where $\underline{x}$ is the column matrix of the indeterminates $x_1,\ldots,x_n$) has a symmetry analogous to those of the determinant, whereby taking a certain linear combination of the variables before evaluating P only alters the result by a constant factor plus a constant additive term,

i.e. $\forall \underline{x} \quad k.P(T\underline{x} + \underline{t}) + k' = P(\underline{x})$,

(where T is an n n matrix of constants, $\underline{t}$ is an n×1 matrix of constants and k,k' are constants). Then P could be evaluated at $\underline{a}$ by evaluating P at $T\underline{a} + \underline{t}$, multiplying by k and adding k'. If $T\underline{a} + \underline{t}$ has more entries equal to zero than $\underline{a}$ then there may be some computational advantage to this scheme as compared to evaluating $P(\underline{a})$ directly.

In order to construct such a scheme whereby P can be evaluated advantageously at any point $\underline{x}$, it must be possible for the symmetry $(T,\underline{t})$ chosen to depend upon $\underline{x}$, in order to introduce zeros into $T\underline{x} + \underline{t}$. (In practice several successive transformations may be made, introducing successively more zeros whilst preserving those previously present. Such a scheme constitutes a Gaussian elimination style algorithm for evaluating P.) Intuitively, nearby points $\underline{x}^{(1)}$ and $\underline{x}^{(2)}$ will need nearby symmetries $(T^{(1)},\underline{t}^{(1)})$ and $(T^{(2)},\underline{t}^{(2)})$, in order for $(T^{(1)}\underline{x}^{(1)} + \underline{t}^{(1)})$ and $(T^{(2)}\underline{x}^{(2)} + \underline{t}^{(2)})$ to have the same entries equal to zero. Thus, in order for this to be possible, it is necessary for some of the symmetries of P to form a continuum, and these "continuous" symmetries of P will include all of the computationally useful symmetries of P. Again intuitively, this continuum of symmetries will have a dimension (the number of "degrees of freedom" involved in specifying a particular symmetry), and this dimension must exceed the number of zeros to be introduced into all inputs. Thus for a family of polynomials (such as the determinant or permanent), this "dimension" must be O(n) for n inputs $(x_1,\ldots,x_n)$, otherwise the fraction of entries that can be transformed to zero by using symmetries will be asymptotically zero, and there will be no asymptotic gain in the complexity of evaluation. Clearly this

criterion is satisfied by the determinant.

Suppose sets of polynomials $\{P_i(\underline{x})\}$ and $\{Q_i(\underline{x})\}$ $(1\leq i\leq m)$ are related by linear transformations in a way analogous to the relation between the wrapped convolution and the Hadamard product (i.e. they are similar)

i.e. $\forall \underline{x},i\ P_i(\underline{x}) = \sum_{j=1}^{m} S'_{ij}Q_j(S\underline{x} + \underline{s}) + s'_i$, (or $\underline{P}(\underline{x}) = S'\underline{Q}(S\underline{x} + \underline{s}) + \underline{s}'$)

(where S',S are non-singular m×m and n×n matrices respectively, and $\underline{s},\underline{s}'$ are n×1 and m×1 matrices of constants respectively). Then $\{P_i\}$ could be evaluated at $\underline{x}$ by evaluating $\{Q_i\}$ at $S\underline{x} + \underline{s}$ and taking a linear combination of the results. Call such a scheme a transformation style algorithm for evaluating $\{P_i\}$. (In the case of the wrapped convolution $\underline{W}(\underline{x},\underline{y})$ and the Hadamard product $\underline{H}(\underline{x},\underline{y})$ this is $\underline{W}(\underline{x},\underline{y}) = F^{-1}\underline{H}(F\underline{x},F\underline{y})$ where F is the discrete Fourier transform matrix [2].) It is natural to allow linear combinations of the "outputs" if there is more than one, rather than just a constant scalar multiplier; this generalisation will be carried over to generalised projections and symmetries (which are just "self-similarities"). The importance of similarity is its invertability: if $\underline{P}$ and $\underline{Q}$ are similar polynomials then a symmetry of $\underline{Q}$ will give rise to a symmetry of $\underline{P}$ as follows:- first transform $\underline{P}$ into $\underline{Q}$, then apply the symmetry of $\underline{Q}$, then transform $\underline{Q}$ back to $\underline{P}$; the composition of the transformations used to perform these steps is clearly a symmetry of $\underline{P}$. Thus similar polynomials will have closely related sets of symmetries, and this may be used as a criterion for similarity. (This relationship between symmetries and similarity is a consequence of the action of symmetries upon the transformations giving rise to similarity. Obviously this is not restricted to similarity: the symmetries of a polynomial will

also act upon any generalised projection to another polynomial, yielding other projections that relate the same polynomials.)

This thesis is concerned with exploring the structure of the set of symmetries of a polynomial function, and identifying the computationally useful symmetries of any such function. The relationships between similarities, symmetries and generalised projections are also considered. Chapter 2 is concerned with the development of definitions and results that go to make up a general theory of such matters for any polynomial function, including a method of obtaining all of the computationally useful symmetries. The remainder of the thesis is concerned with applications of this theory to specific polynomials and the consequences for complexity of the theory, along with a general examination of some more algorithmic aspects of symmetries, similarities and projections, and their relation to linear algebra.

(iii) Notations and Conventions

Throughout let $\underline{x} = \begin{pmatrix} x_1 \\ \vdots \\ x_n \end{pmatrix}$ and $\underline{P}(\underline{x}) = \begin{pmatrix} P_1(\underline{x}) \\ \vdots \\ P_m(\underline{x}) \end{pmatrix}$

where $P_i(\underline{x})$ is a polynomial in the indeterminates $x_1,\ldots,x_n$ for each i in the range $1,\ldots,m$. Unadorned uppercase letters will denote square matrices unless otherwise specified; underlined lowercase letters will denote column matrices; the identity matrix and the column matrix of all zeros will be denoted by I and $\underline{0}$ respectively. The sizes of matrices, if not explicitly stated, will be apparent from the context. The transpose of a column matrix $\underline{a}$ will be denoted by $\underline{a}^T$. The field of numbers will be the complex numbers throughout. Further notation and conventions will be introduced where necessary.

2. CENTRAL DEFINITIONS AND RESULTS

(i) Basic Definitions and Theorems

The transformations of the "inputs" and "outputs" of a polynomial that will be considered in this chapter will consist of linear transformations followed by shifts of origin. For compactness, the following definition will be adopted.

Defn. 2.0 A linear affine transformation $\bar{A}$ is a pair $(A,\underline{a})$ of matrices of the same height, where A is square. $\bar{A}$ is considered as a map from one vector space to another, followed by a shift of origin i.e. $\forall \underline{x}\ \bar{A}\underline{x} = A\underline{x} + \underline{a}$.

Scalar multiplication, addition and product (composition) of linear affine transformations are defined in the obvious way, and possess all the algebraic properties of the corresponding operations for square matrices, except for the distributive property and commutation with scalars. (The convention of using the same letter for an affine transformation and the corresponding matrices as above will be adhered to throughout.)

Thus $\bar{A} + \bar{B} = (A + B,\ \underline{a} + \underline{b})$ and

$\bar{A}\bar{B} = (AB, A\underline{b} + \underline{a})$; $\bar{I} = (I,\underline{0})$ is the identity transformation and $\bar{0} = (0,\underline{0})$ is the zero transformation; a (left) scalar multiplication is $\lambda\bar{A} = (\lambda A, \lambda\underline{a})$.
The inverse of $\bar{A}$ is $\bar{A}^{-1} = (A^{-1}, -A^{-1}\underline{a})$ and exists iff $\det A \neq 0$, when $\bar{A}$ is said to be non-singular.

Defn. 2.1 $\bar{T}'$ is a linear affine symmetry of $\underline{P}(\underline{x})$ if
$\exists \bar{T}''\ \forall \underline{x}\ \bar{T}''\underline{P}(\bar{T}'\underline{x}) = \underline{P}(\underline{x})$.

This corresponds to the notion of being able to compute $\underline{P}$ at $\underline{x}$ by

computing $\underline{P}$ at $\bar{T}'\underline{x}$, and there existing a transformation $\bar{T}''$ to transform the results of the latter computation back to the desired results. We note that it is $\bar{T}'$ that is important for Gaussian elimination type operations.

Homogeneity imposes a constraint upon the form of affine symmetries, and since most "natural" families of polynomials in complexity are homogeneous and of the same degree (for a given number of arguments), e.g. matrix multiplication, convolution, adjoint matrix etc., we adopt the following definition:-

Defn. 2.2 $\underline{P}$ is homogeneous and of degree d iff $\forall \underline{x}\ \underline{P}(\lambda\underline{x}) = \lambda^{d}\underline{P}(\underline{x})$, for all scalar λ.
i.e. each P_i is homogeneous and all are of the same degree d.
Homogeneity is clearly an affine symmetry.

It is natural to consider the question, "under what conditions do the collection of symmetries of a family of polynomials constitute a group under composition of linear affine transformations?" Suppose $\underline{P}$ has a singular affine symmetry $\bar{T}'$, then clearly the symmetries of $\underline{P}$ do not constitute a group since $\bar{T}'$ cannot possess an inverse. Now suppose $\underline{P}$ is independent of a certain linear combination of the indeterminates i.e. the value of $\underline{P}$ does not change if the input is translated in a certain direction (for any input). Then a symmetry of $\underline{P}$ could scale that particular linear combination of the indeterminates by a factor of zero, and thus be singular, so that the symmetries of $\underline{P}$ could not be a group. This motivates the following definition:-

Defn. 2.3 $\underline{a}$ is a translational symmetry of $\underline{P}$ iff $\forall \underline{x}\ \underline{P}(\underline{x} + \underline{a}) = \underline{P}(\underline{x})$.

This is an affine symmetry also.

The shift of origin involved in an affine transformation may upset the homogeneity of a polynomial if applied to the "inputs" or "outputs", unless it happens to coincide with a translational symmetry. The following lemma clarifies this:-

Lemma 2.1 If $\underline{P}$ is homogeneous and of degree $d \geq 2$, and has no translational symmetries, and if $\bar{A}$, $\bar{A}"$ are non-singular linear affine transformations then

$\bar{A}"\underline{P}(\bar{A}'\underline{x})$ is homogeneous iff $\underline{a}' = \underline{0}$ and $\underline{a}" = \underline{0}$

when it is of degree d.

Proof

$\bar{A}"(\underline{P}(\bar{A}'\underline{x})) = \underline{a}" + A"(\underline{P}(A'\underline{x} + \underline{a}'))$. Taylor expansion:-

$= \underline{a}" + A"(\underline{P}(A'\underline{x})$ + polynomials of degree <d).

Thus if $A"\underline{P}(A'\underline{x}) \neq \underline{0}$ homogeneity of $\bar{A}"\underline{P}(\bar{A}'\underline{x})$ implies

$\bar{A}"\underline{P}(\bar{A}'\underline{x}) = A"\underline{P}(A'\underline{x})$ = the degree d part when expanded.

By assumption A" and A' are non-singular, and $\underline{P}(\underline{x}) \neq \underline{0}$ since $\underline{P}$ has no translational symmetries.

Thus $\underline{P}(\underline{x}) \neq \underline{0} \Rightarrow \forall \underline{x}\ \underline{P}(A'\underline{x}) \neq \underline{0} \Rightarrow \forall \underline{x}\ A"\underline{P}(A'\underline{x}) \neq \underline{0}$.

Therefore $A"\underline{P}(A'\underline{x} + \underline{a}') + \underline{a}" = A"\underline{P}(A'\underline{x})$.

$\Rightarrow \forall \underline{x}\ \underline{P}(A'\underline{x} + \underline{a}') + A"^{-1}\underline{a}" = \underline{P}(A'\underline{x})$

$\Rightarrow \forall \underline{x}\ \underline{P}(\underline{x} + \underline{a}') + A"^{-1}\underline{a}" = \underline{P}(\underline{x})$. Taylor expansion:-

$\Rightarrow \forall \underline{x}\ \underline{P}(\underline{x}) + (\underline{a}'\cdot\nabla)\underline{P}(\underline{x})$ + polynomials of degree $<(d-1) = \underline{P}(\underline{x}) - A"^{-1}\underline{a}"$

where $\nabla_j = \frac{\partial}{\partial x_j}$ and the scalar product is defined without complex conjugation.

Since $(d-1) \geq 1$ and $A"^{-1}\underline{a}"$ is of degree zero, we have $(\underline{a}'\cdot\nabla)\underline{P}(\underline{x}) = \underline{0}$.

Thus $\lim_{\varepsilon \to 0} \frac{\underline{P}(\underline{x} + \varepsilon \underline{a}') - \underline{P}(\underline{x})}{\varepsilon} = \underline{0}$

$\Rightarrow \underline{P}(\underline{x} + \frac{1}{k}\underline{a}') = \underline{P}(\underline{x}) + O(\frac{1}{k^2})$, $k \in N$

$\Rightarrow \underline{P}(\underline{x} + \frac{k'}{k}\underline{a}') = \underline{P}(\underline{x}) + k'.O(\frac{1}{k^2})$, $k,k' \in N$, by induction

$\Rightarrow \underline{P}(\underline{x} + \underline{a}') = \underline{P}(\underline{x}) + O(\frac{1}{k})$

$\Rightarrow \underline{P}(\underline{x} + \underline{a}') = \underline{P}(\underline{x})$ when $k \to \infty$.

i.e. $\underline{a}' = \underline{0}$ since $\underline{P}$ has no translational symmetries.

Thus, as before $\underline{P}(\underline{x} + \underline{a}') + A''^{-1}\underline{a}'' = \underline{P}(\underline{x})$, and so $\underline{a}'' = \underline{0}$. ∎

The above lemma is of substantial use in later proofs, and indeed could be used to establish a slightly restricted version of the following theorem, which could not be stated, however, until after several definitions.

Theorem 2.1 If $\underline{P}$ is homogeneous and of degree $d \geq 2$, and has no translational symmetries, then all linear affine symmetries of $\underline{P}$ are of the form $\bar{T}' = (T', \underline{0})$ satisfying $\exists T'' \; \forall \underline{x} \; T''\underline{P}(T'\underline{x}) = \underline{P}(\underline{x})$.

i.e. there are no translational parts to $\bar{T}'$ and $\bar{T}''$.

Proof. If $\bar{T}'$ is a symmetry of $\underline{P}$ then $\bar{T}'$ is non-singular, else otherwise $\exists \underline{a} \neq \underline{0}$ with $\forall \underline{x} \; \bar{T}'(\underline{x} + \underline{a}) = \bar{T}'\underline{x}$, and therefore

$$\exists \bar{T}'' \forall \underline{x} \; \underline{P}(\underline{x}) = \bar{T}''\underline{P}(\bar{T}'\underline{x}) = \bar{T}''\underline{P}(\bar{T}'(\underline{x} + \underline{a})) = \underline{P}(\underline{x} + \underline{a}),$$

i.e. $\underline{P}$ has a translational symmetry contrary to assumption.

Let $\bar{T}'$ be a symmetry of $\underline{P}$, then $\exists \bar{T}'' \; \bar{T}''\underline{P}(\bar{T}\,\underline{x}) = \underline{P}(\underline{x})$ for all $\underline{x}$.

Thus $\exists \bar{T}'' \; \forall x \; \bar{T}''\underline{P}(\underline{x}) = \underline{P}(\bar{T}'^{-1}\underline{x})$.

The left hand side of this equation is homogeneous except for the translational part of $\bar{T}''$ (i.e. $\underline{t}''$), and is of degree d.

$\underline{P}(T'^{-1}\underline{x} - T^{-1}\,\underline{t}') - \underline{t}''$ should be homogeneous and of degree d (using the inverse of an affine linear transformation given at the start of the chapter). Thus $\underline{P}(\underline{x} - \underline{t}') + \underline{t}''$ should be homogeneous and of

degree d, and since $(d-1) \geq 1$ a Taylor expansion gives $(\underline{t}' \cdot \nabla)\underline{P}(\underline{x}) = \underline{0}$, which implies $\underline{P}(\underline{x} + \underline{t}') = \underline{P}(\underline{x})$ (as in the proof of lemma 2.1). Thus $\underline{t}' = \underline{0}$ as $\underline{P}$ has no translational symmetries, and immediately $\underline{t}'' = \underline{0}$. ∎

This theorem is proved here in advance, so that the reasons for defining symmetries of homogeneous polynomials without translational parts are apparent. If the inhomogeneous case was to be considered further, the natural place for this theorem would be after the inhomogeneous equivalent of theorem 2.4. However:-

Hereafter, we will consider only homogeneous families of polynomials. The inhomogeneous case is pursued by the author [31] and differs only in that the expressions are correspondingly more complex. In particular, entirely analogous methods and proofs are given.

Defn. 2.4 T' is a linear symmetry of $\underline{P}$ iff

$$\exists T'' \ \forall \underline{x} \ T''\underline{P}(T'\underline{x}) = P(\underline{x})$$

Defn. 2.5 $\underline{P}$ is linearly independent iff $\underline{a} \cdot \underline{P}(\underline{x}) = o \Rightarrow \underline{a} = \underline{0}$.
This is just the usual definition.
If $\underline{P}$ is not linearly independent, then given a linear symmetry T' of $\underline{P}$ there may exist two distinct matrices, T''_1 and T''_2 that satisfy $T''_1\underline{P}(T'\underline{x}) = T''_2\underline{P}(T'\underline{x}) = \underline{P}(\underline{x})$ because they give rise to the same result when acting on $\underline{P}$ i.e. $T''_1\underline{P}(\underline{x}) = T''_2\underline{P}(\underline{x})$.
We now have enough to explore the algebraic structure of the set of symmetries of a polynomial:-

Defn. 2.6 $G'_{\underline{P}} = (E', *')$ where $E' = \{T' \mid \exists T'' \ T''\underline{P}(T'\underline{x}) = \underline{P}(\underline{x})\}$ and $T'_1 *' T'_2 = T'_1 T'_2$ (matrix multiplication).

$G'_{\underline{P}}$ is the set of symmetries of $\underline{P}$ with matrix multiplication as composition.

Defn. 2.7 $G''_{\underline{P}} = (E'',*'')$ where $E'' = \{T'' \mid \exists T', T''\underline{P}(T'\underline{x}) = \underline{P}(\underline{x})\}$ and $T''_1 *'' T''_2 = T''_2 T''_1$ (reversed matrix multiplication).

Theorem 2.2 $G'_{\underline{P}}$ is a semi-group.

Proof

Closure: If $T'_1, T'_2 \in G'_{\underline{P}}$ then $\exists T''_1, T''_2 \forall \underline{x}\ T''_1\underline{P}(T'_1\underline{x}) = T''_2\underline{P}(T'_2\underline{x}) = \underline{P}(\underline{x})$

so $T''_2T''_1\underline{P}(T'_1(T'_2\underline{x})) = T''_2\underline{P}(T'_2\underline{x}) = \underline{P}(\underline{x})$ i.e. $T'_1T'_2 \in G'_{\underline{P}}$.

Identity: $I \in G'_{\underline{P}}$ since $I\underline{P}(I\underline{x}) = \underline{P}(\underline{x})$; I is the identity for matrix multiplication.

Associativity: Matrix multiplication is associative.

Theorem 2.3 If $\underline{P}$ has no translational symmetry and is linearly independent then $G'_{\underline{P}}$ is a group.

Proof

$G'_{\underline{P}}$ is a semi-group anyway, so all that is needed is for each symmetry in $G'_{\underline{P}}$ to have an inverse in $G'_{\underline{P}}$.

Let $T' \in G'_{\underline{P}}$ and suppose T' is non-singular as a matrix, then $\exists T''\ T''\underline{P}(T'\underline{x}) = \underline{P}(\underline{x})$ and so $T''\underline{P}(\underline{x}) = \underline{P}(T'^{-1}\underline{x})$. If T'' is non-singular then $T''^{-1}\underline{P}(T'^{-1}\underline{x}) = \underline{P}(\underline{x})$ i.e. $T'^{-1} \in G'_{\underline{P}}$.

However T' is non-singular, otherwise $\exists \underline{a} \neq \underline{0}$ with $T'\underline{a} = \underline{0}$, and $\underline{P}(\underline{x}) = T''\underline{P}(T'\underline{x}) = T''\underline{P}(T'(\underline{x} + \underline{a})) = \underline{P}(\underline{x} + \underline{a})$ i.e. $\underline{P}$ has a translational symmetry.

Also T'' is non-singular, otherwise $\exists \underline{a} \neq \underline{0}$ with $\underline{a}^T T'' = \underline{0}^T$, and $\underline{a}^T\underline{P}(\underline{x}) = \underline{a}^T T''\underline{P}(T'\underline{x}) = \underline{0}$ i.e. $\underline{P}$ is linearly dependent.

Theorem 2.4 If $\underline{P}$ is linearly independent with no translational symmetry then $G''_{\underline{P}}$ is a group, and there exists a group homomorphism $\varphi_{\underline{P}}: G'_{\underline{P}} \to G''_{\underline{P}}$ with the property that $\forall T' \in G'_{\underline{P}}$ $\varphi_{\underline{P}}(T')\underline{P}(T'\underline{x}) = \underline{P}(\underline{x})$. i.e. $\varphi_{\underline{P}}(T')$ is the unique T" in defn. 2.6 corresponding to a given $T' \in G'_{\underline{P}}$.

Proof $G''_{\underline{P}}$ is obviously a group under these conditions.

Let $T' \in G'_{\underline{P}}$, then $\exists T'' \in G''_{\underline{P}}$ $T''\underline{P}(T'\underline{x}) = \underline{P}(\underline{x})$.

Suppose $\exists T''_1 \neq T''_2 \in G''_{\underline{P}}$ with $T''_1\underline{P}(T'\underline{x}) = T''_2\underline{P}(T'\underline{x}) = \underline{P}(\underline{x})$.

Then, subtracting, $(T''_1 - T''_2)\underline{P}(T'\underline{x}) = \underline{0}$, for all $\underline{x}$.

Thus $(T''_1 - T''_2)\underline{P}(\underline{x}) = \underline{0}$, which implies $T''_1 = T''_2$ (a contradiction) since $\underline{P}$ is linearly independent. Therefore, given $T' \in G'_{\underline{P}}$, the corresponding $T'' \in G''_{\underline{P}}$ (with $T''\underline{P}(T'\underline{x}) = \underline{P}(\underline{x})$) is unique; i.e. there is a function $\varphi: G'_{\underline{P}} \to G''_{\underline{P}}$ with $\forall T' \in G'_{\underline{P}}$ $\varphi(T')\underline{P}(T'\underline{x}) = \underline{P}(\underline{x})$.

Clearly $\varphi(I) = I$, and $\varphi(T'_1T'_2) = \varphi(T'_2)\cdot\varphi(T'_1)$,

(since $\varphi(T'_1)\underline{P}(T'_1\underline{x}) = \underline{P}(\underline{x})$ and $\varphi(T'_2)\underline{P}(T'_2\underline{x}) = \underline{P}(\underline{x})$

so $\varphi(T'_2)\cdot\varphi(T'_1)\cdot\underline{P}(T'_1T'_2\underline{x}) = \varphi(T'_2)\cdot\underline{P}(T_2\underline{x}) = \underline{P}(\underline{x})$

but $\varphi(T'_1T'_2)\cdot\underline{P}(T'_1T'_2\underline{x}) = \underline{P}(\underline{x})$, and the result follows by uniqueness)

Therefore φ is a homomorphism. ∎

The conditions for $G'_{\underline{P}}$ and $G''_{\underline{P}}$ to be groups and for the homomorphism $\varphi_{\underline{P}}$ to exist are not restrictive. Indeed, for the reasons that follow, it is very unlikely that any naturally (computationally) arising family of polynomials (of degree ≥ 2) will fail to satisfy those conditions. Linear independence is very likely, since otherwise it is pointless evaluating anything more than a maximal linearly independent sub-family, the remainder being computationally redundant. The absence of any translational symmetry is also very likely, since otherwise the family of polynomials may be expressed in

terms of a "basis" of those linear combinations of the indeterminates that they depend upon and those that are translational symmetries. Treating these as new indeterminates, the family does not depend upon those that correspond to translational symmetries, and thus we have a new family depending upon fewer variables, revealing the computational redundancy of the original family. For these reasons we will refer to the conditions for $G'_{\underline{P}}$ and $G''_{\underline{P}}$ to be groups as (computational) irredundance of $\underline{P}$. The above arguments have indicated that if $\underline{P}(\underline{y})$ is redundant then we can find two (possibly non-square) matrices such that $L''\underline{P}(L'\underline{x})$ is an irredundant family, with fewer polynomials or indeterminates.

(ii) Linear Algebraic Groups

This section outlines existing mathematics that will be of extensive use here. There are many works on Algebraic geometry e.g. [15,22] (the study of algebraic sets) and linear algebraic groups e.g. [4]. First, two definitions to show the relationship between the two areas.

Defn. 2.8 An algebraic set is any set of points that consists of all simultaneous roots of some (finite or infinite) set of multivariate polynomials.

Defn. 2.9 A linear algebraic group is a subgroup of GL(n) for some n, that is an algebraic set in the space M(n) of all n×n matrices. The following definitions are to enable something to be said about the structure and nature of algebraic sets.

Defn. 2.10 An algebraic set is said to be reducible if it is the (non-trivial) union of two algebraic sets. If not reducible then irreducible. An irreducible algebraic set is called a variety. (Some authors call an algebraic set a variety, and thus have to resort to the term "irreducible variety").

Theorem 2.5 A Variety is connected.
For proof see [15].

Theorem 2.6 Any algebraic set may be expressed as all simultaneous roots of some finite set of polynomials. (The Basis theorem.).
For proof see [15]. Such a set of polynomials is called a (finite) basis for the algebraic set.

Theorem 2.7 Any algebraic set is uniquely a finite union of varieties,

no one of which is wholly contained in any other, though they may intersect. These are called the components of the algebraic set. For proof see [15].

Defn. 2.11 The Zariski Tangent Space to a variety V (given by the basis $\underline{f}$) at the point $\underline{P} \in V$ is defined to be

$$Z_{\underline{P}}(V) = \{\underline{a} \mid \underline{f}(\underline{P} + \varepsilon\underline{a}) = \underline{f}(\underline{P}) + O(\varepsilon^2)\},$$

and is clearly a vector space.

Defn. 2.12 A variety V is said to be smooth at a point $\underline{P} \in V$ if there is an analytic bijection from a neighbourhood of the origin within the $\underline{P}$-tangent space to a neighbourhood of $\underline{P}$ within V. A variety V is said to be smooth if $\forall \underline{P} \in V$, V is smooth at $\underline{P}$.

Defn. 2.13 The dimension of a variety is defined to be

$$\min\{\dim Z_{\underline{P}}(V) \mid \underline{P} \in V\}$$

Theorem 2.8 A variety is smooth iff $\forall \underline{P}, \underline{q} \in V$ $\dim Z_{\underline{P}}(V) = \dim Z_{\underline{q}}(V)$ i.e. all tangent spaces have the same dimension.
For proof see [15].

Theorem 2.9 A variety is smooth at at least one point. (In fact almost everywhere.)
For proof see [15].
The immediate consequences of algebraic geometry for linear algebraic groups are as follows.

Theorem 2.10 A linear algebraic group has the following property:-

(i) There are a finite number of smooth, non-intersecting, connected components.

(ii) The component containing the identity is a normal subgroup.

(iii) The analytic bijection of definition 2.12, where the tangent space to the identity element of the group is considered, is the exponential function of a matrix (over $\mathbb{C}$, defined by power series); the group is a Lie group [5,26].

For proofs see [4].

(iii) General Properties of Linear Symmetries

Some of the results in this section will depend upon the "existing mathematics", described in the previous section. Where this is so it will be stated, but even so, an outline of a simple direct proof may be given for completeness.

Firstly, a definition to show the relevance of the previous section.

Defn. 2.14 Let $G_{\underline{P}} = \left\{ \left(\begin{array}{c|c} T' & O \\ \hline O & \varphi_{\underline{P}}(T')^{\mathsf{T}} \end{array} \right) \middle| T' \in G'_{\underline{P}} \right\}$

Theorem 2.11 $G_{\underline{P}}$ is a linear algebraic group in GL(m + n) if $\underline{P}$ is irredundant.

Proof $G_{\underline{P}}$ is $\left\{ \left(\begin{array}{c|c} T' & O \\ \hline O & T''^{\mathsf{T}} \end{array} \right) \middle| \forall \underline{x}\ T''\underline{P}(T'\underline{x}) - \underline{P}(\underline{x}) = \underline{0} \right\}$

Thus $G_{\underline{P}}$ is an algebraic set since for each $\underline{x}$ the matrix entries must be a root of a polynomial. Indeed the matrix pair must be a simultaneous root of all the polynomials that can be obtained by substituting any constant $\underline{x}$. If $\underline{P}$ is irredundant then $G_{\underline{P}}$ is clearly a group under matrix multiplication (defns. 2.6, 2.7, theorems 2.2, 2.3, 2.4). (The transpose in the definition 2.14 is to convert the reverse matrix multiplication in defn. 2.7 into matrix multiplication).

The form of construction used in definition 2.14 will be of great further use. Clearly $G'_{\underline{P}}$ and $G''_{\underline{P}}$ are both homomorphic images of $G_{\underline{P}}$. It is convenient to regard ' and " as these homomorphisms, and this use will be extended in the obvious way to similar structures.

Defn. 2.15 Whenever there is a matrix structure E of the form $E = \left\{ \left(\begin{array}{c|c} D' & O \\ \hline O & D'' \end{array} \right) \middle| D'' = \psi(D'),\ D' \in E' \right\}$ where E' is homomorphic to E

and ψ is a homomorphism, we will regard ' as the homomorphism from E to E' and " as the homomorphism obtained by composing ' with ψ. Thus ψ is a homomorphism from E' to E".

By theorem 2.10 $G_{\underline{P}}$ consists of a finite number of smooth, non-intersecting connected components. This suggests the following definition.

Defn. 2.16 $\tilde{G}_{\underline{P}}$ is the component of $G_{\underline{P}}$ that contains the identity. For the purpose of outlined direct proofs we will take this to be the pathwise connected component defined as follows:

$$\tilde{G}_{\underline{P}} = \{g \mid g \in G_{\underline{P}} \text{ and there exists a continuous parameterisation } g(\lambda)$$
$$\text{where } \lambda \in \mathbb{R},\ 0 \leq \lambda \leq 1,\ g(0) = I \text{ (identity)},$$
$$g(1) = g,\ \forall \lambda\ g(\lambda) \in G_{\underline{P}}\}$$

Theorem 2.10 (iii) suggests we define the tangent space to the group at the identity, and look for an exponential map.

Defn. 2.17 The tangent space to $G_{\underline{P}}$ at $\mathbf{I}$ is

$$V_{\underline{P}} = \left\{ \left(\begin{array}{c|c} M' & O \\ \hline O & M''^{T} \end{array} \right) \;\middle|\; \forall \underline{x} (I + \varepsilon M'') \underline{P}((I + \varepsilon M')\underline{x}) = \underline{P}(\underline{x}) + O(\varepsilon^2) \right\}$$

and if $M \in V_{\underline{P}}$ then $(I + \varepsilon M)$ is called an "infinitesimal transformation".

Theorem 2.12 $V_{\underline{P}}$ is a finite dimensional vector space (with matrix addition).

Proof $O \in V_{\underline{P}}$ trivially. The application of two successive infinitesimal transformations shows that $M_1 \in V_{\underline{P}}$ and $M_2 \in V_{\underline{P}}$ implies $M_1 + M_2 \in V_{\underline{P}}$. The substitution of $\lambda\varepsilon$ for ε (for any constant λ) shows that $M \in V_{\underline{P}} \Rightarrow \lambda M \in V_{\underline{P}}$.

Remark: $V_{\underline{P}}$ is a Lie algebra [5] with product [A,B] = AB - BA.

Defn. 2.18 If A is a matrix let $e^A = \sum_{j=0}^{\infty} \frac{1}{j!} A^j$.

This is just the usual power series and converges everywhere for matrices over $\mathbb{C}$ [20].

Theorem 2.13 If $M \in V_{\underline{P}}$ then $e^M \in G_{\underline{P}}$.

Proof If $M = \left(\begin{array}{c|c} M' & O \\ \hline O & M''^T \end{array}\right)$ then $e^M = \left(\begin{array}{c|c} e^{M'} & O \\ \hline O & e^{M''^T} \end{array}\right)$

$M \in V_{\underline{P}} \Rightarrow (I + \varepsilon M')\underline{P}((I + \varepsilon M')\underline{x}) = \underline{P}(\underline{x}) + O(\varepsilon^2)$

$\Rightarrow (I + \frac{1}{j}M')\underline{P}((I + \frac{1}{j}M')\underline{x}) = \underline{P}(\underline{x}) + O(\frac{1}{j^2})$ $j \in \mathbb{N}$

$\Rightarrow (I + \frac{1}{j}M')^2\underline{P}((I + \frac{1}{j}M')^2\underline{x}) = (I + \frac{1}{j}M')\underline{P}((I + \frac{1}{j}M')\underline{x}) + O(\frac{1}{j^2})$

$= \underline{P}(\underline{x}) + 2 \cdot O(\frac{1}{j^2})$

$\Rightarrow (I + \frac{1}{j}M')^j\underline{P}((I + \frac{1}{j}M')^j\underline{x}) = \underline{P}(\underline{x}) + j \cdot O(\frac{1}{j^2})$

Taking lim $j \to \infty$ poses no problems since $\underline{P}$ is continuous and $\lim_{j\to\infty} (I + \frac{1}{j}M')^j$ expands into the power series for $e^{M'}$ since there are no commutation difficulties. Thus

$$e^{M''}\underline{P}(e^{M'}\underline{x}) = \underline{P}(\underline{x}) \text{ and } e^M = \left(\begin{array}{c|c} e^{M'} & O \\ \hline O & e^{M''^T} \end{array}\right) \in G_{\underline{P}}$$

Theorem 2.13 shows that the exponential function maps $V_{\underline{P}}$ into $G_{\underline{P}}$ and theorem 2.10 assures us that this is an analytic bijection within some neighbourhoods of O and I in $V_{\underline{P}}$ and $G_{\underline{P}}$ respectively, but can any more be said? The next theorem assures us that any member of $\tilde{G}_{\underline{P}}$ can be expressed as a finite product of exponentials of tangent space matrices.

Defn. 2.19 Let $\exp(V_{\underline{P}}) = \left\{\prod_{k=1}^{j} e^{M_k} \mid M_k \in V_{\underline{P}},\ j \in \mathbb{N}\right\}$

(where the product is matrix multiplication).

Theorem 2.14 $\exp(V_{\underline{P}}) = \tilde{G}_{\underline{P}}$ if $\underline{P}$ is irredundant.

Proof

(i) $\exp(V_{\underline{P}}) \subseteq \tilde{G}_{\underline{P}}$

Let $g = \prod_{k=1}^{j} e^{M_k}$ be any member of $\exp(V_{\underline{P}})$ where $M_k \in V_{\underline{P}}$.

Then, since $V_{\underline{P}}$ is a vector space $g(\lambda) = \prod_{k=1}^{j} e^{\lambda M_k} \in G_{\underline{P}}$ by theorem 2.13 and group closure.

However, this is just the parameterisation of definition 2.16 that shows $g \in \tilde{G}_{\underline{P}}$.

(ii) $\tilde{G}_{\underline{P}} \subseteq \exp(V_{\underline{P}})$.

A matrix norm is defined in [20 p.152]; denote this by $||X||$.

In [20 p.158] it is proved that the exponential function of a matrix $A \to \exp(A)$ is an analytic bijection from the neighbourhood $\{A| \ ||A|| < \varepsilon\}$ of the zero matrix to the neighbourhood $\{B| \ ||B-I|| < \delta\}$ of the identity matrix, for some pair of positive constants ε,δ. In particular, this must also be true when the neighbourhoods are restricted to lie within $V_{\underline{P}}$ and $G_{\underline{P}}$ respectively, in accordance with theorem 2.10 (iii).

Thus, if $T \in \tilde{G}_{\underline{P}}$ and $||T-I|| < \delta$ then there is a unique $M \in V_{\underline{P}}$ with $||M|| < \varepsilon$ such that $e^M = T$.

Let $T \in \tilde{G}_{\underline{P}}$, and $T(\lambda)$ be the parameterisation from definition 2.16, with $T(0) = I$, $T(1) = T$. If T can be expressed as a finite product of matrices $B_i \in \tilde{G}_{\underline{P}}$, each $||B_i-I|| < \delta$, then T can be written as a finite product of exponentials of tangent space matrices, i.e. $T \in \exp(V_{\underline{P}})$ (definition 2.19).

Choose the matrices B_i as follows:-

For all $0 \leq \lambda \leq 1$ define the open sets (in $\mathbb{R}$)

$$S_\lambda = \{\mu| \ ||T(\lambda)^{-1}T(\mu)-I|| < \delta\}$$

$$S'_\lambda = \{\mu \mid \|T(\mu)^{-1}T(\lambda)-I\| < \delta\}$$

and let Γ_λ be the connected part of $S_\lambda \cap S'_\lambda$ that includes λ.

Clearly if $\lambda \in \Gamma_\lambda$, then $\lambda' \in \Gamma_\lambda$.

The open sets Γ_λ cover the interval [0,1], and so by a well-known theorem of analysis a finite sub-covering may be chosen: let this be Γ_{λ_i} with $1 \leq i \leq N$. Thus $\bigcup_{i=1}^{N} \Gamma_{\lambda_i} = [0,1]$.

In particular we may choose this finite covering to have the following properties:- $\lambda_1 = 0$, $\lambda_N = 1$ (simply by adding into the finite covering the sets Γ_0 and Γ_1), $\lambda_i < \lambda_{i+1}$ for $1 \leq i < N$, (by choosing the numbering correctly) and $\Gamma_{\lambda_i} \not\subseteq \Gamma_{\lambda_j}$ for $i \neq j$ (by omitting any Γ_{λ_i} for which this is false from the finite covering). Then $\Gamma_{\lambda_i} \cap \Gamma_{\lambda_{i+1}} \neq \emptyset$ for $1 \leq i < N$, so choose one point μ_i from $\Gamma_{\lambda_i} \cap \Gamma_{\lambda_{i+1}}$ for each i. Define η_i for $1 \leq i \leq 2N-1$ as follows: if i is even $\eta_i = \mu_{i/2}$; if i is odd $\eta_i = \lambda_{(i+1)/2}$; i.e. the η_i are the λ_i and the μ_i alternated.

Thus $\eta_i \in \Gamma_{\eta_{i+1}}$ for $1 \leq i < 2N-1$.

Therefore $\|T(\eta_i)^{-1}T(\eta_{i+1})-I\| < \delta$.

Choose $B_i = T(\eta_i)^{-1}T(\eta_{i+1})$, then $\prod_{i=1}^{2N-1} B_i = T(0)^{-1}T(1) = T$, and B_i satisfies the condition $\|B_i-I\| < \delta$.

The reason why $G_{\underline{P}}$ is smooth is that given any two members of $G_{\underline{P}}$, T_1 and T_2, there is an analytic bijection between neighbourhoods of them (given by $T_2T_1^{-1}$), whose existence is a consequence of being a group. Thus any neighbourhood in $G_{\underline{P}}$ is analytically homeomorphic to an open ball in $V_{\underline{P}}$, showing $G_{\underline{P}}$ to be an analytic manifold.

The reason why $\tilde{G}_{\underline{P}}$ is a normal subgroup of $G_{\underline{P}}$ (theorem 2.10) is as follows:- clearly $\tilde{G}_{\underline{P}}$ is a subgroup of $G_{\underline{P}}$ since $\tilde{G}_{\underline{P}} = \exp(V_{\underline{P}})$. For any $T \in G_{\underline{P}}$, $T\tilde{G}_{\underline{P}}T^{-1}$ must be a connected component of $G_{\underline{P}}$, but it

contains the identity, and so must be $\tilde{G}_{\underline{P}}$.

Since $G_{\underline{P}}$ is smooth, all tangent spaces have the same dimension, and so, noting definition 2.13, we say the following:-

Defn. 2.20 The dimension of $G_{\underline{P}}$, $\dim(G_{\underline{P}}) = \dim(V_{\underline{P}})$.

This can be interpreted as the number of "degrees of freedom" in $G_{\underline{P}}$ i.e. the number of parameters that can (smoothly) specify a particular member of the group. All degrees of freedom are in $G'_{\underline{P}}$ because of the homomorphism from $G'_{\underline{P}}$ to $G''_{\underline{P}}$.

Theorem 2.15 $V_{\underline{P}}$ can be found by solving the linear equations obtained by equating the coefficient of each monomial to zero in the following expression:-

$$(M'\underline{x})\cdot\nabla\, \underline{P}(\underline{x}) + M''\underline{P}(\underline{x}) = \underline{0}$$

to obtain all $M = \left(\begin{array}{c|c} M' & O \\ \hline O & M''^{T} \end{array}\right)$.

Equally, if M is given and $\underline{P}$ is to possess $M \in V_{\underline{P}}$ then $\underline{P}$ must satisfy the above as a partial differential equation.

Proof

$M \in V_{\underline{P}} \iff (I + \varepsilon M'')\underline{P}((I + \varepsilon M')\underline{x}) = \underline{P}(\underline{x}) + O(\varepsilon^2)$ (definition 2.17)

but $\underline{P}((I + \varepsilon M')\underline{x}) = \underline{P}(\underline{x}) + \varepsilon(M'\underline{x})\cdot\nabla\, \underline{P}(\underline{x}) + O(\varepsilon^2)$

Notice that Euler's equation,

$$(\underline{x}\cdot\nabla)\, f(\underline{x}) + k\cdot f(\underline{x}) = 0,$$

that is satisfied by homogeneous functions of degree k, is a special case of the equation in theorem 2.15 with a single function (so that M" is 1×1) and M' = I.

Hereafter symmetries of $\underline{P}$ that are in the connected normal subgroup $\tilde{G}_{\underline{P}}$, will be referred to as "continuous" symmetries, as they

are essentially generated by the infinitesimal transformations (definition 2.17). In contrast, representative members of other cosets of $\tilde{G}_P$ in G_P will be referred to as "discrete" symmetries. (Alternatively, one could regard $G_P/\tilde{G}_P$ as the group of "discrete" symmetries.) These other cosets are clearly just the other connected components of G_P, and theorem 2.10 asserts that there are only a finite number of them. Thus there are only a finite number of "discrete" symmetries.

The following definitions are motivated by the remarks in the introduction concerning the relationship between the wrapped convolution and the pairwise product, and its generalisation to a form of invertable projection that we call affine similarity.

Defn. 2.21 $P(x)$ and $Q(x)$ are affinely similar polynomials if there exist non-singular linear affine transformations $\bar{S}', \bar{S}''$ with $P(x) = \bar{S}''Q(\bar{S}'x)$. This is written $P \equiv Q$.

It seems likely that homogeneity places some constraint upon the form of such a relation, and so we make the following more restricted definition involving no translations.

Defn. 2.22 $P(x)$ and $Q(x)$ are said to be similar polynomials if there exist non-singular square matrices S', S'' ($n\times n$ and $m\times m$ respectively) with $P(x) = S''Q(S'x)$. This is written $P \sim Q$ via the transformation $S = \left(\begin{array}{c|c} S' & O \\ \hline O & S''^{T} \end{array}\right)$.

Theorem 2.16 If P and Q are irredundant homogeneous polynomials of different degrees (≥ 2), then $P \not\equiv Q$; if they are of the same degree then $P \equiv Q$ implies $P \sim Q$.

Proof Suppose $P \equiv Q$, then $\exists \bar{S}', \bar{S}''$, both non-singular, with

$\bar{S}''\underline{Q}(\bar{S}'(\underline{x})) = \underline{P}(\underline{x})$. Lemma 2.1 says $\bar{S}''\underline{Q}(\bar{S}'\underline{x})$ is homogeneous iff $\underline{s}'' = \underline{0}$ and $\underline{s}' = \underline{0}$, when it is of the same degree as $\underline{Q}$.

Theorem 2.17 Similarity is an equivalence relation.

Proof

Reflexive: $\underline{P} \sim \underline{P}$ via the identity transformation

Symmetric: $\underline{P} \sim \underline{Q}$ via S implies $\underline{Q} \sim \underline{P}$ via S^{-1}

(since $\forall \underline{x}\ \underline{P}(\underline{x}) = S''\underline{Q}(S'\underline{x}) \Rightarrow \forall \underline{x}'\ S''^{-1}\underline{P}(S'^{-1}\underline{x}') = \underline{Q}(\underline{x}')$ by substituting $S'^{-1}\underline{x}'$ for $\underline{x}$).

Transitive: $\underline{P} \sim \underline{Q}$ via S_1 and $\underline{Q} \sim \underline{R}$ via S_2 implies $\underline{P} \sim \underline{R}$ via S_2S_1 (by means of a similar substitution).

As discussed in the introduction, similarity imposes great structural constraints upon any pair of polynomials so related. In particular, it imposes the following restrictions upon symmetries:-

Theorem 2.18 If $\underline{P} \sim \underline{Q}$ via S, then $G_{\underline{P}} \tilde{=} G_{\underline{Q}}$ (group isomorphism and algebraic set isomorphism), with

$$T \in G_{\underline{P}} \Leftrightarrow STS^{-1} \in G_{\underline{Q}},$$

and $V_{\underline{P}} \tilde{=} V_{\underline{Q}}$ (vector space isomorphism),

and in particular $\dim(G_{\underline{P}}) = \dim(G_{\underline{Q}})$.

Proof

Let $T \in G_{\underline{P}}$, then $T''\underline{P}(T'\underline{x}) = \underline{P}(\underline{x}) = S''\underline{Q}(S'\underline{x})$ if $\underline{P} \sim \underline{Q}$ via S.

Thus $T''S''\underline{Q}(S'T'\underline{x}) = \underline{P}(\underline{x}) = S''\underline{Q}(S'\underline{x})$

$\Rightarrow\ S''^{-1}T''S''\underline{Q}(S'T'S'^{-1}\underline{x}) = \underline{Q}(\underline{x})$

$\Rightarrow\ STS^{-1} \in G_{\underline{Q}}$.

Equally, $\underline{Q} \sim \underline{P}$ via S^{-1}, and therefore $U \in G_{\underline{Q}} \Rightarrow S^{-1}US \in G_{\underline{P}}$ by the

same argument.

Thus $T \in G_{\underline{P}} \iff STS^{-1} \in G_{\underline{Q}}$, which is clearly a group isomorphism since it is a bijection that preserves identity, product and inverses, e.g. $T_1, T_2 \in G_{\underline{P}}$ correspond to ST_1S^{-1}, $ST_2S^{-1} \in G_{\underline{Q}}$ and $T_1T_2 \in G_{\underline{P}}$ corresponds to $ST_1T_2S^{-1} = (ST_1S^{-1})\cdot(ST_2S^{-1}) \in G_{\underline{Q}}$.
Thus $G_{\underline{P}} \tilde{=} G_{\underline{Q}}$.
$V_{\underline{P}} \tilde{=} V_{\underline{Q}}$ follows from consideration of infinitesimal transformations (definition 2.17) and application of the above argument to within $O(\varepsilon^2)$.
This gives $M \in V_{\underline{P}} \iff SMS^{-1} \in V_{\underline{Q}}$ which is clearly a linear bijection between $V_{\underline{P}}$ and $V_{\underline{Q}}$. (In fact is a Lie algebra isomorphism.)

Baur and Strassen [3] give a method of transforming a circuit to compute a polynomial into a circuit to compute all of its partial derivatives that is at most three times as large, showing that the complexity of computing partial derivatives is essentially the same as the complexity of computing the function itself. It can be shown that the symmetries of a polynomial are preserved by various integrations and differentiations, but we give only one such result here as it gives rise to recognisable concrete results.

Theorem 2.19 If $T' \in G'_{\underline{P}}$ then $T' \in G'_{\nabla\underline{P}}$ where ∇P consists of all partial derivatives $\frac{\partial P_i}{\partial x_j}$. (Thus $G'_{\underline{P}} \subseteq G'_{\nabla\underline{P}}$.)

Proof $T' \in G'_{\underline{P}} \Rightarrow \exists T''\ T''\underline{P}(T'\underline{x}) = \underline{P}(\underline{x})$

i.e. $\sum_{j=1}^{m} T''_{ij} P_j(T'\underline{x}) = P_i(\underline{x})$

$$\Rightarrow \sum_{j=1}^{m} T''_{ij} \frac{\partial}{\partial x_k} P_j(T'\underline{x}) = \frac{\partial P_i}{\partial x_k}(\underline{x})$$

but let $\underline{y} = T'\underline{x}$, then $\frac{\partial}{\partial x_k} = \sum_{l=1}^{n} \frac{\partial y_l}{\partial x_k} \frac{\partial}{\partial y_l}$

and $\frac{\partial y_l}{\partial x_k} = \frac{\partial}{\partial x_k} \sum_{p=1}^{n} T'_{lp} x_p = T'_{lk}$.

Thus $\frac{\partial}{\partial x_k} P_j(T'\underline{x}) = \sum_{l=1}^{n} T'_{lk} \frac{\partial}{\partial y_l} P_j(\underline{y})$, and substituting this gives:-

$$\sum_{j=1}^{m} \sum_{l=1}^{n} T''_{ij} T'_{lk} \frac{\partial P_j}{\partial y_l}(\underline{y}) \Bigg|_{\underline{y}=T'\underline{x}} = \frac{\partial P_i}{\partial x_k}(\underline{x})$$

The above theorem says that "differentiation increases symmetry", but unfortunately this does not imply that an elimination-style algorithm for a polynomial can be used to evaluate its derivatives. Given a polynomial $\underline{P}$ and its symmetry group $\tilde{G}_{\underline{P}}$, a Gaussian elimination style algorithm to evaluate $\underline{P}$ works as follows:-

1. Given the input $\underline{x}$ choose a $T' \in \tilde{G}'_{\underline{P}}$ such that certain standard entries in $T'\underline{x}$ are zero.

2. Evaluate $\underline{P}$ at $T'\underline{x}$ making use of a small formula (or circuit) for $\underline{P}$ with the standard inputs set to zero.

3. Evaluate the desired results from the results of 2. by taking linear combinations using $T'' = \varphi_{\underline{P}}(T') \in G''_{\underline{P}}$.

In order for such an algorithm to exist, firstly $G'_{\underline{P}}$ must be sufficient to introduce the "standard position zeros" no matter what the input, and secondly $\underline{P}$ with zero substituted for the indeterminates in these standard positions must have a small formula. In the case of the determinant this second criterion is satisfied by the substitution of the zeros into the usual formula "knocking out" all but one of the monomials (since the determinant of a triangular matrix is the product of its diagonal elements). However, the derivative of a monomial may not become zero under a substitution of zeros that sets the monomial to zero, and hence the derivative of

a polynomial may not have an obvious small formula when standard indeterminates are set to zero, even though the polynomial itself does.

With regard to the first criterion, it is clear that diagonal matrices in $G'_{\underline{p}}$ cannot introduce any zeros, since they must be non-singular, as $G'_{\underline{p}}$ is a group. It is also apparent that the coefficients in a set of linear combinations of the inputs that are equal to zero may be written as rational functions of the inputs involving several arbitrary parameters, since they merely consist of the components of any set of vectors orthogonal to the input vector. (If the symmetry $T' \in G'_{\underline{p}}$ is chosen as a rational function of the input $\underline{x}$ in stage 1. of the algorithm, the example of the determinant shows that divisions may be needed, and yet then the circuit so constructed would not compute the polynomial on some algebraic subset of the inputs because of division by zero. Fortunately Strassen [29] shows that a polynomial of degree d and circuit size C using divisions has a circuit of size O(Cd) using no divisions. In this way Gaussian elimination style algorithms can always be regarded as giving rise to small circuits.)

3. APPLICATIONS TO SPECIFIC POLYNOMIALS

In this chapter we will use the general theory developed in chapter 2 to find the continuous symmetries of a number of well-known and computationally interesting polynomials, and the results are used as a stimulus to further investigation, the results of which occur towards the second half of the chapter.

One question of interest is whether or not there is a bilinear product that preserves the permanent (by analogy with the determinant) i.e. a matrix product Δ with the property $\mathrm{per}(A \Delta B) = \mathrm{per}(A)\cdot\mathrm{per}(B)$ where A,B are $n\times n$ square matrices and $(A \Delta B)_{ij} = \sum_{klmp} \psi_{ijklmp} A_{kl} B_{mp}$ with ψ_{ijklmp} independent of A and B.

(i) Notation

Throughout the previous chapter the indeterminates $\underline{x}$ were $\begin{pmatrix} x_1 \\ \vdots \\ x_n \end{pmatrix}$ and the polynomials $\underline{P}$ were $\begin{pmatrix} P_1 \\ \vdots \\ P_m \end{pmatrix}$. In applying the theory to polynomials such as the determinant and the matrix multiplication polynomials, it is necessary that the "inputs" and "outputs" should conform to this even though they may be laid out in a matrix or matrices. For this purpose we will adopt the following notation:- vec will stand for any linear bijection that packs a single or several matrices entries into a column matrix. For given numbers and sizes of its arguments there will be a single function vec supposedly chosen by some fixed convention from the possible different orders of packing the arguments into a column vector.

e.g. $\underline{\text{vec}}\ (X) \triangleq \begin{pmatrix} X_{11} \\ X_{12} \\ \vdots \\ X_{1n} \\ X_{21} \\ \vdots \\ X_{2n} \\ \vdots \\ X_{n1} \\ \vdots \\ X_{nn} \end{pmatrix}$ where X is $n \times n$.

It is not important which convention is chosen provided it is adhered to. Polynomials will be deemed defined both on their usual form of arguments and on arguments packed by vec.

e.g. if $\underline{x} = \underline{\text{vec}}(X)$ then $\det(\underline{x}) \triangleq \det(X)$.

However, polynomials with packed arguments will be deemed to have packed results (where this makes a difference).

e.g. if $\underline{M}$ is the matrix multiplication polynomials i.e. $\underline{M}(X,Y) = XY$ then if $\underline{x} = \underline{\text{vec}}(X,Y)$ then $\underline{M}(\underline{x}) \triangleq \underline{\text{vec}}(XY)$.

When subscripts are used, inputs and outputs will usually remain in their original subscription, and this may occasion changes in the subscripting of the symmetry matrices.

e.g. subscripts of a symmetry matrix of the determinant may be represented by pairs of subscripts. This is to be regarded as the convention that packing with vec does not change the subscription, but rather that subscript pairs (for example) may, on occasion be regarded as a single subscript.

(ii) Continuous Symmetries of the Permanent and Determinant

The procedure used to find all of the continuous symmetries of a polynomial is as follows. First the tangent space to the identity is found using theorem 2.15, then the set of exponentials of tangent space matrices are computed, and closed under matrix multiplication, which gives all continuous symmetries by theorem 2.14. The following theorems summarise the results for the permanent and determinant.

Theorem 3.1 The $n\times n$ permanent has $\dim(G_{per}) = 2n-1$ and

$\tilde{G}_{per} = \left\{\left(\begin{array}{c|c} T' & O \\ \hline O & T''^{T} \end{array}\right) \middle| T'\underline{x} = \underline{vec}(\Lambda X \Lambda') \right.$ where $\underline{x} = \underline{vec}(X)$ and Λ, Λ' are diagonal, non-singular.

$\left. T'' = (\det(\Lambda\Lambda'))^{-1} \right\}$

provided $n>2$.

Theorem 3.2 The $n\times n$ determinant has $\dim(G_{det}) = 2n^2-1$ and

$\tilde{G}_{det} = \left\{\left(\begin{array}{c|c} T' & O \\ \hline O & T''^{T} \end{array}\right) \middle| T'\underline{x} = \underline{vec}(AXB) \right.$ where $\underline{x} = \underline{vec}(X)$ and A,B are non-singular.

$\left. T'' = (\det(AB))^{-1} \right\}$

for $n \geq 1$.

Proof of Theorems 3.1 and 3.2

Firstly we note that the permanent and determinant have no translational symmetry. By Taylor expansion, this is equivalent to having linearly independent partial derivatives which the permanent and determinant clearly possess, since a given monomial only occurs in one (at most) partial derivative.

$\left(\begin{array}{c|c} M' & O \\ \hline O & M''^{T} \end{array}\right) \in V_{det}$ iff $(M'\underline{x})\cdot\nabla \det(\underline{x}) + M''\det(\underline{x}) = 0$ by theorem 2.15.

Thus, by equating the coefficients of the monomials in the above equation to zero, a series of linear equations may be found that must be satisfied only by all members of V_{det}.

Thus:

$$\sum_{ijkl} M'_{ijkl}\, x_{kl} \frac{\partial}{\partial x_{ij}} \det(\underline{x}) + M''\det(\underline{x}) = 0$$

-Where the subscript doubling occurs because the indeterminates constitute a matrix.

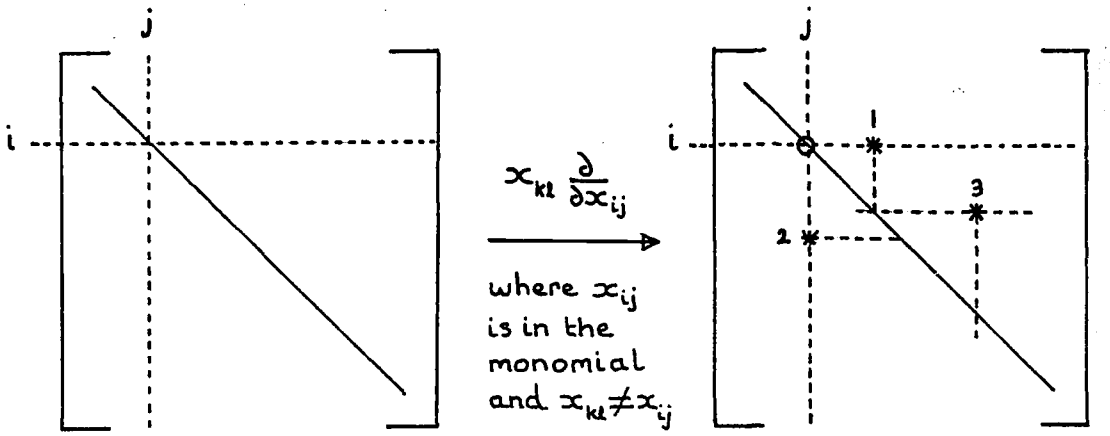
Applying the operator $x_{kl}\frac{\partial}{\partial x_{ij}}$ to a monomial in the determinant will now be investigated. Since each monomial is multilinear, x_{ij} either does not occur in a particular monomial (in which case the differentiation gives zero) or occurs to the first power in which case the effect of the operator $x_{kl}\frac{\partial}{\partial x_{ij}}$ is to "delete" x_{ij} in that monomial, and "replace" it with x_{kl}. There are several distinct cases:-

Case (i) $x_{kl} = x_{ij}$ (i.e. i=k and j=l)

Clearly if x_{ij} is in the monomial then $x_{ij}\frac{\partial}{\partial x_{ij}}$ does nothing.

Also if $x_{ij} \neq x_{kl}$ then $x_{kl}\frac{\partial}{\partial x_{ij}}$ acting on the monomial cannot result in a monomial that occurs in the determinant, as is illustrated by the following diagram:-

(The monomial in question is always a product of n indeterminates, each taken from a distinct row and column of the matrix. Thus, for the purpose of illustration, the columns of the matrix may be permuted in such a way as to place the indeterminates in the monomial along the diagonal.)



The monomial consists of the product of the diagonal entries, illustrated by the line.

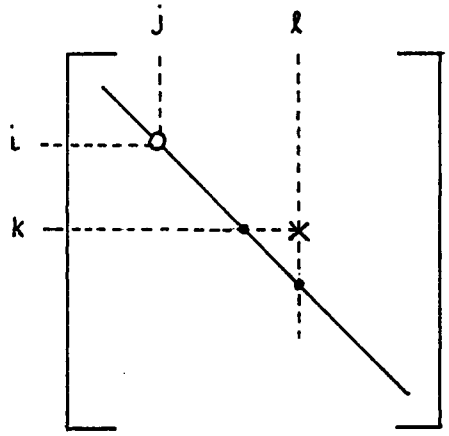
The monomial without the (i,j)th entry, as illustrated by the open circle. Possible locations of x_{kl} are e.g. $*_1, *_2$ or $*_3$.

Clearly if $x_{kl} \neq x_{ij}$ then the resulting monomial will have two indeterminates in the same row or column, and thus <u>cannot</u> be a monomial that occurs in the determinant.

<u>Case (ii)</u> x_{kl} is not in the same row or column as x_{ij}

(i.e. $i \neq k$ and $j \neq l$).

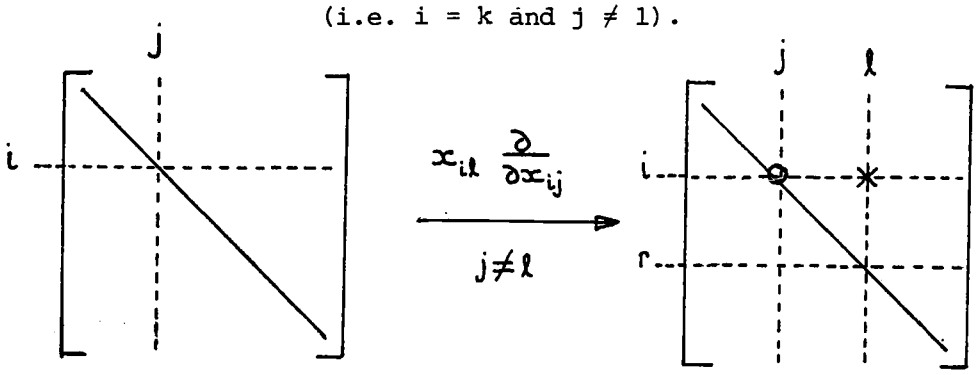
The resulting monomial after application of $x_{kl}\frac{\partial}{\partial x_{ij}}$ is as $*_3$ in the diagram above i.e.

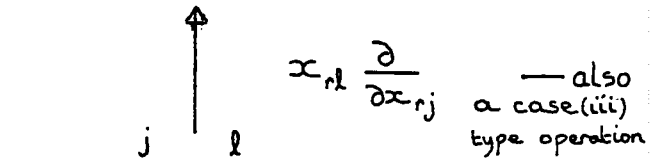


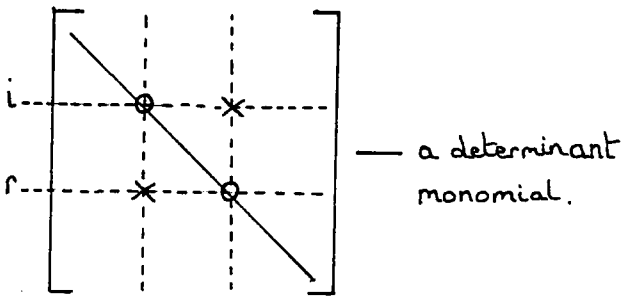
the monomial is the product of the diagonal indeterminates (excepting the circled one) and the crossed indeterminate.

This monomial could only be arrived at from the original monomial (and from no other monomial) in the determinant by means of a "deletion" and an "insertion", since reversing the process x_{kl} must be removed (as it is in the same row as one indeterminate and the same column as another) and thus x_{ij} must be inserted (so that there are no two indeterminates in the monomial in the same row or column).

Case (iii) x_{kl} is in the same row but not the same column as x_{ij} (i.e. $i = k$ and $j \neq l$).







Starting from the final monomial, and working back to a monomial in the determinant, we have two possibilities for a deletion that does not leave two indeterminates in the same row or column: x_{ij} or x_{rl}. The choice then fixes the relevant insertion by the same criterion. Thus the same monomial can be arrived at from two distinct monomials. Note that the latter is of opposite parity to the former as it differs by a single row transposition, and thus will have the opposite coefficient in the determinant.

Case (iv) x_{kl} is in the same column but not the same row as x_{ij} (i.e. $i \neq k$) and $j = 1$).

This case is just the transpose of case (iii).

These cases provide for the following separation of terms in the original equation:-

$$\underbrace{\left(\sum_{\substack{i\neq k\\ j\neq l}} M'_{ijkl}\, x_{kl}\frac{\partial}{\partial x_{ij}}\det(\underline{x})\right)}_{\text{contains all case (ii) monomials}} + \underbrace{\left(\sum_{ij} M'_{ijij}\, x_{ij}\frac{\partial}{\partial x_{ij}}\det(\underline{x}) + M''\det(\underline{x})\right)}_{\text{contains all case (i) monomials}}$$

$$+\underbrace{\left(\sum_{\substack{i\\ j\neq l}} M'_{ijil}\, x_{il}\frac{\partial}{\partial x_{ij}}\det(\underline{x})\right)}_{\text{contains all case (iii) monomials}} + \underbrace{\left(\sum_{\substack{j\\ i\neq k}} M'_{ijkj}\, x_{kj}\frac{\partial}{\partial x_{ij}}\det(\underline{x})\right)}_{\text{contains all case (iv) monomials}} = 0$$

Thus each of the bracketed expressions must vanish separately, since they have no monomials in common.

Case (i) expression. Let $X_{ij} = \frac{\partial}{\partial x_{ij}}\det(\underline{x})$.

Since $\det(\underline{x})$ is homogeneous and of degree n, it satisfies Euler's equation:

$$\sum_{ij} x_{ij}\frac{\partial}{\partial x_{ij}}\det(\underline{x}) = n\cdot\det(\underline{x})$$

i.e. $\det(\underline{x}) = \frac{1}{n}\sum_{ij} x_{ij}X_{ij}$.

Thus, substituting in the case (i) expression:

$$\sum_{ij}\left((M'_{ijij} + \frac{M''}{n})x_{ij}X_{ij}\right) = 0.$$

The coefficient of any monomial $\prod_{i=1}^{n} x_{i\sigma(i)}$ in this expression must vanish (where σ is any permutation of (1..n)). (These are the only monomials occurring here.)

Define $A_{ij} = M'_{ijij} + \frac{M''}{n}$, then $\sum_{ij} A_{ij}x_{ij}X_{ij} = 0$.

Since X_{rs} is the co-factor of x_{rs}, terms of the form $x_{i\sigma(i)}X_{i\sigma(i)}$ are going to contribute to the coefficient of $(-1)^{sgn(\sigma)}\prod_{j=1}^{n} x_{j\sigma(j)}$.

(Sgn(σ) is 1 for odd σ, 0 for even σ, and thus the expression is a single term in the determinant.)

Thus, the coefficient of $(-1)^{sgn(\sigma)} \prod_{j=1}^{n} x_{j\sigma(j)}$ in $\sum_{ij} A_{ij} x_{ij} X_{ij}$ is $\sum_i A_{i\sigma(i)}$, and must be zero for all permutations σ.

i.e. $\sum_{i=1}^{n} (M'_{i\sigma(i)i\sigma(i)} + \frac{M''}{n}) = 0$

$$\Rightarrow \sum_{i=1}^{n} M'_{i\sigma(i)i\sigma(i)} = -M''$$

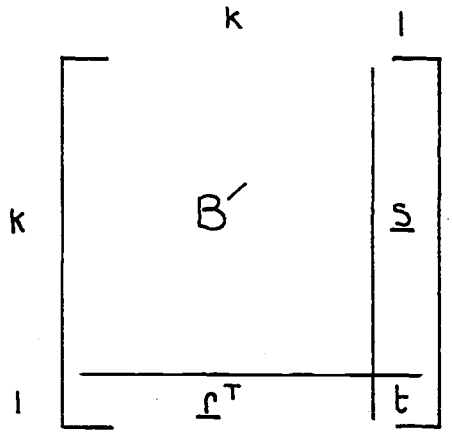
Let $B_{ij} = M'_{ijij}$, then $\sum_{i=1}^{n} B_{i\sigma(i)} = -M''$ (a constant).

We now prove by induction (on the size of a matrix) that any matrix satisfying the above condition is the sum of two matrices, the first having all rows identical, the second all columns identical i.e. $B_{ij} = a_i + b_j$ for some arbitrary a_i and b_j. Call such a matrix "striped".

The hypothesis is clearly true for 1×1 matrices, and so we proceed with the inductive step:-

Suppose the hypothesis is true for $n \leq k$, and B is a (k+1) × (k+1) matrix satisfying $\sum_{i=1}^{k+1} B_{i\sigma(i)} = \lambda$ for any σ.

Partition B as follows:-



Choosing from permutations of (1,2,..,k+1) that map (k+1) to itself, we have $\sum_{j=1}^{k} B'_{j\sigma(j)} = (\lambda - t)$. But therefore B' is "striped" by hypothesis: $B'_{ij} = a_i + b_j$ $i,j \leq k$. Therefore $\sum_{j=1}^{k} (a_j + b_j) = \lambda - t$.

Choosing σ from permutations of (1,2,..,k+1) with $\sigma(k+1) \neq k+1$ we have $\lambda = \sum_{j=1}^{k+1} B_{j\sigma(j)} = \sum_{j=1}^{k} (a_j + b_j) - a_{\sigma^{-1}(k+1)} - b_{\sigma(k+1)} + r_{\sigma(k+1)} + s_{\sigma^{-1}(k+1)}$

thus, since $\sigma(k+1)$ and $\sigma^{-1}(k+1)$ can be chosen independently to be anything less than (k+1):

$$r_j + s_i = a_i + b_j + t$$

Thus $s_i = a_i + (b_1 - r_1 + t)$

$r_j = b_j + (a_1 - s_1 + t)$

So define $a_{k+1} = (a_1 - s_1 + t)$

$b_{k+1} = (b_1 - r_1 + t)$

This clearly makes B "striped" everywhere except possibly at the bottom right hand corner, but $a_{k+1} + b_{k+1} = a_1 + b_1 - s_1 - r_1 + 2t = t$ by putting i = j = 1 in the equations above, so B is "striped" and the induction is complete.

The solution is therefore $M'_{ijij} = a_i + b_j$, and since

$\sum_{i=1}^{n} M'_{i\sigma(i)i\sigma(i)} = -M''$ we have $M'' = -\sum_j (a_j + b_j)$.

Note that there are only (2n-1) degrees of freedom in this solution since $(a_i + c)$, $(b_j - c)$ for any constant c, gives the same value for M'_{ijij}.

Case (ii) expression.

$\sum_{\substack{i\neq k \\ j\neq l}} M'_{ijkl}\, x_{kl} \frac{\partial}{\partial x_{ij}} \det(\underline{x}) = 0$. Since each monomial generated is unique

(by the case (ii) monomial analysis) to the particular term in the determinant differentiated, and the particular value of (i,j,k,l), (and every (i,j,k,l) has at least one non-vanishing monomial as a result of applying $x_{kl}\frac{\partial}{\partial x_{ij}}$ to det($\underline{x}$)) it is clear that

$M'_{ijkl} = 0$ for ($i \neq k$ and $j \neq l$).

Case (iii) expression

$$\sum_{\substack{i \\ j\neq l}} M'_{ijil} x_{il} \frac{\partial}{\partial x_{ij}} \det(\underline{x}) = 0.$$

By the analysis of case (iii) monomials, a monomial in this expression can arise from two distinct monomials in the determinant of opposite sign. From the case (iii) monomial diagram we have:

$M'_{ijil} - M'_{rjrl} = 0$ for all $i \neq r$, $j \neq l$ (since every monomial occurs in the determinant) as these are the coefficients of the resulting monomials.

Thus $M'_{ijil} = M'_{rjrl}$ for $j \neq l$, and for all i,r.

i.e. M'_{ijil} depends only upon j and l, not upon i.

$\underline{\underline{M'_{ijil} = C_{jl}}}$ for $j \neq l$ where C_{jl} are arbitrary.

Case (iv) expression Since this is just the transpose of case (iii) it gives

$M'_{ijkj} = D_{ik}$ for $i \neq k$ where D_{ik} are arbitrary.

Combining case (i) with case (iii) and case (iv) by defining $C_{jj} = b_j$ and $D_{ii} = a_i$, and with case (ii), by noting that there is only the zero solution for M'_{ijkl} unless i=k or j=l gives the general solution:-

$$M'_{ijkl} = \delta_{ik}C_{jl} + \delta_{jl}D_{ik} \text{ and } M'' = -\sum_{i=1}^{n}(C_{ii} + D_{ii})$$

(This can be verified by choosing various of $i = k$, $i \neq k$, $j = 1$ $j \neq 1$ which yield the four cases above.)

The above solution for V_{det} has $2n^2$ arbitrary "constants", and given the "loss" of one degree of freedom in case (ii) it is clearly a vector space of dimension $2n^2-1 = \dim(G_{det})$ by defn.2.20.

Computing $\exp(V_{det})$ (defn. 2.19)

Bearing in mind the doubling of subscripts occasioned by the polynomial in question being the determinant, matrix multiplication of four subscript objects is $(YZ)_{ijmn} = \sum_{kl} Y_{ijkl}Z_{klmn}$

Let us also retain matrix multiplication of two subscript objects as usual.

$\exp\left(\begin{array}{c|c} M' & O \\ \hline O & M''^{T} \end{array}\right) = \left(\begin{array}{c|c} \exp(M') & O \\ \hline O & \exp(M'')^{T} \end{array}\right)$ clearly, since defined by power series.

$M'_{ijkl} = \delta_{ik}C_{jl} + \delta_{jl}D_{ik}$, so define $A_{ijkl} = \delta_{ik}C_{jl}$ and $B_{ijkl} = \delta_{jl}D_{ik}$. Then $M' = A + B$ and $AB = BA$. Thus $e^{M'} = e^{A+B} = e^{A}\cdot e^{B}$ since the latter equality follows directly from the power series for exp, given commutativity. Also therefore $e^{A}\cdot e^{B} = e^{B}\cdot e^{A}$.

$(A^P)_{ijkl} = \delta_{ik}(C^P)_{jl}$ for $p \in \mathbb{N}$ follows inductively,

(where C^P is the P^{th} power of C treated as a two subscript matrix and A^P is the P^{th} power of A treated as a "two subscript pairs" matrix in accordance with the remarks at the start of this short section).

Forming $e^{A} = \sum_{P=0}^{\infty} \frac{1}{P!}A^P$ we have

$$(e^{A})_{ijkl} = \delta_{ik}(e^{C})_{jl}.$$

By identical argument $(e^B)_{ijkl} = \delta_{jl}(e^D)_{ik}$

Thus $(e^{M'})_{ijkl} = \sum_{rs}\delta_{ir}(e^C)_{js}\,\delta_{sl}(e^D)_{rk} = (e^C)_{jl}(e^D)_{ik}$

The exponential function of a matrix maps all matrices onto all non-singular matrices [20] so that e^C and e^D are arbitrary non-singular matrices. When multiplying two $(e^{M'})$ matrices together, clearly the (e^C) parts of each multiply together as "two subscript" matrices without interfering with a similar combination of the (e^D) parts. Closure of the set of exponentials under matrix multiplication does not therefore add any further transformations.

$$(\exp(M")) = \exp(-\sum_{i=1}^{n}(C_{ii} + D_{ii}))$$

$$= (e^{Tr(C+D)})^{-1} \text{ where Tr means trace.}$$

$$= (e^{Tr(C)} \cdot e^{Tr(D)})^{-1} \text{ but } e^{Tr(A)} = \det(e^A) \text{ [20]}$$

$$= (\det(e^C \cdot e^D))^{-1}.$$

Thus since $\tilde{G}_{det} = \exp(V_{det})$ by theorem 2.14 we have

$$\tilde{G}_{det} = \left\{\left(\begin{array}{c|c} T' & O \\ \hline O & T" \end{array}\right) \mid T'_{ijkl} = A_{ik}B_{jl} \text{ where A,B are arbitrary non-singular matrices and } T" = (\det(AB))^{-1}\right\}.$$

Note that the action of $T' \in \tilde{G}'_{det}$ on $\underline{x}$ is as follows:

$T'_{ijkl}x_{kl} = A_{ik}x_{kl}B_{jl}$ i.e. matrix multiplication on both sides by arbitrary non-singular matrices (A and B^T).

This establishes theorem 3.1

Finding V_{per} and computing $\exp(V_{per})$ are very similar.

All of the arguments goes though with per instead of det (and no minus signs by monomials) until the analysis of case (iii) expression).

The equations obtained are almost exactly the same, except that because all monomials in the permanent have positive sign, the two

monomials of the case (iii) monomial analysis that can lead to the same monomial in the case (iii) expression now have the same sign, (rather than the opposite sign as with the determinant). This gives $M'_{ijil} + M'_{rjrl} = 0$ for $j \neq 1$, $i \neq r$ (sum instead of difference).

Consider j,1 fixed: for n>2 we can choose $i \neq k$, $k \neq p$, $p \neq i$ such that

$$-M'_{ijil} = M'_{kjkl} = -M'_{pjpl} = M'_{ijil} = 0 \text{ for } j \neq 1.$$

Case (iv) expression yields $M'_{ijkj} = 0$ for $i \neq k$ since this is merely the transpose of case (iii).

The only degrees of freedom in V_{per} arise from case (ii) in contrast to the determinant, and thus $\dim(G_{per}) = 2n-1$. Combining case (ii) with the other cases, we note that $M'_{ijkl} = 0$ unless i = k and j = 1 thus:

$$M'_{ijkl} = \delta_{ik}\cdot\delta_{jl}(a_i + b_j) \text{ and } M'' = -\sum_{i=1}^{n}(a_i + b_i)$$

is the general solution for V_{per}.

Computing $\exp(\hat{V}_{per})$

Define $A_{ijkl} = \delta_{ik}\delta_{jl}a_i$ and $B_{ijkl} = \delta_{ik}\delta_{jl}b_j$.

Then $M' = A + B$ and $AB = BA$ as before, so

$e^{M'} = e^{A}\cdot e^{B}$.

$(A^P)_{ijkl} = \delta_{ik}\delta_{jl}(a_i)^P \quad P \in \mathbb{N}$

so $(e^A)_{ijkl} = \delta_{ik}\delta_{jl}e^{a_i}$.

Similarly $(e^B)_{ijkl} = \delta_{ik}\delta_{jl}e^{b_j}$.

Thus $(e^{M'})_{ijkl} = \delta_{ik}\delta_{jl}e^{a_i}\cdot e^{b_j}$.

Since e^{a_i} can have any non-zero value, one can regard $\delta_{ik}e^{a_i}$ as an arbitrary non-singular diagonal matrix Λ, and $\delta_{jl}e^{b_j}$ as another, Λ'.

Also $M'' = -\sum_i (a_i + b_i)$ so $e^{M''} = (\prod_i e^{a_i} e^{b_i})^{-1}$

$= (\det(\Lambda'\Lambda))^{-1}$

So $\tilde{G}_{per}$ for $n>2$ is as follows (since closure under matrix multiplication clearly does not give any more transformations):-

$$\tilde{G}_{per} = \left\{ \left(\begin{array}{c|c} T' & 0 \\ \hline 0 & T''T \end{array} \right) \middle| T'_{ijkl} = \Lambda_{ik}\Lambda'_{jl};\ \Lambda',\Lambda \text{ diagonal, non-singular and } T'' = (\det(\Lambda'\Lambda))^{-1} \right\}$$

This establishes theorem 3.2 ⊠ .

Some "discrete" transformations that preserve the permanent are row and column permutations and transposition. These also preserve the determinant, except that the former are a part of $\tilde{G}_{det}$ (i.e. are "continuously generated") and only the latter is a "discrete" transformation for the determinant.

It has been shown that $\dim(G_{det}) = 2n^2-1$ and $\dim(G_{per}) = 2n-1$ (for $n>2$) although there appear to be $2n^2$ and $2n$ arbitrary parameters (respectively) specifying the group members. There is a direct interpretation of this. The group $\tilde{G}_{det}$ is constructed from the identity $(\det(AB))^{-1} \det(A \times B) = \det(X)$ for all non-singular A,B. Although the entries of A,B are $2n^2$ arbitrarily chosen parameters (except for singularity which occurs with probability zero), the action of A,B upon X (i.e. $A \times B$) is the same as the action of λA, $\lambda^{-1}B$ on X for any scalar $\lambda \neq 0$, and this is the "lost" degree of freedom. An almost identical argument applies to $\tilde{G}_{per}$.

For n = 1 clearly the permanent and determinant are identical (and have group dimension 1)! For n = 2 there is the well-known

relationship $\det\begin{pmatrix} x_{11} & x_{12} \\ x_{21} & x_{22} \end{pmatrix} = \operatorname{per}\begin{pmatrix} x_{11} & -x_{12} \\ x_{21} & x_{22} \end{pmatrix}$

It follows from definition 2.22 that $\det_{2\times 2}$ and $\operatorname{per}_{2\times 2}$ are <u>similar</u> and thus have equal group dimension.

Thus $\dim(G_{\operatorname{per}_{2\times 2}}) = \dim(G_{\det_{2\times 2}}) = 2(2^2)-1 = 7.$

(Curiously $\dim(G_{\operatorname{per}_{3\times 3}}) = 2(3)-1 = 5.$)

<u>Corollary 3.1</u> There is no bilinear product Δ that preserves the permanent (for n>2) i.e. $\operatorname{per}(A \Delta B) = \operatorname{per}(A)\cdot\operatorname{per}(B)$.

<u>Proof</u> Suppose the contrary and let $(A \Delta B)_{ij} = \sum_{\alpha\beta\gamma\delta} \psi_{(ij)(\alpha\beta)(\gamma\delta)} A_{\alpha\beta} B_{\gamma\delta}$.

Then $\operatorname{per}(\sum_{\alpha\beta\gamma\delta} \psi_{(ij)(\alpha\beta)(\gamma\delta)} A_{\alpha\beta} B_{\gamma\delta}) = \operatorname{per}(A)\cdot\operatorname{per}(B)$ (*)

For any A with $\operatorname{per}(A) \neq 0$

$\varphi_{(ij)(\gamma\delta)} \overset{\Delta}{=} (\sum_{\alpha\beta} \psi_{(ij)(\alpha\beta)(\gamma\delta)} A_{\alpha\beta})$ must be a non-singular map from $M(n) \to M(n)$, since its null space is included in the translational symmetries of the permanent (this follows immediately from (*)).

Choose $A = I + \varepsilon C$; then for <u>all</u> C :

$$\operatorname{per}((I + \varepsilon C) \Delta X) = \operatorname{per}(I + \varepsilon C)\cdot\operatorname{per}(X).$$

For sufficiently small ε, $\operatorname{per}(I + \varepsilon C) \neq 0$, and so

$$n^2 = \dim\{C \mid (\frac{1}{\operatorname{per}(I + \varepsilon C)})\cdot\operatorname{per}((I + \varepsilon C) \Delta X) = \operatorname{per}(X)\} \leq \dim(G_{\operatorname{per}}) = 2n-1 \quad \text{for } n>2$$

(since φ is non-singular (and will preserve the dimension)), which is a contradiction.

If the permanent could be expressed as a determinant of a linear combination of the indeterminates (i.e. $\operatorname{per}(\underline{x}) = \det(S\underline{x})$ for some S) then the permanent would be easy to compute, contrary to expectation. Marcus and Minc [18] proved the converse by ad hoc

means, but here it follows as an easy corollary of theorems 3.1 and 3.2.

Corollary 3.2 $\nexists S$ such that $per(\underline{x}) = det(S\underline{x})$ for $n>2$.

Proof Assume the contrary: $per(\underline{x}) = det(S\underline{x})$.

if S is singular then $\exists \underline{a} \neq \underline{0}$ such that $S\underline{a} = \underline{0}$.

$\Rightarrow per(\underline{x} + \underline{a}) = det(S(\underline{x} + \underline{a})) = det(S\underline{x}) = per(\underline{x})$

i.e. per has a translational symmetry - contradiction

if S is non-singular then per $\sim$ det (defn. 2.22).

$\Rightarrow \dim(G_{per}) = \dim(G_{det})$ (theorem 2.18).

but $\dim(G_{per}) = 2n-1$ for $n>2$

and $\dim(G_{det}) = 2n^2-1$, which are distinct - contradiction

---

An immediate consequence of theorem 3.1 is that no Gaussian style elimination can be used to assist in evaluating the permanent, since all continuous symmetries of the permanent are diagonal i.e. consist merely of simultaneously scaling the inputs. The proofs of theorems 3.1 and 3.2 show explicitly how a "small" perturbation of the definition of the polynomial destroys all of the useful internal structure. The consequences of the lack of such structure in the case of the permanent are the absence of more restricted forms of structure as exemplified by the corollaries 3.1 and 3.2.

(iii) Generalisations of the Permanent and Determinant

An interesting generalisation of the permanent and determinant are the so-called immanents [17]. These are multilinear polynomials of the form

$\sum_{\sigma \in S_n} \chi(\sigma) \prod_{i=1}^{n} x_{i\sigma(i)}$ where $[x_{ij}]$ is an n×n matrix and $\chi(\sigma)$ is the function of the permutation σ that gives the coefficients. (Clearly, if $\chi(\sigma) = 1$ for any σ, then we have the permanent, and if $\chi(\sigma) = (-1)^{\mathrm{sgn}(\sigma)}$ we have the determinant.) There is a condition on $\chi(\sigma)$ for the above expression to be an immanent, namely that χ be an irreducible character [16,17] of the symmetric group.

Group representation theory gives the following results concerning the irreducible characters of a group. Firstly, distinct irreducible characters are orthogonal, i.e. in the case of the symmetric group $\sum_{\sigma \in S_n} \chi(\sigma) \cdot \psi(\sigma) = 0$ if χ and ψ are distinct irreducible characters [16]. Secondly, for a given character χ, the character of two group elements from the same class in the group is the same [16,17]. Thirdly, the number of distinct irreducible characters of a group is equal to the number of classes in the group [16]. For the symmetric group the second result is that if σ and π are permutations with the same cycle structure, and χ is a character, then $\chi(\sigma) = \chi(\pi)$. Thus any immanent has the property that applying any given permutation to the rows and then the same one to the columns of its matrix leaves it invariant. This is precisely the property necessary and sufficient for a polynomial (in a matrix) to be a combinatorial "counting polynomial" of a graph (specified as its adjacency matrix), i.e. that re-ordering the vertices of the graph in the adjacency matrix

representation leaves the counting function alone: call this "graphical invariance". The first and third results quoted above imply that if $\chi(\sigma)$ is any function that is constant over any class of the symmetric group (i.e. the corresponding multilinear polynomial is graphically invariant) then $\chi(\sigma)$ is a linear combination of the irreducible characters of the symmetric group. Thus the immanents form a basis for all counting polynomials on graphs (in the vector space sense)! Hence any counting polynomial that is a linear combination of a few easy to evaluate immanents is itself easy to evaluate, but the converse is not necessarily true. This prompts the question, "what linear combinations of immanents are easy to evaluate?", and in this context we can look for high symmetry multilinear polynomials, in the hope of finding a "maximum symmetry basis" for the immanents. The following theorem clarifies the situation for all multilinear polynomials of immanent form whether graphically invariant or not.

Theorem 3.3 If $p(\underline{x}) = \sum_{\sigma \in S_n} \phi(\sigma) \prod_{i=1}^{n} x_{i\sigma(i)}$, and has no translational symmetries, and $p(\underline{x}) \neq \lambda \det(\underline{x})$ for any $\lambda \in \mathbb{C}$, then $\tilde{G}'_{\det} \not\subseteq \tilde{G}'_{p}$.

Proof.

Given in [appendix 4].

This shows that the usual form of Gaussian elimination is only applicable to evaluating the determinant.

(iv) Gaussian Elimination and Linear Algebra

In view of the various conjectured and proven uniqueness properties of the determinant, we make brief remarks about symmetries and algorithms in linear algebra.

Corollary 3.3 Let $\underline{subdet}(X)$ be the matrix of (signed) $(n-1)\times(n-1)$ subdeterminants of an $n\times n$ matrix X, i.e. $\underline{subdet}(X)_{ij} = \frac{\partial}{\partial X_{ij}} \det(X)$.

Then $$\frac{A^T \underline{Subdet}(AXB) B^T}{\det(A)\cdot\det(B)} = \underline{Subdet}(X).$$

Proof Application of theorem 2.19 to the determinant.

This corollary is merely a disguised form of an obvious identity: note that the adjoint matrix is the transpose of $\underline{subdet}$, then

$$\frac{B\ Adj(AXB)A}{\det(A)\cdot\det(B)} = Adj(X)$$

i.e. $B(AXB)^{-1}A = X^{-1}$, which forms the basis of the usual algorithm for evaluating the inverse of a matrix. First perform Gaussian elimination on X using A to produce a triangular matrix; secondly, perform transposed Gaussian elimination on the triangular matrix using B to produce a diagonal matrix; thirdly, evaluate the inverse of the diagonal matrix trivially; finally, transform back to the original inverse using A and B. This is clearly a Gaussian elimination style algorithm, and it relies upon $(n-1)\times(n-1)$ subdeterminants of a diagonal matrix being easy to evaluate (compare the remarks at the end of chapter two).

However, now we may take the observation further, by noting firstly that any size subdeterminant of a diagonal matrix is easy to evaluate, and secondly that repeated differentation of the

determinant will leave those symmetries necessary to transform a matrix to diagonal form alone (as in the matrix inversion algorithm) by theorem 2.19. It follows that a simple Gaussian elimination style algorithm exists to compute all $(n-k)\times(n-k)$ subdeterminants of a matrix for any k. This may also be generalised to non-square matrices in the obvious way.

Theorem 3.4 Let Matrix multiplication be denoted by $\underline{\varphi}(\underline{x}) \triangleq \underline{vec}(XY)$ where $\underline{x} = \underline{vec}(X,Y)$ and X,Y are $n\times n$ matrices.

Then $\tilde{G}_{\underline{\varphi}} = \left\{ \left(\begin{array}{c|c} T' & O \\ \hline O & T'' \end{array} \right) \right.$ $\left| T'\underline{x} = \underline{vec}(AXB, B^{-1}YC) \right.$

where A,B,C are arbitrary non-singular matrices;

$T''\underline{z} = \underline{vec}(A^{-1}ZC^{-1})$ where $\underline{z} = \underline{vec}(Z)$ $\}$

Proof - by the usual methods, given in appendix 3.
(Note that a discrete symmetry is transposition of both input matrices and of the result.)

This result shows immediately that the matrix product is not similar to the Hadamard product of two matrices, since the latter has no non-trivial symmetries [31], and the usual dimension argument (theorem 2.18) may be used. Although this is not of any great significance in itself (it shows essentially that $n\times n$ matrix multiplication requires more than n^2 multiplications) it does illustrate the power of the technique. The question as to whether the group action on more general projections than similarity can yield equally powerful results, and the consequences of such for matrix multiplication, are considered in a later section.

(v) A Fast Probabilistic Algorithm for Finding the Continuous Symmetries of a Polynomial

Given a circuit of size C for a polynomial $\underline{P}(\underline{x})$, we outline a fast algorithm (i.e. polynomial time in C) to find $V_{\underline{P}}$. Not surprisingly in view of the conjectured universality of linear algebra, this algorithm generates a collection of linear equations which are then solved for V_P i.e. the problem is reduced to linear algebra.

$$M = \left(\begin{array}{c|c} M' & O \\ \hline O & M''^{T} \end{array}\right) \in V_P \quad \text{satisfies}$$

$\forall \underline{x} \in \mathbb{C}^n \quad (M'\underline{x})\cdot\nabla\underline{P}(\underline{x}) + M''\underline{P}(\underline{x}) = \underline{0}$ by theorem 2.15, which constitutes an infinite set of linear equations for $V_{\underline{P}}$.

Given the circuit of size C for $\underline{P}(\underline{x})$, and using the result of [3], we obtain (by a fast algorithm) a circuit of size proportional to C to compute $\underline{P}(\underline{x})$ and $\nabla\underline{P}(\underline{x})$. Regarding the entries of M' and M" as additional inputs we can thus construct fast a small circuit to compute $(M'\underline{x})\cdot\nabla\underline{P}(\underline{x}) + M''\underline{P}(\underline{x})$ as a function of M',M" and $\underline{x}$. Call this $f(M',M",\underline{x})$.

If (M',M") are in $V_{\underline{P}}$ then the above circuit computes the zero polynomial in $\underline{x}$, otherwise it does not. Schwartz [25] gives a fast probabilistic algorithm for testing whether a circuit computes the zero polynomial or not, and if not, exhibits an $\underline{x}$ where the circuit has a non-zero value. Given any $\varepsilon>0$, this algorithm is accurate with probability $(1-\varepsilon)$ (over infinite fields) where $\varepsilon = \frac{1}{2^N}$ and N is the number of times the randomised step is executed.

The algorithm is as follows:- at any stage there will be a number of linear equations constraining $(M',M") \in V_{\underline{P}}$, obtained

essentially by putting random $\underline{x}$ values into $f(M',M",\underline{x}) = 0$. These equations can be found explicitly, bearing in mind that they are homogeneous and linear, by using finite difference methods.

Initially there are no constraints. At some intermediate stage in the algorithm we choose a basis of pairs (M',M") satisfying the current set of linear constraints, and use Schwartz's method to see if $f(M',M",\underline{x})$ is the zero polynomial in $\underline{x}$ for each basis vector (within some probability ε). If not, this gives an $\underline{x}$ for which $f(M',M",\underline{x}) \neq 0$, which then gives a new linear constraint on $V_{\underline{P}}$ independent of the previous linear constraints, that reduces the possible dimension of $V_{\underline{P}}$ by one; we then go back and repeat the intermediate stage of the algorithm with the new set of linear constraints. If $f(M',M",\underline{x})$ is the zero polynomial (with probability $(1-\varepsilon)$) for all basis vectors (M',M") satisfying the current set of constraints, then we (probably) have a basis for $V_{\underline{P}}$.

Note that the maximum number of times the intermediate stage of the algorithm can be executed is n^2, since this is the maximum dimension of $V_{\underline{P}}$ for any $\underline{P}$ with n inputs, and each time the intermediate stage is executed a new independent linear constraint on $V_{\underline{P}}$ is obtained.

(vi) Conclusions, Conjectures and Open Problems.

The results of this thesis can only strengthen the conjecture that linear algebra is a unique fast computational method in algebraic complexity via some form of reduction, although whether even subdet (defined in section (iii)) is complete for p-circuit via generalised projection is not clear. It may be that some other form of reduction is required.

Similarity is a very weak form of projection that in effect preserves symmetry. In contrast, generalised projection may both create and destroy symmetries, and naive bounds on the change in group dimension that may be accomplished by any projection from an m-input polynomial to an n-input polynomial may be obtained from algebraic geometry by using the affine dimension theorem [15]. Such bounds are very weak however, and are unable even to show that the (n+1) x (n+1) determinant cannot project to the n x n permanent, which is a very old open problem.

Another example where generalised projection is of great importance is matrix multiplication. A bilinear algorithm for matrix multiplication is merely a generalised projection from the pairwise (Hadamard) product to the matrix multiplication polynomial, and the smallest such projection gives the minimum number of multiplications required to perform matrix multiplication, which has important bearing on the overall complexity of matrix multiplication [8].

A general means of assessing the existence or non-existence of generalised projections would therefore be capable of solving two of the most intractable open problems in complexity, namely the pC versus pD question (see the introduction) and the matrix multiplication problem.

As mentioned in the introduction, the symmetries of a polynomial act upon the matrices that give rise to a projection of that polynomial, yielding other matrices that project the polynomial to the same result. In the case of bilinear algorithms for matrix multiplication this is essentially mapping one algorithm into another (invertably) and thus the algorithms fall into equivalence classes under the action of the group. The symmetries of theorem 3·4 have long been known, and the importance of the theorem is that no "new" symmetries exist that might connect a known algorithm to a new fast algorithm via the group action.

Finally, it seems likely that unlike continuous symmetries for which there exists a fast algorithm (outlined in the previous section) to find them, there probably is no such algorithm for generalised projections. Certainly the naive approach, "is there a point in an algebraic set specified by polynomials?", can easily be shown to be an NP-complete problem, and although the existence of projections is a restricted case of this problem, it is one of great scope that in special cases (e.g. matrix multiplication) has been found difficult to solve effectively by algorithm.

Appendix 1.

Theorem Let $\underline{P}(\underline{x})$ be the column matrix of the polynomials $P_1(\underline{x}), P_2(\underline{x}), \ldots, P_m(\underline{x})$ in the indeterminates $\underline{x} = (x_1, x_2, \ldots, x_n)$ over $\bar{\mathbb{Q}}$ (the algebraic closure of the rationals). The circuit size for $\underline{P}$ is the same over $\bar{\mathbb{Q}}$ as over $\mathbb{C}$ (the complex numbers).

Proof

Clearly, the circuit size of $\underline{P}$ over $\mathbb{C}$ is less than or equal to that over $\bar{\mathbb{Q}}$, since a circuit over $\bar{\mathbb{Q}}$ is a circuit over $\mathbb{C}$ (with the obvious extension of the meaning of the arithmetic operations). We show that given any circuit for $\underline{P}$ over $\mathbb{C}$, there is a circuit of equal size for $\underline{P}$ over $\bar{\mathbb{Q}}$.

Suppose we have a circuit C for $\underline{P}$ over $\mathbb{C}$ that is not a circuit for $\underline{P}$ over $\bar{\mathbb{Q}}$, then C must contain members of $\mathbb{C}$ not in $\bar{\mathbb{Q}}$ (and are therefore transcendental over $\bar{\mathbb{Q}}$). Call these $\underline{\pi} = (\pi_1, \pi_2, \ldots, \pi_k)$. From C construct a new circuit C' by replacing the constants $\underline{\pi}$ by additional inputs $\underline{y} = (y_1, y_2, \ldots, y_k)$, and denote by $\underline{f}(\underline{x},\underline{y})$ the column matrix of rational functions of $(\underline{x},\underline{y})$ that C' computes. Note that in particular $\underline{f}(\underline{x},\underline{\pi}) = \underline{P}(\underline{x})$, and that C' computes over $\bar{\mathbb{Q}}$ as it involves no transcendental constants.

We now show that there always exists $\underline{a} = (a_1, a_2, \ldots, a_k)$ with a_i in $\bar{\mathbb{Q}}$ for all i, such that $\underline{f}(\underline{x},\underline{a}) = \underline{P}(\underline{x})$, and therefore the circuit C", obtained from C by replacing $\underline{\pi}$ by $\underline{a}$, computes $\underline{P}(\underline{x})$ over $\bar{\mathbb{Q}}$ and is exactly the same size as C.

To obtain such an $\underline{a}$ we need to solve $\forall \underline{x}\ \underline{f}(\underline{x},\underline{y}) = \underline{P}(\underline{x})$ for $\underline{y}$ over $\bar{\mathbb{Q}}$. Take $f_i(\underline{x},\underline{y}) = g_i(\underline{x},\underline{y})/h_i(\underline{x},\underline{y})$ where g_i and h_i are polynomials. Then the equations to be solved become equivalent to $\forall \underline{x}, i\ \ g_i(\underline{x},\underline{y}) - h_i(\underline{x},\underline{y}).P_i(\underline{x}) = 0$, and these in turn are equivalent to a series of polynomial equations $q_i(\underline{y}) = 0$ for $1 \leq i \leq t$, obtained by equating the coefficient of each $\underline{x}$ monomial to zero.

Let J be the ideal† generated by the polynomials $q_1, q_2 \ldots q_t$ (in the ring $\bar{\mathbb{Q}}[y_1, y_2, \ldots, y_k]$ of multivariate polynomials in $\underline{y}$ over $\bar{\mathbb{Q}}$). Hilbert's Nullstellensatz [15 p.125] states that over an

† The facts necessary to understand this argument are between p.103 and p.131 of [15], for example.

algebraically complete field every proper ideal† of a polynomial ring has a zero. Thus if J is a proper ideal, there exists $\underline{a}$ in $\bar{\mathbb{Q}}^k$ with $q_1(\underline{a}) = q_2(\underline{a}) = \ldots = q_t(\underline{a}) = 0$, and thus $\underline{f}(\underline{x},\underline{a}) = \underline{P}(\underline{x})$ and the theorem is proved.

Suppose J is not a proper ideal, then 1 is in J and thus there are polynomials $r_i(\underline{y})$ over $\bar{\mathbb{Q}}$ with $1 = \sum_{i=1}^{t} r_i(\underline{y})q_i(\underline{y})$, since the q_i's generate† J. However, over the extension field $\mathbb{C}$ of $\bar{\mathbb{Q}}$ we have $\underline{f}(\underline{x},\underline{\pi}) = \underline{P}(\underline{x})$ and thus (by the same argument as for $\underline{a}$ except over $\mathbb{C}$) $q_1(\underline{\pi}) = q_2(\underline{\pi}) = \ldots = q_t(\underline{\pi}) = 0$ which is in contradiction to $1 = \sum_{i=1}^{t} r_i(\underline{\pi})q_i(\underline{\pi})$, and so J must be a proper ideal. ∎

Note that the above argument could equally well be applied to formula size, and that it is not restricted to $\bar{\mathbb{Q}}$ and $\mathbb{C}$: any algebraically complete field contained in another field would do.

Appendix 2. An $O(n2^n)$ circuit for the $n \times n$ permanent.

The permanent of $[x_{ij}]$ is the coefficient of $\prod_i y_i$ in the polynomial $p(y) = \prod_k (\sum_i x_{ki} y_i)$. Using the formal definition of differentiation ($f'(x) = \lim_{\varepsilon \to 0} \frac{f(x+\varepsilon) - f(x-\varepsilon)}{2\varepsilon}$) repeatedly to find $\frac{\partial}{\partial y_1} \frac{\partial}{\partial y_2} \ldots \frac{\partial}{\partial y_n}$ of the above polynomial p, and setting $y_1 = y_2 = \ldots = y_n = 0$ gives the permanent as the sum of p evaluated at the corners of a small hypercube placed symmetrically about the origin, with alternating signs on the corners.

i.e. $\mathrm{per}(\underline{x}) = \lim_{\varepsilon \to 0} \sum_{\underline{u} \in \underline{\mathrm{cube}}} \frac{\mathrm{sgn}(\underline{u}) p(\varepsilon \underline{u})}{2^n \varepsilon^n}$ where cube is the set of vectors whose components are all ±1. However, $p(y)$ is homogeneous and of degree n, and so $p(\varepsilon \underline{y}) = \varepsilon^n p(\underline{y})$, obviating the limiting process. Equally, $p(-\underline{u}) = (-1)^n p(\underline{u})$, halving the number of terms. ($\mathrm{sgn}(\underline{u})$ is ±1 for an even/odd number of +1 components in $\underline{u}$).

i.e. $\mathrm{per}(\underline{x}) = \sum_{\underline{u} \in \underline{\mathrm{cube}/2}} \frac{\mathrm{sgn}(\underline{u})}{2^{n-1}} \prod_k \left(\sum_i x_{ki} u_i \right)$.

Note that the matrix multiplication requires no multiplications, since the components of $\underline{u}$ are ±1, and merely dictate whether to add or subtract the corresponding entries of $\underline{x}$, thus giving a circuit of size $O(n2^n)$ that is very simple. Indeed, empirically, any NP or #P complete problem has a similar character to this one: there is an obvious solution (in this case just evaluate the formula for the permanent directly) and then a simple way of doing a little better by not repeating computational effort (for example the above circuit for the permanent), and nothing else. This is yet more evidence of intractability.

Appendix 3

Theorem Matrix multiplication: $\underline{\phi}(\underline{x}) \triangleq \underline{\mathrm{vec}}(XY)$ where $\underline{x} = \underline{\mathrm{vec}}(X,Y)$ (and X,Y are $n \times n$ matrices) has

$$\tilde{G}_{\underline{\phi}} = \left\{ \left(\begin{array}{c|c} T' & 0 \\ \hline 0 & T''^{T} \end{array} \right) \;\middle|\; \begin{array}{l} T'\underline{x} = \underline{\mathrm{vec}}(AXB, B^{-1}YC) \\ \text{where } A,B,C \text{ are arbitrary non-singular matrices;} \\ T''\underline{z} = \underline{\mathrm{vec}}(A^{-1}ZC^{-1}) \text{ where } \underline{z} = \underline{\mathrm{vec}}(Z) \end{array} \right\}$$

Proof. Using theorem 2.15 to find $V_{\underline{\phi}(\underline{x})}$:-

First put $\underline{x} \longrightarrow \left(\frac{\underline{x}}{\underline{y}}\right)$ where $\underline{x} = \underline{\mathrm{vec}}(X)$, $\underline{y} = \mathrm{vec}(Y)$.

Partition $M' = \left(\begin{array}{c|c} A & S \\ \hline S' & A' \end{array} \right)$, and $\nabla \longrightarrow \left(\frac{\nabla \underline{x}}{\nabla \underline{y}}\right)$. $M'' = -k$.

Then $\left\{ \left(\begin{array}{c|c} A & S \\ \hline S' & A' \end{array} \right) \left(\frac{\underline{x}}{\underline{y}}\right) \right\} \left(\frac{\nabla \underline{x}}{\nabla \underline{y}}\right) \underline{\phi}(\underline{x},\underline{y}) + M''\underline{\phi}(\underline{x},\underline{y}) = \underline{0}.$

i.e. $(A\underline{x} + S\underline{y}) . \nabla \underline{x}\, \underline{\phi}(\underline{x},\underline{y}) + (S'\underline{x} + A'\underline{y}) . \nabla \underline{y}\, \underline{\phi}(\underline{x},\underline{y}) + M''\, \underline{\phi}(\underline{x},\underline{y}) = \underline{0}$.

$$\sum_{ijkl} \left[(A_{ijkl}x_{kl} + S_{ijkl}y_{kl}) \frac{\partial}{\partial x_{ij}} + (S'_{ijkl}x_{kl} + A'_{ijkl}y_{kl}) \frac{\partial}{\partial y_{kl}} \right] \sum_{q} x_{pq}y_{qr}$$

$$+ \sum_{stu} M''_{prst}\, x_{su}y_{ut} = 0.$$

$$\sum_{klq} \left[(A_{pqkl}x_{kl} + S_{pqkl}y_{kl})y_{qr} + (S'_{qrkl}x_{kl} + A'_{qrkl}y_{kl})x_{pq} \right]$$

$$- \sum_{stu} k_{prst}x_{su}y_{ut} = 0 .$$

$$\sum_{klq} (A_{pqkl}x_{kl}y_{qr} + S_{pqkl}y_{kl}y_{qr} + S'_{qrkl}x_{kl}x_{pq} + A'_{qrkl}y_{kl}x_{pq} - k_{prkl}x_{kq}y_{ql}) = 0$$

Considering the terms of degree 2 in $\underline{x}$:

$$\boxed{\forall p,r \;:\; \sum_{klq} S'_{qrkl}x_{kl}x_{pq} = 0} \qquad ①$$

& for $\underline{y}$:

$$\boxed{\forall p,r \;:\; \sum_{klq} S_{pqkl}y_{kl}y_{qr} = 0} \qquad " \quad ②$$

Leaving :

$$\forall p,r \quad : \quad \sum_{klq} (A_{pqkl} x_{kl} y_{qr} + A'_{qrkl} x_{pq} y_{kl} - K_{prkl} x_{kq} y_{ql}) = 0 \qquad (3)$$

Coeff. of $x_{\alpha\beta} . x_{\gamma\delta}$ in (1)

$$\frac{\partial}{\partial x_{\alpha\beta}} \frac{\partial}{\partial x_{\gamma\delta}} \sum_{klq} S'_{qrkl} x_{kl} x_{pq} = 0$$

$$\therefore \frac{\partial}{\partial x_{\alpha\beta}} \sum_{klq} S'_{qrkl} (\delta_{\gamma k} \delta_{\delta l} x_{pq} + \delta_{\gamma p} \delta_{\delta q} x_{kl}) = 0$$

$$\therefore \sum_{klq} S'_{qrkl} (\delta_{\gamma k} \delta_{\delta l} \delta_{\alpha p} \delta_{\beta q} + \delta_{\gamma p} \delta_{\delta q} \delta_{\alpha k} \delta_{\beta l}) = 0 .$$

$$\therefore \quad \forall_{\alpha,\beta,\gamma,\delta}^{p,r} : S'_{\beta r\gamma\delta} . \delta_{\alpha p} + S'_{\delta r\alpha\beta} . \delta_{\gamma p} = 0 \qquad (*1)$$

Coeff. of $y_{\alpha\beta} y_{\gamma\delta}$ in (2) :

$$\forall_{\alpha,\beta,\gamma,\delta}^{p,r} \quad : \quad S_{p\gamma\alpha\beta} \delta_{\delta p} + S_{p\alpha\gamma\delta} \delta_{\beta r} = 0 \qquad (*2)$$

Coeff. of $x_{\alpha\beta} y_{\gamma\delta}$ in 3 :

$$\forall_{\alpha,\beta,\gamma,\delta}^{p,r} \quad : \quad A_{p\gamma\alpha\beta} . \delta_{r\delta} + A'_{\beta r\gamma\delta} . \delta_{\alpha p} - K_{pr\alpha\delta} . \delta_{\beta\gamma} = 0 \qquad (*3)$$

Consider (*2)

(i) $\beta = r$: $S_{p\gamma\alpha\beta} . \delta_{\delta\beta} + S_{p\alpha\gamma\delta} = 0$.

$\delta \neq \beta$: $S_{p\alpha\gamma\delta} = 0$

i.e. $S = 0$/ $S' = 0$

By symmetry with (*1)

Consider (*3) :

(i) $r = \delta$:

$$A_{p\gamma\alpha\beta} + A'_{\beta r\gamma r} \cdot \delta_{\alpha p} - K_{pr\alpha r} \cdot \delta_{\beta\gamma} = 0$$

$\alpha \neq p$, $\beta \neq \gamma$: $A_{p\gamma\alpha\beta} = 0$ (.1)

$\alpha = p$:

$$A_{\alpha\gamma\alpha\beta} + A'_{\beta r\gamma r} - K_{\alpha r\alpha r} \cdot \delta_{\beta\gamma} = 0$$

$\beta \neq \gamma$: $A_{\alpha\gamma\alpha\beta} + A'_{\beta r\gamma r} = 0$ (.2)

$$A_{\alpha\beta\alpha\beta} + A'_{\beta r\beta r} - K_{\alpha r\alpha r} = 0 \qquad (.3)$$

$\beta = \gamma$:

$$A_{p\beta\alpha\beta} + A'_{\beta r\beta r} \cdot \delta_{\alpha p} - K_{pr\alpha r} = 0$$

$p \neq \alpha$: $A_{p\beta\alpha\beta} - K_{pr\alpha r} = 0$ (.4)

(ii) $r \neq \delta$:

$$A'_{\beta r\gamma\delta} \cdot \delta_{\alpha p} - K_{pr\alpha\delta} \cdot \delta_{\beta\gamma} = 0$$

$\beta \neq \gamma$, $r \neq \delta$: $A'_{\beta r\gamma\delta} = 0$ .5

$r \neq \delta$: $A'_{\beta r\beta\delta} - K_{\alpha r\alpha\delta} = 0$ .6

$\alpha \neq p$, $r \neq \delta$: $- K_{pr\alpha\delta} = 0$ .7

(.1) , (.5) , (.7) are self explanatory.
All of .1 - .7 are independent of eachother.

Consider (.6) :

$r \neq \delta$:

$$A'_{\beta r \beta \delta} = K_{\alpha r \alpha \delta}$$

independent of α. independent of β.

$\therefore$

$A'_{\beta r \beta \delta} = C'_{r\delta}$ (arbitrary) for $r \neq \delta$

$K_{\alpha r \alpha \delta} = C'_{r\delta}$ (n(n-1) parms.)

Consider (.4) :

$A_{p\beta\alpha\beta} = C_{p\alpha}$ for $p \neq \alpha$

$K_{pr\alpha r} = C_{p\alpha}$ n(n-1) parms.

Consider (.2)

$A_{\alpha\gamma\alpha\beta} = C''_{\gamma\beta}$ for $\gamma \neq \beta$

$A'_{\beta r \gamma r} = C''_{\gamma\beta}$ n(n-1) parms.

Consider (.3) :

$$A_{\alpha\beta\alpha\beta} + A'_{\beta r\beta r} = K_{\alpha r\alpha r} \triangleq f_{\alpha r}$$

($A_{\alpha\beta\alpha\beta} \to g_{\alpha\beta}$, $A'_{\beta r\beta r} \to h_{\beta r}$)

$K_{\alpha r\alpha r}$ independent of β.

$$g_{\alpha\beta} + h_{\beta r} = f_{\alpha r} \qquad \text{fix } \beta = 1$$

$$g_{\alpha 1} + h_{1r} = f_{\alpha r}$$

Call this $\tilde{g}_\alpha$ $\tilde{h}_r$

$\therefore\ f_{\alpha r} = \tilde{g}_\alpha + \tilde{h}_r$

$\therefore\ g_{\alpha\beta} + h_{\beta r} = \tilde{g}_\alpha + \tilde{h}_r$

$$(g_{\alpha\beta} - \tilde{g}_\alpha) = -(h_{\beta r} - \tilde{h}_r) \triangleq l_\beta$$

independent of r independent of α

$\therefore$

$$g_{\alpha\beta} = \tilde{g}_\alpha + l_\beta$$
$$h_{\beta r} = \tilde{h}_r - l_\beta$$
$$f_{\alpha r} = \tilde{g}_\alpha + \tilde{h}_r$$

$A_{\alpha\beta\alpha\beta} = g_\alpha + l_\beta$

$A'_{\beta r\beta r} = h_r - l_\beta$

$K_{\alpha r\alpha r} = g_\alpha + h_r$

3n parameters but $l_\beta+c$, h_r+c, $g_\alpha-c$ gives the same value so 3n-1 independent ones

Total dimension = $3n(n-1) + 3n-1 = 3n^2-1$.

The transformations of the theorem certainly preserve matrix multiplication since $A^{-1}(AXB)(B^{-1}YC)C^{-1} = XY$, and these also correspond to a group dimension of $3n^2-1$ (since A and B can be inversely scaled to λA, $\lambda^{-1}B$ along with B^{-1} and C being inversely scaled to λB^{-1}, $\lambda^{-1}C$, without changing the linear transformation of the inputs). Thus the transformations of the theorem are the only ones (as they are closed under the group product).

ndix 4

Theorem If $p(\underline{x}) = \sum_{\sigma \in S_n} \phi(\sigma) \prod_{i=1}^{n} x_{i\sigma(i)}$ and is irredundant, and $p(\underline{x}) \neq \lambda \det(\underline{x})$ for some $\lambda \in \mathbb{C}$ then $\tilde{G}'_{det} \not\subseteq \tilde{G}'_{p}$.

Proof

The only solution of $\forall A,B \quad f(AB) = f(A).f(B)$ for scalar functions of square matrices is $f(X) = g(\det(X))$ where g has the property $g(xy) = g(x).g(y)$ for all scalar x,y [1 p.350], and the only analytic solution of the latter is $g(x) = x^{\alpha}$ for some constant α[1 p.39]. Thus the only solution of the former equation that is a multilinear polynomial, is $\det(X)$.

Suppose $\tilde{G}'_{det} \subseteq \tilde{G}'_{p}$, then $T''_{A,B}\, p(AXB) = p(X)$, where the subscription by A,B denotes the functional dependence of T''(thm.2.4). Thus $T''_{AB,I}\, p(ABX) = p(X)$, $T''_{A,I}\, p(A(BX)) = p(BX)$ and $T''_{B,I}\, p(BX) = p(X)$, giving $T''_{AB,I} = T''_{A,I}\, T''_{B,I}$ by uniqueness (theorem 2.4). Thus $T''_{A,I} = \det(A)^{\alpha}$ for some constant α (since G_p is smooth (thm.2.10)). Therefore $\det(A)^{\alpha}\, p(AX) = p(X)$, and putting $A = X^{-1}$ gives $p(X) = p(I)\, \det(X)^{-\alpha}$. For multilinearity $\alpha = -1$; call $p(I) = \lambda \in \mathbb{C}$. Thus $p(X) = \lambda \det(X)$ ∎ .

REFERENCES

[1] Aczel, J. Lectures on Functional Equations and their applications. Academic Press, 1966.

[2] Aho, A.V., Hopcroft, J.E. and Ullman, J.D. The Design and Analysis of Computer Algorithms. Addison-Wesley, 1974.

[3] Baur, W. and Strassen, V. The Complexity of Partial Derivatives. Theoretical Computer Science, 22:3 (1983) 317-332.

[4] Borel, A. Linear Algebraic Groups. Benjamin, 1969.

[5] Borodin, A. and Munro, I. The Computational Complexity of Algebraic and Numeric Problems. American Elsevier, 1975.

[6] Chevalley, C. Theory of Lie Groups. Princeton University Press, 1946.

[7] Cook, S.A. The Complexity of Theorem Proving Procedures. Proc. 3rd ACM Symp. on Theory of Computing, (1971) 151-158.

[8] Coppersmith, D. and Winograd, S. On the Asymptotic Complexity of Matrix Multiplication. Proc. 22nd IEEE Symp. on Foundations of Computer Science (1981) 82-90.

[9] Even, S. Graph Algorithms. Pitman, 1979.

[10] Garey, M.R. and Johnson, D.S. Computers and Intractability - a Guide to the Theory of NP-Completeness. Freeman, 1979.

[11] Hyafil, L. On the Parallel Evaluation of Multivariate Polynomials. SIAM J. on Computing 8:2 (1976) 120-123.

[12] Jerrum, M.R. Applications of Algebraic Completeness to Problems of Network Reliability and Monomer-Dimer Systems. University of Edinburgh Computer Science Department internal report CSR-45-79, 1979.

[13] Jerrum, M.R. and Snir, M. Some Exact Complexity Results for Straight-Line Computations over Semi-Rings. JACM 29:3 (1982) 874-897.

[14] Kalorkoti, K.A. A Lower Bound on the Formula Size of Rational Functions. Lecture Notes in Computer Science, Springer-Verlag, Vol. 140 (1982) 330-338.

[15] Kendig, K. Elementary Algebraic Geometry. Springer-Verlag, 1977.

[16] Ledermann, W. Introduction to Group Characters. Cambridge University Press, 1977.

[17] Littlewood, D.E. The Theory of Group Characters. Oxford University Press, 1940.

[18] Marcus, M. and Minc, H. On the Relation Between the Determinant and the Permanent. Illinois J. Math. 5 (1961) 376-381.

[19] Moon, J.W. Counting Labelled Trees. Canadian Math. Congress, Montreal (1970).

[20] Miller, W. Symmetry Groups and Their Applications. Academic Press, 1973.

[21] Minc, H. Permanents. Addison Wesley, 1978.

[22] Mumford, D. Algebraic Geometry I. Springer-Verlag, 1976.

[23] Nijenhuis, A. and Wilf, H.S. Combinatorial Algorithms. Academic Press, 1975.

[24] Ryser, H.J. Combinatorial Mathematics. Carus Math. Monograph No. 14, 1963.

[25] Schwartz, J.T. Probabilistic Algorithms for Verification of Polynomial Identities. Tech. Rept. 604, Dept. of Computer Science, Courant Inst. Math. Sci., New York Univ. (1978).

[26] Serre, J.P. Lie Algebras and Lie Groups. Benjamin, 1965.

[27] Skyum, S. A Measure in Which Boolean Negation is Exponentially Powerful. Comp. Sci. Departmental Report, Aarhus University, 1982.

[28] Skyum, S. and Valiant, L.G. A Complexity Theory Based on Boolean Algebra. Proc. 22nd IEEE Symp. on Foundations of Computer Science (1981) 244-253.

[29] Strassen, V. Vermeidung von Divisionen. J. Reine und Angewandte Mathematik 264 (1973) 182-202.

[30] Strassen, V. Die Berechanungskomplexetät von elementary-symmetrischen Funktionen und von Interpolationskoeffizienten. Numerische Mathematik vol. 20 No. 3 (1973).

[31] Sturtivant, C. Generalised Symmetries of Polynomials in Algebraic Complexity. Proc. 23rd IEEE Symp. on Foundations on Computer Science (1982) 72-79.

[32] Valiant, L.G. Negation Can be Exponentially Powerful. Proc. 11th ACM Symp. on Theory of Computing (1979).

[33] Valiant, L.G. The Complexity of Computing the Permanent. Theor. Comp. Sci. 8 (1979) 189-201.

[34] Valiant, L.G. Completeness Classes in Algebra. Proc. 11th ACM Symp. on Theory of Computing (1979) 249-261.

[35] Valiant, L.G. Reducibility by Algebraic Projections. L'Enseignement Mathématique, Monographie No. 30 (1982) 365-380.

[36] Valiant, L.G. Negative Results on Counting. 4th GI Conference on Theoretical Computer Science (1979).

[37] Valiant, L.G. The Complexity of Combinatorial Computations: an Introduction. Edinburgh University Dept. of Computer Science internal report CSR-30-78 (1978).