Algorithms for Symmetric Differential Systems

Elizabeth L. Mansfield

Institute of Mathematics and Statistics,

University of Kent,

Canterbury, CT2 7NF, U.K. *

April 26, 2001

Abstract

Over determined systems of partial differential equations may be studied using differential-elimination algorithms as a great deal of information about the solution set of the system may be obtained from the output. Unfortunately, many systems are effectively intractable by these methods due to the expression swell incurred in the intermediate stages of the calculations. This can happen when, for example, the input system depends on many variables and is invariant under a large rotation group, so that there is no natural choice of term ordering in the elimination and reduction processes.

This article describes how systems written in terms of the differential invariants of a Lie group action may be processed in a manner analogous to differential-elimination algorithms. The algorithm described terminates and yields, in a sense which we make precise, a complete set of representative invariant integrability conditions which may be calculated in a "critical pair" completion procedure. Further, we discuss some of the profound differences between algebras of differential invariants and standard differential algebras. We use the new, regularised moving frame method of Fels and Olver [11, 12] to write a differential system in terms of the invariants of a symmetry group. The methods described have been implemented as a package in MAPLE.

The main example discussed is the analysis of the $2+1$ d'Alembert-Hamilton system,

$$\begin{aligned} u_{xx} + u_{yy} - u_{zz} &= f(u), \\ u_x^2 + u_y^2 - u_z^2 &= 1. \end{aligned} \tag{1}$$

We demonstrate the classification of solutions due to Collins [7] for $f \neq 0$ using the new methods.

* *Email address:* `E.L.Mansfield@ukc.ac.uk`

1 Introduction

Consider the $n+1$ d'Alembert-Hamilton system [14],

$$\begin{aligned} u_{x_1x_1} + u_{x_2x_2} + \ldots + u_{x_nx_n} - u_{tt} &= f(u), \\ u_{x_1}^2 + u_{x_2}^2 + \ldots + u_{x_n}^2 - u_t^2 &= 1, \end{aligned} \tag{2}$$

which is invariant under the usual action of the Poincaré group in $(\mathbf{x}, t)$-space. One of many questions one can ask about (2) is, "for which choices of the arbitrary function f does a solution exist?" In [14] it is shown that f necessarily takes the form $f(u) = g'(u)/g(u)$ where g is a polynomial of degree at most n. Exact solutions are known if

$$f(u) = \frac{m}{u}, \qquad m = 0, 1, 2, \ldots, n,$$

when for example $u = \sqrt{x_1^2 + \ldots + x_m^2}$ $(m \neq 0)$, or $u = x_1$ if $m = 0$, and it is conjectured that f must in fact be of this more restricted form for solutions, smooth in some domain, to exist [8].

Algorithms which determine the consistency of over determined systems, and yield either a characteristic decomposition, or an involutive basis or differential Gröbner basis form are known and implemented [3, 17, 19, 23, 35, 39] and have proved useful in applications (cf. [22, 24] and references therein). Such algorithms will, in principle, find the necessary conditions that f must satisfy, along with a great deal of other information about the solution space. However, when they are applied to the system (2) for $n \geq 3$ the expression swell is enormous. The conditions on f are not found within the available memory, and the intermediate expressions contain so many summands that little insight is to be had. One reason is thought to be that there is no natural choice of term ordering which can be used in the elimination and reduction calculations as the independent variables appear on an equal footing, a consequence of the invariance of the system under the Poincaré group. If this is the case, then in order to calculate integrability conditions systematically with a reasonable chance of success, it may be helpful to first "divide out" the symmetry. More generally, the solution space of invariant systems has a structure by virtue of the fact that the group acts on it. This structure is often exploited in applications such as the integration of invariant ODEs, or the design of numerical approximations which preserve some important qualitative feature. Analysing a system in terms of this structure seems likely to improve our understanding of the solution spaces of physically important or mathematically interesting examples.

The central idea explored in this article is to write the system in terms of the invariants of the group action and to perform a Gröbner basis type calculation on the invariantised system. The following simple example from the classical literature illustrates the idea in essence. The Halphen-Darboux system of ordinary differential equations ([1], p. 336)

$$\begin{aligned}\frac{\mathrm{d}w_1}{\mathrm{d}t} &= w_2w_3 - w_1(w_2+w_3)\\ \frac{\mathrm{d}w_2}{\mathrm{d}t} &= w_1w_3 - w_2(w_1+w_3)\\ \frac{\mathrm{d}w_3}{\mathrm{d}t} &= w_1w_2 - w_3(w_1+w_2)\end{aligned} \tag{3}$$

is invariant under permutations of the w_i. One generating set of invariants of the permutation group is

$$\begin{aligned}s_1 &= w_1+w_2+w_3\\ s_2 &= w_1w_2+w_2w_3+w_1w_3\\ s_3 &= w_1w_2w_3\end{aligned} \tag{4}$$

In terms of the dependent variables s_1, s_2, s_3 the system is

$$\begin{aligned}\frac{\mathrm{d}s_1}{\mathrm{d}t} &= -s_2\\ \frac{\mathrm{d}s_2}{\mathrm{d}t} &= -6s_3\\ \frac{\mathrm{d}s_3}{\mathrm{d}t} &= -4s_1s_3+s_2^2\end{aligned} \tag{5}$$

Eliminating s_3 from the third equation using the second and then eliminating s_2 using the first equation, we obtain the system in the desired triangular form

$$\begin{aligned}s_2+\frac{\mathrm{d}s_1}{\mathrm{d}t} &= 0\\ 6s_3-\frac{\mathrm{d}^2s_1}{\mathrm{d}t^2} &= 0\\ \frac{\mathrm{d}^3s_1}{\mathrm{d}t^3}-6\left(\frac{\mathrm{d}s_1}{\mathrm{d}t}\right)^2+4s_1\frac{\mathrm{d}^2s_1}{\mathrm{d}t^2} &= 0\end{aligned} \tag{6}$$

The analytic solutions of systems (5) and (6) are the same. The ordinary differential equation for s_1 is the *Chazy* equation, whose solutions are well understood [6], and thus all the invariants can be written down in terms of solutions of the Chazy equation. Finally, the w_i are obtained from the s_j as the roots of the cubic

$$w^3 - s_1w^2 + s_2w - s_3 = 0.$$

In this article we are concerned with differential systems which are invariant under a Lie group. Amazingly, complete sets of differential invariants, the relations between them, and maximal sets of invariant differential operators, are unknown for even the most physically significant groups such as as the Euclidean, Poincaré and symplectic groups acting on $\mathbb{R}^n$ in the standard way, for $n \geq 3$ ([30] p. 173, although some are known [13]). Despite this, a theory for analyzing over determined systems of such invariants is not premature. Based on a profound understanding of the Cartan Equivalence Method, Fels and Olver have formulated a new, regularised version of the so-called moving frame algorithm which yields

> the construction and classification of differential invariants and invariant differential operators on jet bundles ... complete classifications of generating systems of differential invariants, explicit commutation formulae for the associated invariant differential operators, and a general classification theorem for syzygies of the higher order differentiated differential invariants. [12]

The chief advantages of the Fels and Olver method are, that the method applies to systems in both the usual and the exterior calculus, the moving frame can be obtained in examples other than those already calculated by Cartan himself, and many of the formulae, used here in the analysis of over determined systems of invariants, involve only the group infinitesimals, that is, the associated Lie algebra. This last point means that a great deal of the calculation can be performed automatically using commercial computer algebra systems such as MAPLE [28]. A brief outline of the Fels and Olver moving frame, adequate for the purposes of this article, is given in §3.

Section §3 concludes with a discussion of the profound differences that exist between the standard differential algebras and algebras of invariants.

The systematic search for integrability conditions is at the heart of, or at least embedded implicitly into, algorithms which seek to complete an over determined differential system to either a characteristic decomposition form, or an involutive basis, or a differential Gröbner basis. We restrict our discussion to the Kolchin-Ritt algorithm which is the simplest of the differential-elimination algorithms. In §2 an introduction to the Kolchin-Ritt algorithm, and the nature of its output, is given.

Because the invariant differential operators do not commute, and for a variety of other reasons to be explained later, the generalisation of the Kolchin-Ritt algorithm to invariantised differential systems is not completely straightforward. We begin the discussion in §4 where a naive translation of the Kolchin-Ritt algorithm is carried out on a simple example. This shows some of the features that need to be taken into account for a differential-elimination algorithm on systems of invariants to be robust and effective in practice. We take for our main example, discussed in §5, the system

$$\begin{aligned} u_{xx} + u_{yy} + u_{zz} &= f(u), \\ u_x^2 + u_y^2 + u_z^2 &= 1, \end{aligned} \tag{7}$$

which is invariant under the Euclidean group. This example reveals further features. The method developed yields “complete information” for this system in the sense that we show that no further integrability conditions exist.

In §6 we give the technical details necessary for an implementation of an invariantised Kolchin-Ritt algorithm for general systems of polynomial type. We assume that the induced action of the Lie Algebra is of rational type and the moving frame is of polynomial type. The algorithm described terminates and yields, in a sense which we make precise, a complete set of representative invariant integrability conditions which may be obtained in a “critical pair” completion procedure. The notions of coherence and a “Buchberger second criterion” for our algorithm are discussed, as is the role of the so-called “differential syzygies of the frame”. Some conjectures are made for further study. We conclude with a comparison with the earlier method of Lisle [21] and a brief description of an implementation of our methods in Maple.

2 The Kolchin-Ritt Algorithm

The Kolchin-Ritt algorithm is one translation to *nonlinear* differential systems of Buchberger's algorithm for a Gröbner basis [2, 10] of a set of polynomials. Much of the power of Gröbner bases can be harnessed for differential systems, allowing one to obtain “complete” sets of integrability conditions for many classes of differential systems, and many useful properties of the solution space of the system may be deduced from the output of this and related algorithms. In this section the concepts and formulae needed to explain the Kolchin-Ritt algorithm are given briefly. They will be adapted to invariantised differential systems in §6, so some detail is required. For the full set of formulae and further details we refer to [23]. The Kolchin-Ritt algorithm has been implemented, and the manual of the computer algebra package `diffgrob2`, [25] contains many examples of how the output may be used. For a simple discussion of the algorithm and its many applications for linear systems see [22]. A brief discussion and examples of the algorithm on nonlinear systems may be seen in [26].

More recent advances in the analysis of this and related algorithms, with a variety of both practical and theoretical innovations, have been given by [3, 19]. In particular, the paper by E. Hubert, in the language of Kolchin, gives the most recent, and rigorous results. The MAPLE library package `diffalg` implements these innovations. A different development, pioneered by G. Reid [35] and his co-workers, which is applicable to systems not necessarily of polynomial type, has its roots in classical methods of analysis of differential systems by Riquier and Janet. These classical methods have the Kolchin-Ritt algorithm embedded, albeit implicitly. A third strand of algorithms which seek to complete a system of PDE to involutive form, and which is based on the Cartan-Kähler theory for exterior differential systems, has been implemented [17, 39]. Despite the variety of methods available, it is the author's belief that anyone developing an algorithm which analyses systems of PDE expressed in the invariants of a group action would benefit by first understanding how a translation of the Kolchin-Ritt algorithm might look, and the pitfalls that can arise.

2.1 Differential Polynomials

We consider systems of partial differential equations that can be regarded as polynomials in some unknown functions $u^1,\dots, u^q$, their derivatives u^α_K where

$$u^\alpha_K = \frac{\partial^{|K|}u^\alpha}{\partial x_1^{K_1}\dots \partial x_n^{K_n}}, \tag{8}$$

and the independent variables $x_1,\dots, x_p$, over $\mathbb{C}$.

Definition 2.1. We denote the set of all such differential polynomials (d.p.'s) by $R_{p,q}$.

In the sequel, we will not hesitate to use complex numbers or complex changes of variables where these are expedient. Further, when manipulating nonlinear systems of differential equations, expressions can factor and then one considers one factor or the other to be zero separately. This requires, in general, that the dependent variables are analytic functions of their arguments. Hence we assume that all expressions are complex differentiable over the complex field. When manipulating polynomials in a computer algebra environment, however, further considerations apply, and we must assume that all coefficients belong in a computable extension of the integers. The issues involved are subtle but far-reaching, and we refer the reader to [16], Chapter 3.

Remark: There are two styles of multi-index notation K for derivatives as in equation (8), in current use. One is an element of $\mathbb{N}^p$, so that $K = (K_1, \dots, K_p)$ while the second is of the form of a string of integers between 1 and p, in which case K_1 is the number of 1's appearing in the string K, and so forth. For both notations, $|K| = K_1 + \ldots + K_p$. In the second notation, the sum of two multi-indices K and L is given by their concatenation KL. This second style of index is useful when considering possibly non-commutative differential operators, which is the case in this article, and mirrors the index notation in use in undergraduate calculus texts, and so we use it here.

2.1.1 Orderings

All the calculations required for differential elimination algorithms depend upon an ordering on the derivative terms.

Definition 2.2. The *traditional total degree ordering* based on $u^1 < u^2 < \ldots < u^q$ and $x_1 < x_2 < \ldots < x_p$ is given by

$$
\begin{array}{ll}
u^\alpha_L > u^\beta_K & \\
\text{if } |L| > |K|, & \\
\text{else} & |L| = |K|,\ u^\alpha > u^\beta, \\
\text{else} & |L| = |K|, \quad \alpha = \beta \text{ and } L_1 > K_1, \\
\text{else} & |L| = |K|, \quad \alpha = \beta, \quad L_1 = K_1, \ldots,\ L_{j-1} = K_{j-1},\ L_j > K_j \\
 & \qquad \text{for some } j \text{ such that } 2 \leq j < n - 1.
\end{array}
$$

There are many orderings, a complete classification is given in [37]. Each is suited to a different purpose; lexicographic orderings yield elimination ideals, while "total degree" orderings are used to obtain both integrability conditions and the "Initial Data" for formal power series solutions to converge [34]. Orderings on the derivative terms need to satisfy two *compatibility* conditions in order for reduction processes to terminate:

$$
\begin{array}{ll}
1. & u^\alpha_J < u^\beta_K \text{ implies that } u^\alpha_{JL} < u^\beta_{KL}, \\
2. & u^\alpha_K < u^\alpha_{KL} \text{ for } |L| \neq 0.
\end{array}
\tag{9}
$$

It is assumed that expressions in the derivative terms are greater than expressions in the independent variables. In all that follows, we assume that the term ordering is specified.

Remark: In modern parlance, the derivative term u^α_K has *order* $|K|$, the term *degree* being reserved for the highest power to which the highest derivative term appearing in an equation is taken. However, most of the definitions and results for over determined systems are based on those for polynomial systems, by direct analogy of u^α_K with the monomial $\mathbf{x}^K$. Hence this use of the term *degree* in describing term orderings.

Definition 2.3. The *highest derivative* term relative to the term ordering occurring in a d.p., f, is denoted $\mathrm{HDT}(f)$. This is denoted by some authors as the "leader" or "leading derivative term", or "principal derivative", and is the analogue of "tip" or "initial" in the (non)-commutative algebra literature.

The *highest power* of the $\mathrm{HDT}(f)$ occurring in f is denoted $Hp(f)$.

The *highest coefficient*, $H\mathrm{coeff}(f)$, is defined to be $\mathrm{coeff}(f,\, \mathrm{HDT}(f)^{Hp(f)})$. Some authors denote this as the "leading coefficient".

The *separant* of f is the highest coefficient of $\partial f/\partial x_i$ for any i (cf. [42]).

2.2 Pseudo-reduction and cross-differentiation

Formulae for pseudo-reduction and cross-differentiation may be found in [25]. Here we give a brief discussion and some simple examples.

A *pseudo-reduction* of a d.p. f by $G = \{g_1,\, g_2, \ldots, g_k\} \subset R_{n,m}$ effects elimination from f of any terms that are derivatives of the $\mathrm{HDT}(g_i)$, for some $i \in \{1, \ldots, k\}$. A simple example will illustrate our meaning. Suppose one wishes to reduce the equation $f \equiv u_{xyy} - u_{xx}u_y = 0$ by the equation $g \equiv u_y u_{xy} - u_x^2 = 0$. We have that the highest derivative term in g is u_{xy}. Then a (one-step) pseudo-reduction of f with respect to g is given by

$$f \to_g u_y f - \frac{\partial g}{\partial y}. \tag{10}$$

Definition 2.4. The *pseudo-normal* form of f with respect to a set G is obtained when no further pseudo-reduction with respect to any member of G is possible, and is denoted $\mathrm{normal}^p(f, G)$.

Theorem 2.5. [22] The compatibility requirements that an ordering on the derivative terms must satisfy (§2.1.1), ensure that reduction to pseudo-normal form is achieved in a finite number of steps when automated on a computer.

Pseudo-reduction leads to the build-up of differential coefficients in simplification calculations. For a set of d.p.'s G, we collect in the set $S(G)$ all the factors by which a d.p. may be multiplied during some pseudo-reduction process, *and are therefore assumed to be non-zero*. This set represents, in some sense, the set of singular integrals, see for example [42].

Definition 2.6. For a finite system of d.p.'s G, define $S(G)$ to be the multiplicative set generated by factors of the highest coefficients and separants of the elements of G, that is, the set of all expressions that can be obtained by multiplying together a finite number of such factors.

Definition 2.7. The *diffSpolynomial* or *cross-derivative* of two d.p.'s is obtained first by differentiating the two equations so that their highest derivative terms become equal, then by cross-multiplying by the highest coefficients, and subtracting. For equations with different highest unknowns, we take their diffSpolynomial to be zero.

Example 2.8. If $f_1 = u_x u_{xx} - u_y^2$ and $f_2 = u_{yy}$, then in a lexicographic ordering based on $x < y$, we have $\mathrm{HDT}(f_1) = u_y$, and $\mathrm{HDT}(f_2) = u_{yy}$, and thus $\mathrm{diffSpolynomial}(f_1, f_2) = \partial f_1/\partial y + 2u_y f_2$. But in an ordering based on $y < x$, then $\mathrm{diffSpolynomial}(f_1, f_2) = \partial^2 f_1/\partial y^2 - u_x \partial^2 f_2/\partial x^2$.

2.3 Characteristic Sets and Differential Gröbner Bases

The purpose of differential elimination algorithms is to find, in some sense, complete sets of integrability conditions that allow one to deduce the existence, or nonexistence, of certain types of equations which are differential consequences of the given system. In this section we first define what is meant by the set of all differential consequences, which is called the ideal of the given system. We then define both a characteristic set and a differential Gröbner basis. It is the reduction property of these sets that allows the information we seek to be readily deduced. For a simple discussion of the kinds of results that can be obtained, see [22].

Definition 2.9. Given a finite set of differential polynomials, $\Sigma \subset R_{p,q}$, we define the *ideal* $I(\Sigma)$ to be all those expressions that can be obtained from the elements of Σ by differentiating and adding, and multiplying by arbitrary elements of $R_{p,q}$, a finite number of times. If the system is linear, we allow multiplication by polynomials in the variables $x_1, \ldots, x_n$ over $\mathbb{C}$ only, that is, we maintain linearity.

Definition 2.10. A *differential Gröbner basis* (DGB) of $I(\Sigma)$ is defined to be a set of generators G of $I(\Sigma)$ such that every element of $I(\Sigma)$ pseudo-reduces to zero with respect to G. A DGB of a system of PDES depends on the ordering of derivative terms. A characteristic set is a subset of the ideal which has the same reduction property but is not necessarily a basis[36].

The algorithm we discuss here, the Kolchin-Ritt algorithm, is a simple translation of Buchberger's algorithm for a Gröbner basis for an algebraic polynomial ideal [4, 10].

Algorithm Kolchin-Ritt

Input: a finite basis $F = \{f_1, f_2, \ldots, f_N\}$ for the differential ideal $I(F)$
a term ordering

Output: G

```
G:=F
pairset:= {{f_i, f_k} | f_i, f_k ∈ G}
while pairset ≠ {}
    for {f_i, f_k} in pairset do
        pairset:=pairset minus {{f_i, f_k}}
        m :=normal^p(diffSpoly(f_i, f_k), G)
        if m ≠ 0 do
            pairset:=pairset union {{f_i, m} | f_i ∈ G}
            G := G union {m}
```

Theorem 2.11. [23] The Kolchin-Ritt algorithm terminates. The set $G = \text{Kolchin-Ritt}(F)$ satisfies $I(G) = I(F)$ and if $S(G) \cap I(G) = \emptyset$ then for all $g \in I(G)$, there exists $s \in S(G)$ such that sg pseudo-reduces to 0 with respect to G.

This algorithm, along with various extensions of it, was implemented in MAPLE [25]. The theorem shows the limitations of the Kolchin-Ritt algorithm in calculating a DGB when one begins with, or obtains en route, an equation where the coefficient or separant of the highest derivative contains differential terms. Indeed, the condition $S(G) \cap I(G) = \emptyset$ encodes the fact that these coefficients are assumed to be non-zero. If an expression is found which implies one of the elements of $S(G)$ is zero, then the algorithm must be re-run with the relevant element of $S(G)$ included at the outset. If an expression is found in which the highest derivative term occurs in a factor raised to a power, then the condition fails. More generally, when seeking all solutions, including singular solutions, of a system, one needs to systematically set each element of $S(G)$ to zero in a branching calculation.

See [3, 19] for a more sophisticated discussion of differential elimination algorithms in terms of radical ideals and ideal decompositions, and improved results.

Our major example in this article is a system containing an unspecified function of the dependent variable. While the theory of DGBS is not developed for such systems, nevertheless one can perform calculations with them in a natural way. Consider the system

$$u_x^2 + u_y^2 - 1 = 0, \quad u_{xx} + u_{yy} - f(u) = 0. \tag{11}$$

Derivative terms in $f(u)$ are considered to be "differential parameters" in the calculations, and conditions obtained only in the derivatives of f are actually *consistency conditions* for the system. If during the course of the calculation we obtain two or more conditions in the derivatives of f only, it is then quite natural to want to calculate a DGB for the system of consistency conditions, with f as the unknown, and u as the independent variable. This does not lead to any inconsistencies; the entire calculation is in two separate pieces, which do not interfere with one another. To date, `diffgrob2` is the only package which can process systems containing arbitrary functions of a dependent variable.

Example 2.12. For the system (11), the output G of the Kolchin-Ritt algorithm is,

$$\begin{aligned} (u_x - 1)(u_x + 1)(f_u + f^2)(2f_u f + f_{uu}) &= 0, \\ (u_x - 1)^2(u_x + 1)^2(f_u + f^2) &= 0, \\ u_{xx} + f(u_x^2 - 1) &= 0, \\ u_x^2 + u_y^2 - 1 &= 0, \\ u_{yy} - f u_x^2 &= 0. \end{aligned} \tag{12}$$

The second equation is in $S(G) \cap I(G)$, so the output fails the condition given in the output statement of the algorithm. In fact, the condition $f_u + f^2 = 0$ is implied by the given set, and is the consistency condition for a solution to exist. Extensions of the Kolchin-Ritt algorithm will, in this case, correct the problem [25]. In §4, we show how dividing out the symmetry, the Euclidean group, gives the equation that f must satisfy literally in one line.

3 The Moving Frame Method

The discussion here is brief, to fix the notation and record the formulae needed in the sequel. We provide a simple expository example which will be needed for §4. We refer the reader to the original articles [11, 12] for details and proofs, and for an exposition of the moving frame method for exterior differential systems, for right group actions, and pseudo-group actions.

3.1 Group actions

We begin with a specified smooth left group action

$$G \times M \to M, \qquad (g, z) \mapsto g \bullet z, \qquad (h, g \bullet z) \mapsto (hg) \bullet z$$

on the manifold M by a Lie group G. Our interest is particularly in the case when M is $X \times U$, where X is the space of independent variables and U the dependent variables. We take the dimension of X to be p and the dimension of U to be q.

Example 3.1. The Euclidean group $E(2) = SO(2) \ltimes \mathbb{R}^2$ acts on $\mathbb{R}^2 \times \mathbb{R}$ as follows,

$$\begin{aligned} (\theta, (a, b)) \bullet (x_1, x_2) = & \begin{pmatrix} \cos(\theta) & -\sin(\theta) \\ \sin(\theta) & \cos(\theta) \end{pmatrix} \begin{pmatrix} x_1 \\ x_2 \end{pmatrix} + \begin{pmatrix} a \\ b \end{pmatrix} \\ (\theta, (a, b)) \bullet u = & \; u \end{aligned} \tag{13}$$

Definition 3.2. The *left regularisation* of a group action is the map

$$G \times (G \times M) \to G \times M, \qquad (g, (h, z)) \mapsto (gh, g \bullet z)$$

and the *left fundamental lifted invariants* are

$$w(h, z) = h^{-1} \bullet z.$$

They are invariants since

$$g \bullet w(h, z) = w(gh, g \bullet z) = h^{-1} g^{-1} g \bullet z = h^{-1} \bullet z = w(h, z).$$

We adopt the notation that the lifted invariants obtained from the independent variables x are labelled y and those from the dependent variables u are labelled v.

Example 3.1(cont) For the Euclidean group action above, the lifted invariants are

$$\begin{aligned} (y_1, y_2) = \quad & (\theta, (a, b))^{-1} \bullet (x_1, x_2) & = \begin{pmatrix} \cos(\theta) & \sin(\theta) \\ -\sin(\theta) & \cos(\theta) \end{pmatrix} \begin{pmatrix} x_1 - a \\ x_2 - b \end{pmatrix} \\ v = \quad & (\theta, (a, b))^{-1} \bullet u & = u \end{aligned} \tag{14}$$

3.2 Prolongation formulae

We assume the group action to be infinitely differentiable so that by using implicit differentiation we obtain an action by G on the jet bundle $J^n(M) \to M$ for all $n > 0$.

Definition 3.3. The *higher order left fundamental lifted invariants* are obtained by implicit differentiation of the lifted invariants defined in Definition 3.1.12.

Example 3.1(cont) For the Euclidean group action above, we obtain

$$\begin{aligned}
v_{y_1} &= \cos\theta\, u_{x_1} + \sin\theta\, u_{x_2} \\
v_{y_2} &= -\sin\theta\, u_{x_1} + \cos\theta\, u_{x_2} \\
v_{y_1y_1} &= \cos^2\theta\, u_{x_1x_1} + 2\sin\theta\cos\theta\, u_{x_1x_2} + \sin^2\theta\, u_{x_2x_2} \\
v_{y_1y_2} &= (\cos^2\theta - \sin^2\theta)u_{x_1x_2} + \sin\theta\cos\theta(u_{x_1x_1} - u_{x_2x_2}) \\
v_{y_2y_2} &= \sin^2\theta\, u_{x_1x_1} - 2\sin\theta\cos\theta\, u_{x_1x_2} + \cos^2\theta\, u_{x_2x_2}
\end{aligned} \tag{15}$$

In fact, the higher order lifted invariants can be obtained recursively.

Definition 3.4. The *lifted invariant differential operators* are

$$\mathcal{E}_j(F) = \frac{\mathrm{D}(y_1, \ldots, y_{j-1}, F, y_{j+1}, \ldots, y_p)}{\mathrm{D}(y_1, \ldots, y_p)} \tag{16}$$

In fact, $\mathcal{E}_j = \partial/\partial y_j$, and we have $v_{Kj} = \mathcal{E}_j v_K$. Note that the $\mathcal{E}_j$ are linear, first order operators which commute and obey the product rule.

With the group action comes an action of its Lie algebra. For a one-parameter subgroup $g(\epsilon) \subset G$ such that $g(0) = e$, the identity element, set $x(\epsilon) = g(\epsilon) \bullet x$, $u(\epsilon) = g(\epsilon) \bullet u$. Then we define ξ_j, $j = 1, \ldots, p$ and ϕ^α_K, $\alpha = 1, \ldots, q$ and K a multi-index, to be such that

$$\begin{aligned}
x_i(\epsilon) &= x_i + \epsilon\xi_i + O(\epsilon^2) \\
u^\alpha(\epsilon) &= u^\alpha + \epsilon\phi^\alpha + O(\epsilon^2) \\
\frac{\partial^{|K|} u^\alpha(\epsilon)}{\partial x(\epsilon)^K} &= u^\alpha_K + \epsilon\phi^\alpha_K + O(\epsilon^2).
\end{aligned} \tag{17}$$

The ξ's and ϕ's are called the *infinitesimals* of the group action and classically depend only on (x, u). Again by iterative use of the chain rule, we may obtain recursion formulae for the ϕ^α_K. These formulae and their derivation can be found in textbooks on symmetries of differential equations cf. [31]. Further, they have been implemented in virtually every computer algebra system as part of Lie's algorithm to find symmetries of differential equations. A review of the software packages available has been given by W. Hereman in [20], (Vol. III, Chapter 13).

Example 3.1(cont) Considering the Euclidean group of our running example, we have for the translations $x_i \mapsto x_i + \epsilon$ that $\xi_j = \delta^i_j$ and $\phi_K = 0$ for all K. For the rotation parametrised by θ we have

$$\begin{gathered}\xi_1 = -x_2, \qquad \xi_2 = x_1, \qquad \phi = 0,\\ \phi_1 = -u_{x_2}, \quad \phi_2 = u_{x_1},\\ \phi_{11} = -2u_{x_1x_2}, \qquad \phi_{12} = u_{x_1x_1} - u_{x_2x_2}, \qquad \phi_{22} = 2u_{x_1x_2}.\end{gathered} \tag{18}$$

3.3 Moving frames

Definition 3.5. A *moving frame* is a G-equivariant map,

$$\rho : M \to G, \qquad \rho(g \bullet z) = g\rho(z).$$

A moving frame will exist if and only if the group action is free and regular. For sufficiently high n, a prolonged group action on $J^n(M)$ will be locally free provided the action on M is locally effective. We refer the reader to [12] for the considerable technical details.

We next show how to construct a local moving frame in the open neighbourhood $\mathcal{U}$. Take a submanifold $\mathcal{K} \subset \mathcal{U}$ which is transverse to the group orbits. We take $\mathcal{U}$ to be small enough so that each orbit intersects $\mathcal{K}$ at most once. For $z \in \mathcal{U}$ take $k \in \mathcal{K}$ and $h \in G$ such that $z = h \bullet k$. The moving frame $\rho : \mathcal{U} \to G$ is then defined by $\rho(z) = h$. The conditions guaranteeing existence and uniqueness of k and h, and thus that ρ is well-defined, are that the group action is regular and free, respectively. The map ρ is equivariant since if $z' = g \bullet z$ and $z = h \bullet k$ then $z' = gh \bullet k$ and thus $\rho(g \bullet z) = \rho(z') = gh = g\rho(z)$.

Definition 3.6. If the cross-section $\mathcal{K}$ is defined by a set of equations, these are known as the *normalisation equations*.

If the normalisation equations take the form $g \bullet z = k$ then $\rho(z) = g^{-1}$. In practice, the frame gives equations for the group parameters in terms of the co-ordinates on the manifold M. "Evaluating on the frame" then means using these specific values of the group parameters in the lifted variables.

Example 3.1(cont) In our running example of the action of E(2) on $J(\mathbb{R}^2 \times \mathbb{R})$, if we take the normalisation equations

$$y_1 = 0, \qquad y_2 = 0, \qquad v_{y_1} = 0,$$

then the frame is given by

$$\rho(x_1, x_2, u, u_{x_1}, u_{x_2}) = (\theta, (a, b)) = \left(\arctan\left(-\frac{u_{x_1}}{u_{x_2}}\right), (x_1, x_2)\right). \tag{19}$$

3.4 Invariants

Evaluating the remaining unnormalised higher order lifted invariants on the frame leads to the *fundamental normalised differential invariants* I_K^α of the group action on M. Any ordinary differential invariant is a function of these.

Definition 3.7. *I_K^α is defined to be v_K^α evaluated on the frame. Further, J^i is defined to be y_i evaluated on the frame.*

Example 3.1(cont) In our running example, substituting (19) into (15) yields the invariants

$$\begin{aligned}
I_0 &= u \\
I_1 &= 0 \\
I_2 &= (u_{x_1}^2 + u_{x_2}^2)^{1/2} \\
I_{11} &= (u_{x_2}^2 u_{x_1x_1} - 2u_{x_1}u_{x_2}u_{x_1x_2} + u_{x_1}^2 u_{x_2x_2})/I_2^2 \\
I_{12} &= \left((u_{x_2}^2 - u_{x_1}^2)u_{x_1x_2} - u_{x_1}u_{x_2}(u_{x_2x_2} - u_{x_1x_1})\right)/I_2^2 \\
I_{22} &= (u_{x_1}^2 u_{x_1x_1} + 2u_{x_1}u_{x_2}u_{x_1x_2} + u_{x_2}^2 u_{x_2x_2})/I_2^2
\end{aligned} \tag{20}$$

Note that $I_{11} + I_{22} = v_{y_1y_1} + v_{y_2y_2} = u_{x_1x_1} + u_{x_2x_2}$ verifying that the Laplacian is invariant. This is an example of the *Fels–Olver–Thomas replacement theorem* [12] (Theorem 10.3), which states

Theorem 3.8. If $F(x_i, u_{K_j}^{\alpha_j})$ is an ordinary differential invariant then

$$F(x_i, u_{K_j}^{\alpha_j}) = F(J^i, I_{K_j}^{\alpha_j}).$$

This is because an ordinary differential invariant takes the same functional form when written in terms of the corresponding fundamental lifted invariants, which is unaffected by evaluation on the frame. While this theorem seems a tautology, in practice it can seem remarkable. The Fels–Olver–Thomas Replacement theorem means that obtaining the invariantised version of a differential system is virtually trivial, at least symbolically. The situation is in stark contrast to that involving discrete symmetries, where considerable effort employing the elimination properties of Gröbner bases must be used, and the Hilbert basis of polynomial invariants must be known explicitly in advance. For a recent paper detailing the ideas involved for systems with discrete symmetries, see [15].

Similarly, evaluating the $\mathcal{E}_j$ operators on the frame leads to a complete set of invariant differential operators.

Definition 3.9. The *fundamental invariant differential operator* $\mathcal{D}_j$ is the operator $\mathcal{E}_j$ evaluated on the frame.

Example 3.1(cont) In the running example this leads to

$$\mathcal{D}_1 = \frac{1}{I_2}\left(u_{x_2}\mathrm{D}_{x_1} - u_{x_1}\mathrm{D}_{x_2}\right), \qquad \mathcal{D}_2 = \frac{1}{I_2}\left(u_{x_1}\mathrm{D}_{x_1} + u_{x_2}\mathrm{D}_{x_2}\right).$$

Note that $\mathcal{D}_1 u = 0 = I_1$, $\mathcal{D}_2 u = I_2$ but $\mathcal{D}_1^2 u = 0 \neq I_{11}$.

3.5 Correction terms for Invariantised Differentiation

Despite the fact that $v_{Kj} = \mathcal{E}_j v_K$, we have in general that $I^\alpha_{Kj} \neq \mathcal{D}_j I^\alpha_K$. By their definition, the invariants I^α_K are left unchanged by permutations within the index K. However, the $\mathcal{D}_j$ do not commute.

Definition 3.10. The *correction terms* M^α_K are defined by

$$\mathcal{D}_j I^\alpha_K = I^\alpha_{Kj} + M^\alpha_{Kj}. \tag{21}$$

We further define M^i_j to be such that $\mathcal{D}_j J^i = \delta^i_j + M^i_j$.

The M^α_{Kj} are not invariant under permutations in the index Kj. In [12] can be found the proofs that there exists an $r \times p$ "correction matrix" K such that

$$\begin{aligned} M^i_j &= \sum_{\kappa=1}^{r} \xi^i_\kappa(I)\mathsf{K}^\kappa_j \\ M^\alpha_{Kj} &= \sum_{\kappa=1}^{r} \phi^\alpha_{K,\kappa}(I)\mathsf{K}^\kappa_j \end{aligned} \tag{22}$$

where, κ is the index for the Lie algebra generators and $\dim(G) = r$. The notation $\xi^i_\kappa(I)$ means the infinitesimal of the κth group generator acting on x_i, lifted and then evaluated on the frame. Similarly, $\phi^\alpha_{K,\kappa}(I)$ means the infinitesimal action of the κth group generator acting on u^α_K lifted and then evaluated on the frame. Since the group action is given *a priori*, these functions are readily calculated using the prolongation formulae.

The matrix K is found by combining these formulae with the normalisation equations. For example, if the first s of the lifted independent variables and a further $r - s$ of the lifted derivatives are normalised to be constants, then from $\mathcal{D}_j(J^i) = 0$, $i = 1, \ldots, s$ and $\mathcal{D}_j(I^{\alpha_\ell}_{K_\ell j}) = 0$, $\ell = 1, \ldots, r - s$ we may solve for the K^κ_j from the pr linear equations

$$\begin{aligned} \sum_{\kappa=1}^{r} \xi^i_\kappa(I)\mathsf{K}^\kappa_j &= -\delta^i_j, \qquad & i = 1, \ldots, s;\ j = 1, \ldots, p \\ \sum_{\kappa=1}^{r} \phi^{\alpha_\ell}_{K_\ell,\kappa}(I)\mathsf{K}^\kappa_j &= -I^{\alpha_\ell}_{K_\ell j} \qquad & \ell = 1, \ldots, r - s;\ j = 1, \ldots, p \end{aligned} \tag{23}$$

The existence and uniqueness of the solution of these equations is due to the transversality of the cross-section defining the frame. Unfortunately, the converse is false; the existence of K does not guarantee that the normalisation equations yield a bona fide moving frame.

Example 3.1(cont) In our running example, the correction matrix is given by,

$$\mathsf{K} = -\begin{pmatrix} 1 & 0 & -I_{11}/I_2 \\ 0 & 1 & -I_{12}/I_2 \end{pmatrix} \tag{24}$$

Definition 3.11. More generally, let the equations giving the moving frame be

$$\psi_\lambda(y_j, v^\alpha_K) = 0, \qquad \lambda = 1, \ldots, r, \ |K| \leq N$$

say. Then $\psi_\lambda(J^j, I^\alpha_K) = 0$ will also hold, by definition of the J^j, I^α_K. Both versions will be denoted as normalisation equations.

The correction matrix for these more general normalisation equations is calculated as follows. Let the variables actually occurring in the ψ_λ be $\mathcal{X} = \{y_{j_i}, v^{\alpha_i}_{K^i}\}$. Define J to be the Jacobian of ψ evaluated on the frame with respect to the variables $\mathcal{X}$. Let $\mathbf{\Phi}$ be the matrix whose entries are the ξ_{j_i}, $\phi^{\alpha_i}_{K^i}$ evaluated on the frame, and let T be the total derivative matrix $(\delta^k_{j_i}, I^{\alpha_i}_{K^ij})$, where $|K| \leq N$ and δ^k_j is the Kronecker delta, evaluated on the frame. Then [32]

$$\mathsf{K} = -\mathsf{T}\mathsf{J}^T(\mathbf{\Phi}\mathsf{J}^T)^{-1} \tag{25}$$

We give an example in §5. We then have that

$$\mathcal{D}_j(\psi_\lambda(J^j, I^\alpha_K)) = 0, \qquad \text{for all } j,\ \lambda. \tag{26}$$

From the recursion formulae for the induced action of the Lie Algebra, the formulae for the correction terms M^α_{Kj} and the correction matrix K above, we have the following theorem.

Theorem 3.12. If the normalisation equations are of polynomial type, and the induced action of the Lie Algebra is of rational type, then the correction terms are rational. Moreover, the multiplicative set generated by factors of the denominators is finitely generated.

3.6 Generating invariants and syzygies

The set $\mathcal{L} = \{u^\alpha_K\}$ has a well-known differential structure. It is in one-to-one correspondence with a finite number of copies (one for each α) of $\mathbb{N}^p$, where the p-tuple of integers corresponds to the multi-index of differentiation. Each component of $\mathcal{L}$ is generated by u^α using the differential operators $\partial/\partial x_j$, $j = 1, \ldots, p$. Further, these differential operators commute. By contrast, the invariant differential operators do not commute, in general, and $\mathcal{D}_j I^\alpha_K$ need not even be of polynomial type. In fact for the examples studied in this article, the $\mathcal{D}_j I^\alpha_K$ are rational expressions in the invariantised derivatives.

Secondly, the normalisation equations define identities on the invariants.

Definition 3.13. If the normalisation equations are of the simplest form, where certain of the y_j, v^α and v^α_K are set to be constant, then we say that the corresponding J^j, I^α and I^α_K are the *highest normalised invariants*. For more general normalisation equations, we say an invariant is a *highest normalised invariant* if it is the the "highest derivative term" of some normalisation equation according to some (specified) ordering on the derivatives and independents. The set of highest normalised invariants will be denoted by $\mathcal{HNI}$.

If $\psi(J^i, I^\alpha_K) = 0$ is a normalisation equation, then $\mathcal{D}_j \psi(J^i, I^\alpha_K) = 0$ for all j i.e. the equations (26) hold. Hence, if $I^\alpha_K = c$ is a normalisation equation, then $\mathcal{D}_j I^\alpha_K = 0$ and so it is not possible to obtain I^α_{Kj} by differentiating I^α_K. Note that the invariants I^α_{Kj}, $j = 1, \ldots, p$ are not necessarily zero; this follows from the definition of the I^α_K. Further, it is not possible to obtain I^α_K by differentiating I^α_J where $Jk = K$; the term simply does not appear as it is cancelled by the relevant correction term.

In a computer algebra environment, where invariantised derivatives are manipulated symbolically, "evaluating on the frame" is achieved by simplifying with respect to the normalisation equations. Thus if the normalisation equations are of polynomial type, an algebraic Gröbner basis for the normalisation system is calculated and then simplification is effected by calculating the unique normal form. One can assume without loss of generality that the normalisation equations are already an algebraic Gröbner basis for the ideal they generate. Not all Gröbner bases are appropriate as normalisation equations, however. The solution surface must be a single component of the correct dimension. Further, normalisation equations such as $(I_K)^2 = 0$, for example, can lead to serious problems such as undetected zero leading coefficients. Thus the ideal formed by the normalisation equations must at least be prime. The precise conditions needed on the normalisation equations for them to be suitable for computation in a symbolic environment is a topic for further study. In the meantime, we make the following standing assumption.

Standing assumption We will assume that the normalisation equations are linear in the highest normalised invariants that is, their highest derivative terms.

Under this assumption, one can think of the set of invariantised derivative terms as a "lattice with holes"; the normalised invariants never appear in any calculation.

Because of these "holes", we need to include additional generating differential invariants, given in the next theorem, to obtain all invariants in terms of the generators and their derivatives. A classical theorem due to Tresse [40] states that all differential invariants can be obtained as functions of a finite number of invariants and their invariant derivatives. The following theorem gives precisely what they are.

Theorem 3.14. ([12], Theorem 13.4) The set given by

$$\{ J^i, I^\alpha, I^\alpha_{Kj} \mid J^i, I^\alpha \notin \mathcal{HNI}, I^\alpha_K \in \mathcal{HNI}, j = 1, \dots, p \} \tag{27}$$

is the *fundamental generating set* of differential invariants.

Another major difference between the sets $\{u^\alpha_K\}$ and $\{I^\alpha_K\}$ is the existence of nontrivial *syzygies*.

Definition 3.15. If I^α_J, I^α_L are two generating differential invariants, and indices K, M are such that $I^\alpha_{JK} = I^\alpha_{LM}$, then

$$\mathcal{D}_K I^\alpha_J - \mathcal{D}_M I^\alpha_L = M^\alpha_{JK} - M^\alpha_{LM} \tag{28}$$

is called a *fundamental differential syzygy of the third kind*. Syzygies of the first and second kind are actually the equations (26).

Example 3.1(cont) For our running example, the generating set of differential invariants is

$$\{I_0, I_{12}, I_{11}\}$$

and the syzygy between I_{12} and I_{11} is

$$\mathcal{D}_1 I_{12} - \mathcal{D}_2 I_{11} = (I_{11}^2 - I_{11} I_{22} + 2I_{12}^2)/I_2.$$

3.7 The "differential structure" of the set of invariants

Summarizing from the previous section, the "differential structure" of the set $\{J^i, I^\alpha_K \mid i, \alpha, K\}$ differs from that of $\{x_i, u^\alpha_K \mid i, \alpha, K\}$ in the following ways.

1. There are functional relations between the invariants, given by the normalisation equations.

2. The invariant differentiation operators may not commute.

3. $\mathcal{D}_j I^\alpha_K$ may not be of polynomial type. If the induced action of the Lie algebra is rational and the normalisation equations polynomial, then the correction terms will be rational. Further, the denominators belong to a finitely generated multiplicative set.

4. More than one generator may be required to obtain the full set of invariants under invariant differentiation.

5. There may be nontrivial differential syzygies.

Differential elimination algorithms are analysed by means of an *algebraic model* of a differential system. For non-invariantised differential systems one considers an ideal of some differential ring to be the model, and theorems are known that enable the output of the algorithm at hand to be stated and proved precisely. By contrast, the construction of a useful algebraic model for invariantised systems appears to be an open problem.

4 Invariantised Differential Systems – a simple example

Consider the system

$$\begin{aligned} u_{xx} + u_{yy} &= f(u), \\ u_x^2 + u_y^2 &= 1, \end{aligned} \tag{29}$$

Using the Fels–Olver–Thomas Replacement theorem and the moving frame calculated in the previous section, the system can be written as

$$\begin{aligned} I_{11} + I_{22} &= f(I_0), \\ I_2 &= 1, \end{aligned} \tag{30}$$

where we are assuming the positive value of the square root. The calculation for the other choice is entirely analogous. Using (21), (22) and the correction matrix (24), we have from $I_2 = 1$ that

$$I_{12} = 0, \qquad I_{22} = 0. \tag{31}$$

We can now simplify the correction matrix (24) to be

$$\mathsf{K} = -\begin{pmatrix} 1 & 0 & -I_{11} \\ 0 & 1 & 0 \end{pmatrix} \tag{32}$$

We next use $I_{22} = 0$ in order to simplify the system (30) to

$$I_{11} - f(I_0) = 0, \qquad I_2 - 1 = 0. \tag{33}$$

If we cross differentiate these two equations, and simplify using (31,33), we obtain

$$\mathcal{D}_2(I_{11} - f(I_0)) - \mathcal{D}_1^2(I_2 - 1) = -f'(I_0) - f(I_0)^2 = 0$$

or

$$f'(u) + f(u)^2 = 0.$$

Hence

$$f(u) \equiv 0 \quad \text{or} \quad f(u) = \frac{1}{u+c}$$

where c is a constant. The equation that f must satisfy for a solution to exist is found to be precisely the syzygy between the two generating invariants I_{11} and I_{12} evaluated on the system. Since there are no more integrability conditions that can be calculated by cross differentiation and simplification, we have the output of an algorithm analogous to the Kolchin Ritt algorithm to be

$$\begin{aligned} I_{11} - f(I_0) &= 0 \\ I_2 - 1 &= 0 \\ f'(I_0) + f(I_0)^2 &= 0. \end{aligned}$$

It is clear that every higher order differential invariant can be evaluated on solutions of the system by applying the invariant differential operators to an element of the output system. What we have not demonstrated is that different paths of differentiation in the calculation of these higher order invariants will not lead to further compatibility conditions.

Comparing the output of the algorithm on the invariantised system to that of the original system, it appears that the ordinary differential equation for u, the third equation of the system (12) is lost. However, from the equations $I_{11} = f(u)$, $I_{12} = I_{22} = 0$, we can obtain by linear algebra expressions for the $u_{x_ix_j}$ using (20). This yields

$$\begin{aligned} u_{x_1x_1} &= f(u)(1 - u_{x_1}^2) \\ u_{x_1x_2} &= -f(u)u_{x_1}u_{x_2} \\ u_{x_2x_2} &= f(u)(1 - u_{x_2}^2) \end{aligned}$$

From here the general solution to the system is easily found, showing *a fortiori* that no further compatibility conditions exist. This final section of the calculation is the "inverse" part of the problem, where we use our knowledge of the values of the invariants on solutions of the system to calculate the values of the usual derivatives on the system.

There are several points that even this simple example highlights:

1. denominators in the correction matrix K could be be zero on solutions of the system.

2. the invariantised system could be incompatible with the normalisation equations.

Thus the normalisation equations must be chosen with care. The theories that analyze the correctness and termination of the Kolchin-Ritt algorithm assume that all differential expressions are of polynomial type. Denominators in the correction matrix K mean that all cross-differentiation and simplification formulae need to be modified to make the result of all calculations to be of polynomial type.

5 The $2+1$ d'Alembert-Hamilton System

In this section we show the calculations for a more significant example. Our first aim is to find all possible f for which the system [7, 14]

$$\begin{aligned} u_{xx} + u_{yy} + \epsilon u_{zz} &= f(u), \\ u_x^2 + u_y^2 + \epsilon u_z^2 &= 1, \end{aligned} \tag{34}$$

with $\epsilon^2 = 1$, is consistent. The sign of ϵ is not significant for the symbolic calculations performed here, and the transformation $z \mapsto iz$ takes one to the other. For convenience, we take $\epsilon = 1$. The example demonstrates several problems that need to be taken into account when designing an invariantised Kolchin Ritt algorithm.

5.1 Calculations using the invariantised system

We consider the six-dimensional Euclidean group E(3) of translations and rotations acting on $\mathbb{R}^3$ in the usual way, with the dependent variable u as an invariant. The system (34) is invariant under this group action. First the complete set of generating differential invariants and the invariant differential operators are calculated. Then integrability conditions for the invariantised system are calculated. The method used to do this in a systematic way which may be implemented on a computer are developed into algorithms in §6. The ordinary differential equation that $f(u)$ must satisfy for (7) to be consistent, as well as equations for the invariants on solutions of the system are found. From these the equations for the second, third and higher order derivatives of u on solutions can be determined using only linear algebra. The entire calculation can be performed using a commercial computer package such as MAPLE with only a modicum of internal memory. By contrast, the analysis of the system (34) using techniques from differential algebra (in the usual derivative terms) was intractable, due to intermediate expression swell.

We build a representation of SO(3), the rotation part of the Euclidean group on $\mathbb{R}^3$, near the identity element by taking the product of rotations. Set

$$\begin{aligned}
\mathbf{B}_1 &= \begin{pmatrix} \cos(\theta) & -\sin(\theta) & 0 \\ \sin(\theta) & \cos(\theta) & 0 \\ 0 & 0 & 1 \end{pmatrix} \\
\mathbf{B}_2 &= \begin{pmatrix} \cos(\psi) & 0 & -\sin(\psi) \\ 0 & 1 & 0 \\ \sin(\psi) & 0 & \cos(\psi) \end{pmatrix} \\
\mathbf{B}_3 &= \begin{pmatrix} 1 & 0 & 0 \\ 0 & \cos(\chi) & -\sin(\chi) \\ 0 & \sin(\chi) & \cos(\chi) \end{pmatrix}
\end{aligned}$$

and set

$$\mathbf{A} = \mathbf{B}_1\mathbf{B}_2\mathbf{B}_3.$$

The translation part of the group leaves invariant the derivatives of u. Thus, we must normalize the translation part of the group using the independent variables. Only the rotation part of the group needs to be lifted to the derivatives and the normalisation of the rotation parameters must involve the derivatives of u.

Remark on notation It is convenient at times to refer to x, y and z as x_1, x_2 and x_3 in summation formulae, and to denote derivatives and other subscripted variables only by the numerical part of the index. Thus, for example we refer to the lifted derivative operator corresponding to $\partial/\partial x$ by $\mathcal{E}_1$ while $v_{y_i y_j}$ is denoted v_{ij}, and so forth.

Using the fact that $\det(\mathbf{A}) = 1$ and $\mathbf{A}^T = \mathbf{A}^{-1}$, the lifted derivative operators $\mathcal{E}_i$ are given by the ith component of $\mathbf{A}^T(\nabla)$, or

$$\mathcal{E}_i(F) = \sum_{k=1}^{3} \mathbf{A}^T_{ik}(\nabla F)_k$$

so that, for example,

$$\begin{aligned}
v_2 &= (-\sin(\theta)\cos(\chi) - \cos(\theta)\sin(\psi)\sin(\chi))u_x \\
&\quad (\cos(\theta)\cos(\chi) - \sin(\theta)\sin(\psi)\sin(\chi))u_y + \cos(\psi)\sin(\chi)u_z \\
v_3 &= (\sin(\theta)\sin(\chi) - \cos(\theta)\sin(\psi)\cos(\chi))u_x \\
&\quad (-\cos(\theta)\sin(\chi) - \sin(\theta)\sin(\psi)\cos(\chi))u_y + \cos(\psi)\cos(\chi)u_z
\end{aligned}$$

The first five normalisation equations are

$$y_i = 0, \ i = 1, 2, 3, \qquad v_2 = 0, \qquad v_3 = 0$$

The translation part of the group is normalised to be translation to (x, y, z) and the first two rotation parameters are given by

$$\theta = \arctan\left(\frac{u_y}{u_x}\right), \qquad \psi = \arctan\left(\frac{u_z}{\sqrt{u_x^2 + u_y^2}}\right) \tag{35}$$

Substituting for θ and ψ into v_1 yields the differential invariant of order one,

$$I_1 = \sqrt{u_x^2 + u_y^2 + u_z^2}$$

To obtain an equation for the rotation parameter χ we need to consider the action of E(3) on the second order derivative terms. Recall this is achieved by operating on u recursively using the $\mathcal{E}_i$. Since our action is linear, we have

$$v_{ij} = \sum_{k=1}^{3}\sum_{\ell=1}^{3} \mathbf{A}^T_{ik}\mathbf{A}^T_{j\ell}u_{k\ell}$$

Back substituting for θ and ψ into the v_{ij} yields one invariant, namely

$$I_{11} = \left(u_x^2u_{xx} + 2u_xu_yu_{xy} + u_y^2u_{yy} + 2u_yu_zu_{yz} + u_z^2u_{zz} + 2u_xu_zu_{xz}\right)/I_1^2. \tag{36}$$

The selection of the sixth normalisation equation is crucial to the remainder of the calculation. Taking for example v_{23} to be zero leads to incompatibilities when considering the invariantised version of the system (34); on solutions of the system the denominators of some entries in the associated correction matrix $\mathbf{K}$ are zero. The normalisation equation chosen was

$$2v_{22} - v_{33} = 0.$$

There will be many compatible choices. Solving for the angle χ at this point of the calculation is unproductive. The expression involves nested radicals and is too unwieldy to give any significant insight. We will need the expression for χ only in the later stages of our exposition. It is helpful to simplify the equation for χ using information gained from processing the invariantised system symbolically, and this we now proceed to do.

In some respects the first object to be calculated should be the correction matrix K. This can be done completely symbolically for various normalisation equations giving a first insight into which normalisation equations might be compatible and useful. It must always be checked, however, that the normalisation equations define a bona fide moving frame. We use the formula (25). In constructing J, $\boldsymbol{\Phi}$ and T below we need consider only those derivative terms actually appearing in the normalisation equations, namely the y_i and v_2, v_3, v_{22} and v_{33}. The κth row of the matrix $\boldsymbol{\Phi}$ consists of the group infinitesimals ξ_1, ξ_2, ξ_3, ϕ_2, ϕ_3, ϕ_{22} and ϕ_{33} for the κth group parameter, lifted and then evaluated on the frame. In lifting and evaluating an expression on the frame, all occurrences of x, y and z are replaced by zero, as are u_y and u_z. Further, all occurrences of I_{33} are replaced by $2I_{22}$. Hence in this example,

$$\boldsymbol{\Phi} = \begin{pmatrix} 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & I_1 & 0 & 2I_{12} & 0 \\ 0 & 0 & 0 & 0 & I_1 & 0 & 2I_{13} \\ 0 & 0 & 0 & 0 & 0 & -2I_{23} & 2I_{23} \end{pmatrix}$$

The matrix J is the Jacobian of the normalisation equations evaluated on the frame;

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

The third ingredient T needed for the correction matrix K is the total derivative matrix evaluated on the frame. Thus T is given by

$$\mathsf{T} = \begin{pmatrix} 1 & 0 & 0 & I_{12} & I_{13} & I_{122} & I_{133} \\ 0 & 1 & 0 & I_{22} & I_{23} & I_{222} & I_{233} \\ 0 & 0 & 1 & I_{23} & 2I_{22} & I_{223} & I_{333} \end{pmatrix}.$$

Finally the 3×6 correction matrix is

$$\mathbf{K} = -\begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \\ \dfrac{I_{12}}{I_1} & \dfrac{I_{22}}{I_1} & \dfrac{I_{23}}{I_1} \\ \dfrac{I_{13}}{I_1} & \dfrac{I_{23}}{I_1} & 2\dfrac{I_{22}}{I_1} \\ \dfrac{2I_{12}^2}{3I_1 I_{23}} - \dfrac{I_{13}^2}{3I_1 I_{23}} & \dfrac{2I_{12}I_{22}}{3I_1 I_{23}} - \dfrac{I_{13}}{3I_1 I_{23}} & \dfrac{I_{12}}{3I_1} - \dfrac{2I_{22}I_{13}}{I_1 I_{23}} \\ -\dfrac{2I_{122} - I_{133}}{6I_{23}} & -\dfrac{2I_{222} + I_{233}}{6I_{23}} & -\dfrac{2I_{223} - I_{333}}{6I_{23}} \end{pmatrix}^T \tag{37}$$

It can be seen that this correction matrix contains third order invariants, while our system is second order. Performing cross differentiation and reduction calculations with this correction matrix on the invariantised system, in a naive translation of the Kolchin-Ritt algorithm to invariantised derivatives using invariantised differentiation, will result in infinite loops when carried out automatically by computer. The reason for this is the lack of compatibility of any term ordering with invariantised differentiation (cf. §2.1.1).

The key new idea of the invariantised Kolchin-Ritt algorithm is to take all the equations defining the system and differentiate them to the order of the derivative terms appearing in the correction matrix. This will be one order higher than the derivative terms appearing in the normalisation equations. An algebraic Gröbner basis or characteristic set is then calculated, using a term ordering which depends on the total degree of differentiation. If an equation of order lower than those used to obtain it algebraically is discovered, this is differentiated and the new equations added and a new Gröbner basis calculated. The matrix entries of $\mathbf{K}$ are then reduced using the final Gröbner basis. Only then does one calculate cross-derivatives of equations of order greater than those appearing in the normalisation equations. A precise statement of the algorithm appears in the next section.

Using the Replacement Theorem, the system (34), lifted and evaluated on the frame is

$$I_{11} + 3I_{22} = f(I_0), \qquad I_1 = 1 \tag{38}$$

where we have taken the positive square root. The case $I_1 = -1$ is entirely analogous. Using (21), (22) and with $\mathbf{K}$ as in (37) we have from $\mathcal{D}_j(I_1 - 1) = 0$, $j = 1, 2, 3$ that

$$I_{11} = I_{12} = I_{13} = 0.$$

Differentiating these and also $3I_{22} - f(I_0) = 0$ three different ways and calculating an algebraic Gröbner basis on the set of all equations obtained thus far yields the condition

$$18I_{23}^2 + 9f' + 5f^2 = 0 \tag{39}$$

This last condition is differentiated three different ways to obtain three new third order equations. The final Gröbner basis obtained using the term ordering $f < f' < f'' < I_1 < I_{22} < I_{23} < I_{12} < I_{13} < I_{11} < I_{222} < I_{223} < I_{233} < I_{333} < I_{122} < I_{123} < I_{112} < I_{113} < I_{111}$ is

$$\begin{gathered}
f'' + 3ff' + f^3 = 0, \\
I_1 - 1 = 0, \qquad 3I_{22} - f = 0, \\
18I_{23}^2 + 9f' + 5f^2 = 0, \\
I_{12} = 0, \qquad I_{13} = 0, \qquad I_{11} = 0 \\
I_{222}(2f' + f^2) = 0, \\
fI_{223} + 6I_{23}I_{222} = 0, \qquad f'I_{223} - 3fI_{23}I_{222} = 0, \qquad 6I_{23}I_{223} - fI_{222} = 0, \\
I_{233} + I_{222} = 0, \quad I_{333} + I_{223} = 0, \\
6I_{122} - 3f' - f^2 = 0, \quad I_{123} + fI_{23} = 0, \quad 6I_{133} - 3f' + f^2 = 0 \\
I_{112} = 0, \qquad I_{113} = 0, \qquad I_{111} = 0.
\end{gathered} \tag{40}$$

The ordinary differential equation for f has for its solution,

$$f = 0, \qquad \frac{1}{u+c}, \quad \text{or} \quad \frac{1}{u+c_1} + \frac{1}{u+c_2}.$$

Taking factors and recalculating Gröbner bases leads to several cases. Those with $f = 0$ and thus $I_{23} = 0$ are unacceptable, as then K will have zero denominators. The case $f = 0$ needs to be calculated separately.

In the case $2f' + f^2 = 0$, there are no further compatibility conditions, and the three conditions for I_{223}, I_{222} are equivalent.

In the case $I_{222} = 0$, $f \neq 0$, there is one further compatibility condition. Calculating the invariantised diffSpolynomial between $I_{333} = 0$ and $I_{233} = 0$, to obtain two equations for I_{2333} yields, after simplification,

$$f' + f^2 = 0.$$

It is worthwhile noting that this particular diffSpolynomial is none other than the fundamental syzygy between the two fundamental invariants I_{333} and I_{233} (cf. §3.5) evaluated on the system.

5.2 The "inverse problem"

The final step is to calculate the value of the usual derivative terms from knowledge of the higher order invariants. This is where we must "bite the bullet" of the nonlinear equations for the group parameters. The strategy is first to obtain the expression for the group parameter χ *evaluated on the system*, which is more manageable.

The invariants I_{ij} are $v_{ij} = \mathcal{E}_i\mathcal{E}_j(u)$ evaluated on the frame. The angles θ and ψ are given by (35). The equation for the rotation angle χ is $2v_{22} - v_{33} = 0$. Further, we have $I_{11} = 0$, $I_{12} = I_{13} = 0$, $I_{23} = G(u)$ and $3I_{22} = f(u)$, where $G(u)^2 = -(9f' + 5f^2)/18$. Thus we have six equations in the six u_{ij} and the angle χ. It can be observed by direct calculation that I_{11} (36) is independent of χ, while $I_{12} = H_1 \sin\chi + H_2\cos\chi$ and $I_{13} = H_2\sin\chi + H_1\cos\chi$, where the H_i are expressions in the u_{ij}, u_i only. The remaining expressions are linear in $\sin(2\chi)$ and $\cos(2\chi)$. The strategy employed was to solve $I_{11} = 0$, $H_1 = H_2 = 0$, $I_{23} = G(u)$ and $3I_{22} = f(u)$ for u_{11}, u_{12}, u_{13}, u_{23} and u_{33} in terms of u_{22}, u_1, u_2, u_3 and $\sin(2\chi)$, $\cos(2\chi)$. The result was back substituted into $2v_{22} - v_{33} = 0$. Simplifying the result using $\sin^2(2\chi) + \cos^2(2\chi) = 1$ and $u_1^2 + u_2^2 + u_3^2 = 1$ yielded an equation of the form

$$A\sin(2\chi) + B\cos(2\chi) + C = 0$$

where

$$\begin{aligned}
A &= 6G(u)(u_x^2(u_y^2+1) + u_y^2(u_y^2-1)) - 2u_xu_yu_zf(u) \\
B &= f(u)(u_x^2(u_y^2+1) + u_y^2(u_y^2-1)) + 12u_xu_yu_zG(u) \\
C &= 3(u_x^2+u_y^2)(2u_{yy} + f(u)(u_y^2-1))
\end{aligned}$$

Then

$$\sin(2\chi) = \frac{-AC + B\sqrt{A^2+B^2-C^2}}{A^2+B^2}, \qquad \cos(2\chi) = \frac{BC - A\sqrt{A^2+B^2-C^2}}{A^2+B^2} \tag{41}$$

is a suitable solution for χ. The denominator factors to be $(u_y^2-1)^2(u_x^2+u_y^2)(36G^2+f^2)$ which is zero if f satisfies $2f' + f^2 = 0$, in which case $f = 2/(u+c)$.

If $2f' + f^2 = 0$, then $B = \mathrm{i}A$ and we have

$$\sin(2\chi) = -\frac{A^2+C^2}{2AC}, \qquad \cos(2\chi) = -\mathrm{i}\frac{A^2-C^2}{2AC} \tag{42}$$

We can obtain the third order derivatives of u using

$$u_{ijp} = \sum_{\ell,k,s=0}^{3} \mathbf{A}_{i\ell}(I)\mathbf{A}_{jk}(I)\mathbf{A}_{ps}(I)I_{\ell ks},$$

where all the entries of $\mathbf{A}$ are evaluated on the frame. This uses the fact that $\mathbf{A}^T = \mathbf{A}^{-1}$.

5.2.1 $f' + f^2 = 0,\ f \neq 0$

We have $G(u)^2 = 2f^2/9$. Then I_{111}, I_{112}, I_{113}, I_{222}, I_{223}, I_{233}, I_{333} are all zero, while $I_{123} = -G(u)f(u)$, $I_{122} = -f(u)^2/3$, $I_{133} = -2f(u)^2/3$. Further, we may take $f(u) = 1/u$.

Obtaining the expression for $u_{222} \equiv u_{yyy}$ yields

$$u_{yyy} = -3\frac{u_y u_{yy}}{u}. \tag{43}$$

This is an ordinary differential equation for u as a function of y. Similarly, u will satisfy the same ordinary differential equation in terms of x or z obtained by substituting x or z for y in (43), as can be proved directly by performing a similar calculation as above but obtaining expressions for all second order derivatives of u in terms of u_{xx} or u_{zz} respectively.

Thus we obtain

$$u_{yy} = \frac{Y_1(x,z)}{u^3}, \qquad u_y^2 = -\frac{Y_1(x,z)}{u^2} + Y_2(x,z) \tag{44}$$

and similarly for u_{xx}, u_{zz}. From the system (7) itself, we then have

$$X_1(y,z) + Y_1(x,z) + Z_1(x,y) = u^2 \tag{45}$$

$$X_2(y,z) + Y_2(x,z) + Z_2(x,y) = 2 \tag{46}$$

Using these last three equations and the fact $X_{i,x} = Y_{i,y} = Z_{i,z} = 0$, it is relatively straightforward to solve for u using repeated differentiations and decoupling. The result is

$$u^2 = c_{11}x^2 + c_{22}y^2 + c_{33}z^2 + c_{12}xy + c_{13}xz + c_{23}yz + c_1 x + c_2 y + c_3 z + c_0$$

where

$$c_{11} + c_{22} + c_{33} = 2$$

and c_0 is free, and the remaining coefficients may be determined in terms of the c_{ii} by direct substitution into the system (7) and then equating coefficients.

5.2.2 $2f' + f^2 = 0,\ f \neq 0$

In this case, we no longer have 10 equations for the 10 third order invariants, merely 9, indicating a family of solutions depending on an arbitrary function of a single argument. It is not difficult to display such a solution family. Without loss of generality we may take $f(u) = 2/u$ and then the system (7) is invariant under the scaling symmetry,

$$\mathbf{v} = x\partial_x + y\partial_y + z\partial_z + u\partial_u.$$

The Lie symmetry reduction of the system is then

$$u(x,y,z) = zw(t_1,t_2), \qquad \text{where } t_1 = x/z,\ t_2 = y/z$$

and

$$\begin{aligned} w_1^2 + w_2^2 + (w - t_1 w_1 - t_2 w_2)^2 &= 1, && (47)\\ (1+t_1^2)w_{11} + 2t_1 t_2 w_{12} + (1+t_2^2)w_{22} &= 2/w && (48) \end{aligned}$$

In fact, (48) is a differential consequence of (47). This can be seen easily if we apply the Legendre transform. Following [9], (Vol. II, page 32ff), we set

$$\begin{gathered} t_1 = v_\phi, \qquad t_2 = v_\eta \\ w_1 = \phi, \qquad w_2 = \eta, \\ v + w = t_1\phi + t_2\eta \\ w_{11} = \rho v_{\eta\eta}, \quad w_{12} = -\rho v_{\phi\eta}, \quad w_{22} = \rho v_{\phi\phi} \\ \rho = \frac{1}{v_{\phi\phi}v_{\eta\eta} - v_{\phi\eta}^2} \end{gathered}$$

then equations (47,48) become

$$\begin{aligned} \phi^2 + \eta^2 + v^2 &= 1 && (49)\\ (1+v_\phi^2)v_{\eta\eta} - 2v_\phi v_\eta v_{\phi\eta} + (1+v_\eta^2)v_{\phi\phi} &= 2(v_{\phi\phi}v_{\eta\eta} - v_{\phi\eta}^2)/(\phi v_\phi + \eta v_\eta - v) && (50) \end{aligned}$$

It is simple to show that (49) is a solution of (50) and thus it suffices to solve (47) by the method of characteristics, [9] (Vol. II, Ch. II, §3), leading to a family of solutions with one arbitrary function of one argument.

5.3 Discussion

The solutions demonstrated in the previous section show *a fortiori* that no further compatibility conditions exist. Further, they show that all the information necessary to obtain the solution is in the invariantised system. It is not necessary to do more calculations other than those needed in the inverse problem, that of writing the invariantised system in terms of the original variables. For the examples illustrated here, the output of the invariantised Kolchin-Ritt algorithm to be elaborated in the next section, is "complete" in some sense.

6 An invariantised Kolchin-Ritt procedure

In this section we describe an algorithm which calculates an invariant integrability condition arising from each "critical pair path", a concept we define in §6.2, for a large class of input systems, normalisation equations and group actions.

Recall our standing assumptions are that the normalisation equations are of polynomial type, and are linear in the highest normalised invariants. Further, the induced action of the Lie Algebra must be of rational type. These imply that the $\mathcal{D}_j I^\alpha_K$ are rational. Moreover, the denominators are products of a finite number of expressions which can be determined at the outset. These must be non-zero on the system being studied.

First we discuss the compatibility of term orderings with invariant differentiation. We then discuss the analogues of cross differentiation and reduction calculations. We then detail an analogue of the Kolchin-Ritt algorithm which avoids all known difficulties.

6.1 Invariantised differential polynomials

The Fels–Olver–Thomas Replacement Theorem in §3.4 gives an easy translation from differential polynomials to their expression in terms of the invariants. Thus we may speak of invariant(ised) differential polynomials, and we may order the invariant derivatives in a completely analogous way.

Definition 6.1. Given $u^1 < u^2 < \ldots < u^q$ and $x_1 < x_2 < \ldots < x_p$, the traditional total degree ordering **ttdeg** on the differential invariants is given by

$$\begin{aligned}
&\mathrm{I}^\alpha_K >_{\mathbf{ttdeg}} \mathrm{I}^\beta_L\\
&\text{if } |K| > |L|,\\
&\text{else } |K| > |L|,\ u^\alpha > u^\beta,\\
&\text{else } |K| > |L|,\ \alpha = \beta \text{ and } K_1 > L_1,\\
&\text{else } |K| > |L|,\ \alpha = \beta,\ K_1 = L_1,\ \ldots,\ K_{j-1} = L_{j-1},\ K_j > L_j\\
&\qquad\qquad \text{for some } j \text{ such that } 2 \le j \le p-1.
\end{aligned}$$

By interchanging the first two criteria in the above we obtain a mixed elimination, total degree ordering. More generally, we may take a partition on the dependent variables $\mathcal{P}_j$,

$j=1,\ldots,r$ and define an ordering **mtdeg** by

$$
\begin{aligned}
&\mathrm{I}^{\alpha}_{K} >_{\mathbf{mtdeg}} \mathrm{I}^{\beta}_{L}\\
&\text{if } u^{\alpha} \in \mathcal{P}_i,\ u^{\beta} \in \mathcal{P}_j,\ \text{and } i > j\\
&\text{else } i = j \text{ and } |K| > |L|,\\
&\text{else } i = j,\ |K| = |L|,\ u^{\alpha} > u^{\beta},\\
&\text{else } |K| = |L|,\ \alpha = \beta \text{ and } K_1 > L_1,\\
&\text{else } |K| = |L|,\ \alpha = \beta,\ K_1 = L_1,\ \ldots,\ K_{j-1} = L_{j-1},\ K_j > L_j\\
&\qquad\qquad \text{for some } j \text{ such that } 2 \le j \le p.
\end{aligned}
$$

In what follows, we will assume that the term ordering τ is one of the above. Such term orderings are said to be *total degree term orderings*.

Given an ordering which is compatible with differentiation, as described in the conditions (1) and (2) in §2.1.1, it is *not* the case that its invariant analogue will be compatible with invariant differentiation. The problem can be seen by examining the formula (21). The terms M_{Kj} contain by construction invariants of order $N+1$ where N is the order of the normalisation equations, as well as the invariantised infinitesimals of the group action of order $|K|$. Thus, if $|J|$, $|K| < N$, and $I^{\alpha}_{K} < I^{\alpha}_{J}$, it may not be true that $\mathcal{D}_j I^{\alpha}_{K} < \mathcal{D}_j I^{\alpha}_{J}$. However, we have the following result.

Theorem 6.2. If $|J|$, $|K| > N$, then we do obtain compatibility,

$$I^{\alpha}_{K} < I^{\alpha}_{J} \Longrightarrow \mathcal{D}_j I^{\alpha}_{K} < \mathcal{D}_j I^{\alpha}_{J}$$

provided the term ordering is a total degree term ordering.

Definition 6.3. We denote the set of all invariant(ised) differential polynomials (i.d.p.'s) obtained by invariantisation of the differential ring $R_{p,q}$ to be $\mathcal{I}_{p,q}$.

Definition 6.4. Given an i.d.p., we may speak of its highest invariant derivative term, which we denote IHDT, the coefficient of the highest invariant derivative term which we denote IHcoeff, and so forth, (cf. §2.1) and we do so without further comment.

In addition, we need to define the highest monomial of an i.d.p.

Definition 6.5. The *head* of f is defined to be $\mathrm{Head}(f) = \mathrm{IHcoeff}(f) \bullet \mathrm{IHDT}(f)^{\mathrm{Hp}(f)}$. The *highest monomial*, $\mathrm{IHmon}(f)$, is defined recursively as follows: if f is a monomial, $\mathrm{IHmon}(f) = f$, else $\mathrm{IHmon}(f) = \mathrm{IHmon}(\mathrm{Head}(f))$. In the algebraic Gröbner basis literature, the highest monomials are often referred to as "initials", "leaders" or "tips".

Recall the normalisation equations are given by a set of equations $\psi_\lambda(J^j, I_K^\alpha) = 0$, $\lambda = 1, \ldots, r = \dim(G)$, $|K| \leq N$.

Definition 6.6. We denote by $\mathcal{N}$ the set of normalisation equations, while recall $\mathcal{HNI}$ is the set of highest derivative terms of the normalisation equations. We call the set $\mathcal{HNI}$ the set of *highest normalised invariants.*

Example 6.7. If the moving frame equations are

$$y_i = 0, \qquad v_2 = 0, \qquad v_3 = 0, \qquad 2v_{22} - v_{33} = 0$$

then the normalisation equations are J^i=0, i=1,2,3, $I_2 = 0$, $I_3 = 0$ and $2I_{22} - I_{33} = 0$. In an ordering with $y_2 < y_3$ the highest normalised invariants are I_2, I_3 and I_{33}.

Remark: In a computer algebra environment, invariantisation consists of translating each term u_K^α to I_K^α, by virtue of the Fels-Olver-Thomas translation theorem, and then simplifying with respect to the normalisation equations. The simplification is effected by computing a reduced algebraic Gröbner basis of the normalisation equations and then the unique normal form is given as the simplification. There is no loss in generality in assuming that $\mathcal{N}$ is a reduced Gröbner basis for the algebraic ideal it generates.

6.2 Invariantised cross differentiation and reduction formulae

The pseudo-reduction and cross-differentiation formulae are easily translated to formulae for invariantised differential polynomials using the Fels–Olver–Thomas replacement theorem, the only additional twist being that we need to take numerators, and we speak of the pseudo-reduction of an i.d.p. f with respect to the i.d.p. g, and the diffSpolynomial of two i.d.p.s. However, the result may not exist, or be well-defined. The following discussion identifies the conditions under which these calculations may be performed with confidence.

Because the invariant derivative operators do not commute, it is necessary to distinguish *paths of differentiation* in the various formulae.

Definition 6.8. A path of differentiation is a sequence $\{(K_1^i, \ldots, K_p^i)\}$ in the integer lattice $\mathbb{N}^p$ where each component is an increasing sequence in $\mathbb{N}$ and for each i there is a unique j such that $K_j^i < K_j^{i+1}$.

We take one integer lattice, $\mathbb{N}^{p,\alpha}$, for each dependent variable u^α.

Definition 6.9. A point $(K_1, \ldots, K_p)$ in the lattice $\mathbb{N}^{p,\alpha}$ is said to be *associated* with the invariant I_K^α where K is the multi index consisting of K_1 1's, K_2 2's, and so forth.

Recall that by construction, the invariantised derivative terms I_K^α are symmetric in the index K, that is, $I_{12} = I_{21}$ and so forth, and thus the identification of the point $(K_1, \dots, K_p) \in \mathbb{N}^{p,\alpha}$ with I_K^α is well-defined. It is the correction terms M_{Kj} in the invariant differentiation formula that are not symmetric in their indices, leading to path dependent invariant differentiation.

Definition 6.10. We say two paths of differentiation in $\mathbb{N}^{p,\alpha}$ with the same beginning and end points are *comparable* if one may be deformed to the other through paths of differentiation without ever passing through a lattice point associated with a highest normalised invariant (HNI).

Definition 6.11. Given two i.d.p.'s f_1, f_2 such that $\mathrm{IHDT}(f_1) = I^\alpha_{K^1}$, $\mathrm{IHDT}(f_2) = I^\alpha_{K^2}$, let L^1, L^2 be the unique indices of least order such that $K^1L^1 = K^2L^2$. Then we call $I^\alpha_{K^1L^1}$ the *least common invariant derivative term* of f_1 and f_2 and denote it by $\mathrm{LCD}(f_1, f_2)$.

Suppose we are given two i.d.p.'s f_1, f_2 whose highest invariant derivative terms are associated to the same dependent variable u^α. Calculating the invariantised diffSpolynomial of f_1, f_2 will not be possible if the path of differentiation of one IHDT to the $\mathrm{LCD}(f_1, f_2)$ must go through a highest normalised invariant, HNI (see Figure 1). In this case, it is not then possible to obtain two equations for $\mathrm{LCD}(f_1, f_2)$; what happens is that the HNI never appears as it is immediately simplified by the normalisation equations (this simplification is built into the invariantised differentiation formulae) and therefore the derivatives of the HNI do not appear either.

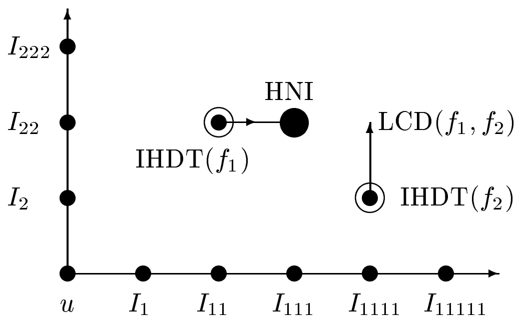

Figure 1: The diffSpolynomial(f_1, f_2) does not exist.

However, diffSpolynomials of *derivatives* of the f_i may exist. Depending on the locations of the HNI in the lattice, for any two i.d.p.'s f_1, f_2, there may be several diffSpolynomials

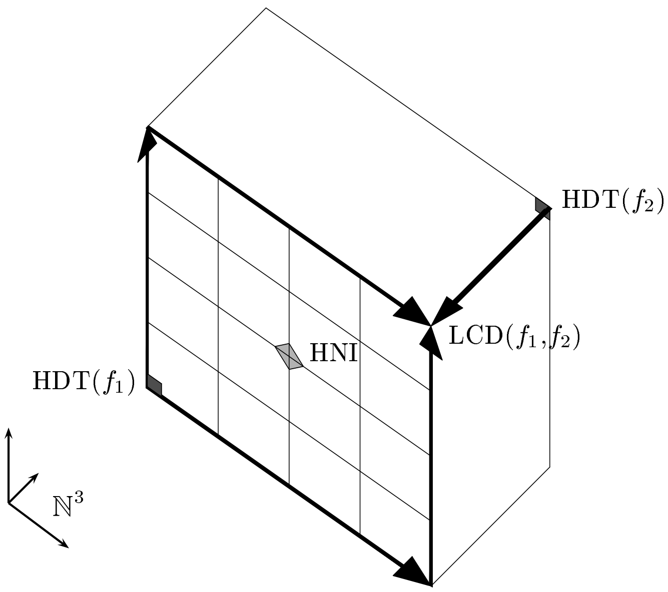

Figure 2: diffSpolynomials may be path dependent

of derivatives of the f_i on non-comparable paths of differentiation. The situation is in stark contrast to that of the non-invariantised case, where diffSpolynomials of derivatives of differential polynomials lead to no new integrability conditions.

In higher dimensions, there may be several diffSpolynomials of the f_i themselves that are not comparable. This is illustrated in Figure 2, where two non-comparable paths taking HDT(f_1) to LCD(f_1, f_2) exist.

The same considerations occur for the reduction of one i.d.p. with respect to another. Reductions may be path dependent, or may not exist at all.

All these difficulties exist only for equations whose highest derivative terms I_K^α satisfy $|K| < N+1$ where N is the order of the normalisation equations. The next definition makes precise the region of the lattices where we may experience these kinds of difficulties.

Definition 6.12. We define N^α to be order of the normalisation equations whose IHDT's are invariantised derivatives of u^α;

$$N^\alpha = \max\{\,-2,\, |K| \,|\, I_K^\alpha \in \mathcal{HNI}\}. \tag{51}$$

The map Norder : $\mathcal{I}_{p,q} \to \mathbb{Z}$ is given by

$$\mathrm{Norder}(f) = N^\alpha, \qquad \text{where } \mathrm{HDT}(f) = I_K^\alpha \tag{52}$$

for some K.

Note that the map Norder depends on the dependent variable u^α occurring in the highest derivative term of the argument.

Definition 6.13. We define the *order* of an invariantised derivative term to be,

$$\mathrm{ord}(I^\alpha_K) = |K|, \tag{53}$$

and by extension, the order of an i.d.p. to be the order of its highest invariant derivative term. Finally, we say an i.d.p. f is $\mathcal{N}$*-bounded* if

$$\mathrm{ord}(f) \leq \mathrm{Norder}(f) + 1,$$

and we say a set $\mathcal{A}$ of i.d.p.'s is $\mathcal{N}$-bounded if for all $f \in \mathcal{A}$, f is $\mathcal{N}$-bounded.

We can summarise the discussion in this section by the following

Theorem 6.14. Analogues of pseudo-reduction and cross differentiation may not exist, or may be path dependent, for $\mathcal{N}$-bounded invariant differential polynomials.

The $\mathcal{N}$-bounded invariantised differential polynomials need to be handled carefully. It will be necessary to modify the Kolchin Ritt algorithm to take into account the difficulties experienced in this "inner region" of the lattice.

The `IKolRitt` algorithm we describe in §6.3 will calculate a representative diffSpolynomial for every "critical path pair". Next we define this notion. The first definition we make is the analogue of the concept of a critical pair as it is standardly used in the computational algebra literature.

Definition 6.15. Two i.d.p.'s whose highest invariants are associated with the same dependent variable u^α, are said to be a *critical pair* provided at least one diffSpolynomial or a diffSpolynomial of a derivative of them, may be calculated.

Because of the path-dependence of diffSpolynomials, the notion of a critical pair needs to be extended to include the path of differentiation.

Definition 6.16. Let two not necessarily distinct i.d.p.'s f_1 and f_2 of a given system, be a critical pair, and let P_1 and P_2 be paths of differentiation from their respective IHDT's to some common derivative term IDT. We say that

$$(P_1, P_2) \sim (P_1', P_2') \tag{54}$$

if one path pair may be deformed to the other through paths of differentiation, keeping their endpoints equal, without passing though any highest normalised invariant. The equivalence class of such a path pair is said to be a *critical path pair* for f_1 and f_2.

This equivalence relation partitions, in some sense, the set of diffSpolynomials that may be calculated for a given system. Figure 3 in §7.2 shows how a nontrivial diffSpolynomial between a single i.d.p. and itself may arise. Note that diffSpolynomials are not constant on these partitions. We will conjecture that provided at least one diffSpolynomial from each partition pseudo-reduces to zero, then they all will.

6.3 The Procedures

Typically we are given Σ, a set of differential polynomials, and the action of the Lie Group G on the independent and dependent variables. The invariant differentiation formulae mean that we need only the infinitesimal action of the group on these variables. The choice of the normalisation equations $\mathcal{N}$ must be compatible with the system Σ as well as providing a bona fide moving frame. From this information, we may easily compute the set $\mathcal{HNI}$, the map Norder and the correction matrix K. In what follows, we assume that these initial calculations have been done.

In the following procedures, the function `simplify` with respect to a set of polynomials $\mathcal{C}$ refers to the process of finding an algebraic Gröbner basis of $\mathcal{C}$ and taking the unique normal form.

6.3.1 Procedure `Inner`

This first procedure is for i.d.p.'s which are $\mathcal{N}$-bounded. The difficulties outlined above mean that diffSpolynomial and pseudo-reduction calculations are not appropriate. Thus, we revert to a primitive form of "differential Gröbner basis" calculation, in which equations are differentiated but *algebraic* Gröbner bases calculated, as though the differential equations were polynomials with each derivative term regarded as a separate indeterminant. Several differential elimination algorithms use a prolong and algebraic simplification mechanism (cf. [5, 39]). The path dependence of the calculations is incorporated by performing all possible differentiations.

We have observed that invariantised differentiation may introduce denominators into our polynomials, and these need to be removed. Repeated invariantised differentiation leads to a build up of factors of denominators of elements of the matrix K. These must be assumed to be non-zero when evaluated on the input system, and hence in the following we remove all such factors in the simplification processes.

In algebraic Gröbner basis algorithms, it is standard that "new" conditions are recognised by their highest monomials not belonging to the current set of highest monomials $\mathcal{IN}$, known in the commutative algebra literature as the "initial set".

Procedure `Inner`($\mathcal{A}$, τ, '$\mathcal{C}$', '$\mathcal{P}_{outer}$')

Input: an $\mathcal{N}$-bounded set $\mathcal{A} \subset \mathcal{I}_{p,q}$

a total degree termordering τ

Output: two sets $\mathcal{C}$, $\mathcal{P}_{outer} \subset \mathcal{I}_{p,q}$

$N := \max\{N^\alpha\}$; $\mathcal{C} := \{\}$; $\mathcal{P}_{outer} := \{\}$;
$\mathcal{IN} := \{\}$; $\mathcal{P} := \mathcal{A}$;
while $\mathcal{P} \neq \{\}$ do
  $\mathcal{IN} := \mathcal{IN} \cup \{\mathrm{IHmon}(f) \mid f \in \mathcal{P}\}$;
  for j from 0 to N do $\mathcal{M}_j := \{f \in \mathcal{P} \mid \mathrm{ord}(f) = j\}$
  $\mathcal{P} := \{\}$;
  for n from 0 to N
    for $f \in \mathcal{M}_n$
      for j to p
        $g := \mathrm{simplify}\,(\mathrm{numer}(\mathcal{D}_j(f)), \mathcal{C})$;
        if $g = 0$ then next j
        if $\mathrm{ord}(g) \geq \mathrm{Norder}(g) + 1$ then
          $\mathcal{P}_{outer} := \mathcal{P}_{outer} \cup \{g\}$
        if $\mathrm{ord}(g) \leq n$ then
          $\mathcal{P} := \mathcal{P} \cup \{g\}$;
        if $\mathrm{Norder}(g) \geq \mathrm{ord}(g) > n$ then
          $\mathcal{M}_{\mathrm{ord}(g)} := \mathcal{M}_{\mathrm{ord}(g)} \cup \{g\}$
  $\mathcal{C} :=$ Gröbner Basis$(\mathcal{C} \cup \mathcal{M}_0 \cup \ldots \cup \mathcal{M}_N \cup \mathcal{P}_{outer}, \tau)$
  $\mathcal{P} := \mathcal{P} \cup \{f \in \mathcal{C} \mid \mathrm{IHmon}(f) \notin \mathcal{IN},\ \mathrm{ord}(f) \leq\ \mathrm{Norder}(f) + 1\}$

The output sets of the `Inner` procedure, $\mathcal{C}$, and $\mathcal{P}_{outer}$, satisfy the following properties, by construction.

Theorem 6.17. The output set $\mathcal{C}$ is $\mathcal{N}$-bounded and a Gröbner basis. For all $f \in \mathcal{C}$ and all $j = 1, \ldots, p$, either $\mathcal{D}_j(f)$ reduces to zero with respect to $\mathcal{C}$, or $\mathrm{ord}(\mathcal{D}_j f) > \mathrm{Norder}(f) + 1$. For all $f \in \mathcal{P}_{outer}$, $\mathrm{ord}(f) \geq \mathrm{Norder}(f) + 1$.

The set $\mathcal{P}_{outer}$ are those integrability conditions found *en route* which are either not $\mathcal{N}$-bounded or on the boundary of the $\mathcal{N}$-bounded region. This set may contain non $\mathcal{N}$-bounded elements if the **mtdeg** ordering is used, in which case conditions with a different highest unknown, and thus a different value under the map $Norder$, may result.

6.3.2 The `Outer` Procedure

The `Outer` procedure is the obvious direct translation of the Kolchin Ritt algorithm for invariantised polynomials, and is valid in the outer region only. There is one new twist, which is we perform algebraic simplifications with respect to a set $\mathcal{A}$ of $\mathcal{N}$-bounded polynomials. In some sense, we are finding a characteristic set "relative" to $\mathcal{A}$, which will of course be the output of the `Inner` procedure.

In this algorithm, the diffSpolynomial calculated is the standard one, obtained by taking straight line paths of integration from the IHDT's of f_i and f_j to their least common derivative term. The function $\mathrm{normal}^p(\bullet, \mathcal{C})$ means the normal form with respect to differential pseudo-reduction by elements of the set $\mathcal{C}$.

Procedure `Outer`($\mathcal{B}$,τ,$\mathcal{A}$,'$\mathcal{C}$','$\mathcal{P}_{inner}$')

Input: a set $\mathcal{B} \subset \mathcal{I}_{p,q}$ such that $\mathrm{ord}(f) \geq \mathrm{Norder}(f) + 1$, all $f \in \mathcal{B}$
an $\mathcal{N}$-bounded set $\mathcal{A} \subset \mathcal{I}_{p,q}$
a total degree term ordering τ

Output: two sets $\mathcal{C}$, $\mathcal{P}_{inner} \subset \mathcal{I}_{p,q}$

$\mathcal{P}_{inner} := \{\}$;
$\mathcal{C} := \mathcal{B}$;
pairset:=$\{ \{f_i, f_j\} \mid f_i,\ f_j \in \mathcal{B},\ f_i \neq f_j\}$
while pairset$\neq \{\}$ do
  for $\{ \{f_i, f_j\}\} \in$ pairset do
    pairset:=pairset $\setminus \{ \{f_i, f_j\}\}$

$m :=$ normal^p(diffSpolynomial(f_i, f_j),$\mathcal{C}$)
$m :=$ simplify(m,$\mathcal{P}_{inner} \cup \mathcal{A}$)
if $m \neq 0$ then
  if ord$(m) \leq$ Norder$(m) + 1$ then
    $\mathcal{P}_{inner} := \mathcal{P}_{inner} \cup \{m\}$
  if ord$(m) \geq$ Norder$(m) + 1$ then
    pairset:=pairset $\cup\{\ \{f, m\} \mid f \in \mathcal{C}\}$
  $\mathcal{C} := \mathcal{C} \cup \{m\}$

By construction, the output of the `Outer` procedure is two sets $\mathcal{C}$ and $\mathcal{P}_{inner}$ of i.d.p.s satisfying the following result.

Theorem 6.18. The (pseudo)-reduction of all invariantised standard diffSpolynomials of elements of $\mathcal{C}$ with respect to $\mathcal{C}$ and $\mathcal{A}$ is either zero or is an element of $\mathcal{P}_{inner}$, and every element of $\mathcal{P}_{inner}$ satisfies ord$(f) \leq$ Norder$(f) + 1$.

The elements of $\mathcal{P}_{inner}$ are those integrability conditions found *en route* whose order places them in the $\mathcal{N}$-bounded region or on its boundary.

6.3.3 The `IKolRitt` Procedure

Finally the invariantised Kolchin Ritt procedure `IKolRitt` combines the `Inner` and `Outer` procedures. It can be seen as a translation of Buchberger's algorithm for completion of polynomial systems to a Gröbner basis where there are two regions, an inner and an outer. In the inner region, algebraic Spolynomials and reductions are calculated, with all equations differentiated to one plus the order of the normalisation equations, while in the outer region, differential Spolynomials and differential pseudo-reductions are calculated. Because it is possible that the inner calculation finds integrability conditions in the outer region, and the outer calculation finds integrability conditions in the inner region, the inner and outer procedures loop until no further integrability conditions are found.

Procedure `IKolRitt`$(\Sigma, \tau,' \mathcal{C}')$
Input: a finite set $\Sigma \subset \mathcal{I}_{p,q}$
  a total degree term ordering τ

Output: a set $\mathcal{C} \subset \mathcal{I}_{p,q}$

$\mathcal{P} := \Sigma;$

$\mathcal{A} := \{ f \in \Sigma \,|\, \mathrm{ord}(f) \leq \mathrm{Norder}(f) + 1 \}$

$\mathcal{B} := \Sigma \setminus \mathcal{A}$

while $\mathcal{P} \neq \{\}$ do

  `Inner`($\mathcal{A}$,τ,'$\mathcal{A}$','$\mathcal{A}_{outer}$')

  $\mathcal{B} :=$ simplify$(\mathcal{B}, \mathcal{A}) \cup \mathcal{A}_{outer}$

  `Outer`($\mathcal{B}$,τ,$\mathcal{A}$,'$\mathcal{B}$','$\mathcal{P}$');

  $\mathcal{A} := \mathcal{A} \cup \mathcal{P};$

  $\mathcal{B} := \mathcal{B} \setminus \mathcal{P};$

$\mathcal{C} := \mathcal{A} \cup \mathcal{B};$

As written here, the procedure is hopelessly inefficient. With careful bookkeeping, however, (diff)Spolynomials will not be calculated more than once.

Theorem 6.19. The `IKolRitt` procedure terminates by the usual arguments involving Dickson's lemma (a simple discussion of these arguments is given in [22]).

Theorem 6.20. If the group action is trivial, the IKolRitt procedure reduces to the Kolchin-Ritt algorithm.

Proof: If there are no normalisation equations then for every $f \in \mathcal{I}_{p,q}$, $\mathrm{Norder}(f) = -2$ and so $\mathrm{ord}(f) > \mathrm{Norder}(f) + 1$. Hence the sets $\mathcal{A}$ and $\mathcal{A}_{outer}$ calculated in `IKolRitt` are identically empty. Since the differential operators are the usual $\partial/\partial x_j$, `IKolRitt` reduces to the Kolchin Ritt algorithm.

It can happen that the group action is not trivial but that the normalisation equations are in terms of differential parameters appearing in the system (this can happen in classification problems). In this case the sets $\mathcal{A}$ and $\mathcal{A}_{outer}$ calculated in `IKolRitt` are identically empty. However, if the differential operators are non-commutative, `IKolRitt` will not reduce to the Kolchin Ritt algorithm.

Just as for the Kolchin-Ritt algorithm, it is possible to collect those nontrivial expressions by which i.d.p.'s are multiplied during the course of a pseudo-reduction calculation. These will be the IHcoeffs and separants of the i.d.p.'s given or obtained en route. These expressions are assumed to be nonzero. Setting these by turn to zero and recalculating the algorithm leads to a branching calculation which seeks the so-called singular solution varieties. See [19] for a rigorous discussion of a branching algorithm for characteristic set decomposition of differential ideals.

7 Discussion concerning properties of the output

In this section we discuss the property of coherence, which is the central concept leading to output properties of several differential elimination algorithms, including the Kolchin-Ritt algorithm. Then we discuss the role of the normalisation equations and the syzygies defined in §3.6. We end with some brief remarks concerning the Buchberger second criterion, which is used to eliminate unnecessary cross-differentiation calculations.

7.1 Coherence

By construction, the output G of standard Kolchin Ritt algorithm has the property that for all pairs of differential polynomials in G, their diffSpolynomial pseudo-reduces to zero with respect to G. From this property, known as *coherence*, the output statement of the Kolchin Ritt algorithm can be deduced. We examine the analogous property for invariantised systems. Recall the notion of a critical path pair from §6.2.

Theorem 7.1. The output of the `IKolRitt` procedure has the property that at least one diffSpolynomial for every critical path pair class (pseudo-)reduces to zero.

Proof: By construction, the standard diffSpolynomial of every pair of equations in the outer region pseudo-reduces to zero with respect to the output, and such pairs of equations have only one critical path pair class. Consider next an i.d.p. f_1 in the inner region and an i.d.p. f_2 in the outer region. An integrability condition may be obtained by differentiating f_1 to the boundary of the inner region; this occurs in the course of the `Inner` procedure. This derivative is passed to the input system of the `Outer` procedure, and so a diffSpolynomial between f_2 and the derivative of f_1 will be calculated. Since all possible paths of differentiation of f_1 to the boundary are calculated, a diffSpolynomial from each critical path pair class will be calculated in this way. We remark that it is possible that *every* path of differentiation of f_1 to the boundary passes through an HNI, so that no integrability condition with f_2 exists. The same arguments can be made for two i.d.p.'s in the inner region.

For the standard Kolchin Ritt algorithm, it is sufficient to calculate the minimal (standard) diffSpolynomial for each pair of equations given or obtain en route. That is, it is not necessary to calculate diffSpolynomials of derivatives of the members of the input system. The reason is that the reduction to zero of the diffSpolynomial of any derivatives is guaranteed, provided *all* the minimal diffSpolynomials reduce to zero[23]. We give here a conjecture of the analogous result for `IKolRitt`.

Conjecture 1 The output system of `IKolRitt` has the property that every diffSpolynomial of every critical path pair, (pseudo-)reduces to zero.

In the outer region, calculation of pseudo-reductions, cross derivatives and so forth, are not affected by the non-commutativity of the differential operators provided one is using a total degree term ordering. This is because the correction terms are bounded by the order of the normalisation equations. Hence proofs of properties of output that depend only on the highest order terms should be adaptable to the invariantised procedures. Hence we make the following definitions and conjecture.

Definition 7.2. For an arbitrary set of i.d.p.'s, $G \subset \mathcal{I}_{p,q}$, define

$$\mathcal{O}(G) = \{f \in G \,|\, \mathrm{ord}(f) > \mathrm{Norder(f)}\}$$

Further, let $I(G)$ denote the set of all i.d.p.'s obtainable from G by invariant differentiation, taking numerators, adding, and multiplying by elements in $\mathcal{I}_{p,q}$ a finite number of times. Finally we denote by $S(G)$ the multiplicative set generated by all factors of highest coefficients and separants of elements of G.

Conjecture 2 If $\mathcal{C}$ denotes the output of `IKolRitt`, then for all $f \in I(\mathcal{O}(\mathcal{C}))$ there exists $s \in S(\mathcal{C})$ such that sf pseudo-reduces to 0 with respect to $\mathcal{C}$.

7.2 The role of the normalisation equations and syzygies

The normalisation equations are used to simplify algebraically the input system during the invariantisation process, and also the result of every invariant differentiation, and thus they are effectively additional constraints on the input system.

Theorem 7.3. There are two ways in which `IKolRitt` can yield an inconsistency. The first is that the input system is inconsistent. The second is that the normalisation equations are inconsistent with the input system.

As stated in (3.6), one big difference between invariantised and non-invariantised systems is the existence of nontrivial syzygies. We discuss next how `IKolRitt` interacts with these and the role they play in the analysis of the input system.

The syzygies of the moving frame, given by (26) and (28), are the fundamental compatibility conditions of the frame itself, and must hold irrespective of any additional constraints imposed by the system of i.d.p.'s under consideration. The syzygies must be zero when evaluated on this system. In particular, if the correction matrix K is simplified with respect to the system being analysed (to prevent multiple reductions of elements of the correction matrix K which are introduced with every invariant differentiation), then the normalisation equations need to be added to the system to prevent loss of information.

Definition 7.4. We say a syzygy is *relevant* to a set of i.d.p.'s if there exist two (not necessarily distinct) equations f_1, f_2 in the set so that the left hand side of the syzygy (28) is expressible in terms of IHDT(f_1) and IHDT(f_2). This will be the case when, for example, the IHDTs are I^{α}_{JK}, I^{α}_{LM}, and I^{α}_{J}, I^{α}_{L} are fundamental generating invariants, or vice versa.

An example of how `IKolRitt` computes a relevant syzygy on the input system is shown in Figure 3. In this example, the two fundamental invariants are FI_1, FI_2, and the syzygy as evaluated on the system is computed as the diffSpolynomial of derivatives of f.

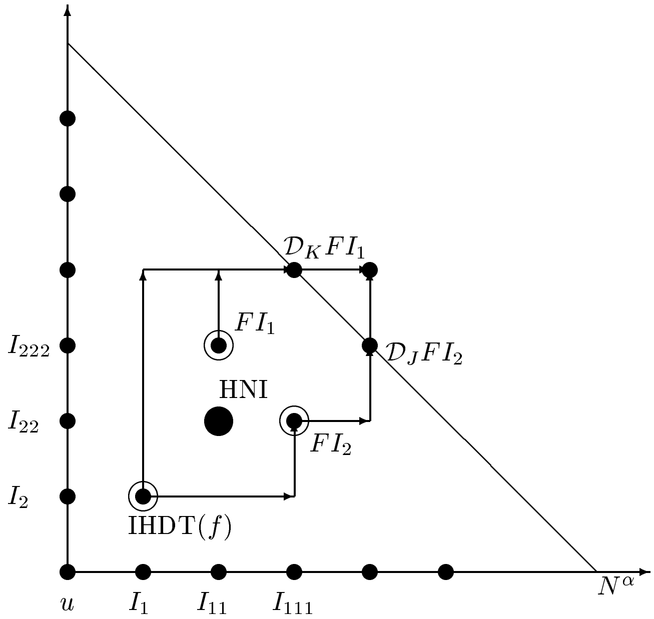

Figure 3: How a relevant syzygy between the fundamental invariants FI_1,FI_2 evaluated on the system may be computed by `IKolRitt`.

Conjecture 3 DiffSpolynomials evaluated on inequivalent critical path pairs differ by a relevant syzygy. Further, every relevant syzygy (pseudo)-reduces to zero with respect to the output of `IKolRitt`.

Even if Conjectures 1, 2 and 3 are proven, it is not clear that a completeness property similar to that for the non-invariantised algorithm can be readily deduced. As far as the author is aware, no results in differential algebra, commutative algebra or non-commutative Gröbner basis theories apply. Further analysis of the `IKolRitt` procedure awaits more experience with applications and examples, as well as theoretical developments.

7.3 The Buchberger Second Criterion

If the set $\mathcal{P}_{outer}$ calculated in the `Inner` procedure is not output in simplified form, then a much larger number of diffSpolynomials need to be calculated in the `Outer` procedure. Thus, in order to improve the efficiency of the algorithm, one would want $\mathcal{P}_{outer}$ to be inter-reduced. However, at least on the face of it, this could lead to possible diffSpolynomials not being calculated. To see this, consider the diffSpolynomials of an i.d.p., f_1, in the inner region with f_2 (say) in the outer region. If some derivative of f_1 does not appear in the output of `Inner` procedure, one of the (path-dependent) diffSpolynomials of f_1 with f_2 may not be calculated.

In order to output inter-reduced sets from the `Inner` procedure with impunity, as well as for a major efficiency advantage in the `Outer` procedure, what is required is a version of the Buchberger second criterion [2] (Proposition 5.70) appropriate for invariantised systems. The original Buchberger second criterion is implemented as a standard feature in algebraic Gröbner basis packages as it eliminates the calculation of Spolynomials which may be guaranteed to lead to no new information. In its simplest form suitable for i.d.p.'s, this criterion would look like:

> if $\{f_1, f\}$ and $\{f_2, f\}$ have been "treated" and the index of differentiation associated with $\mathrm{IHDT}(f)$ divides the index of differentiation associated with $\mathrm{LCD}(f_1, f_2)$, then the pair $\{f_1, f_2\}$ does not need to be "treated".

In the implementation of `IKolRitt` discussed in the next section, it is assumed that this criterion is valid in the `Outer` procedure. The reason is that the correction terms are of order strictly less than the highest derivative terms, and thus the proof of such a criterion should not be affected. Further, we assume that outputting inter-reduced sets from the

`Inner` procedure leads to no loss of information.

8 Conclusions, comparisons and implementation

In this article we have shown that many problems arise when trying to analyze a Gröbner basis type procedure for nonlinear differential equations, written in terms of invariants of a Lie group action. Despite these problems, the methods are a worth-while addition to the applied mathematician's armory. The potential of the method to obtain both an increase in efficiency and a substantial decrease in expression swell over the non invariantised Kolchin-Ritt algorithm is excellent: the invariantised procedure is able to analyze the $2+1$-d'Alembert-Hamilton system in seconds (see §8.2.2) whereas the non-invariantised calculation is intractable. There are several reasons why improvements might hold for general classes of systems.

1. The normalisation equations are effectively extra conditions, making the system more over determined, and therefore more tractable. Indeed, the input system has been projected down to a space with fewer variables, namely, the transversal $\mathcal{K}$ of the moving frame (§3.3).

2. The fundamental syzygies of the normalisation equations evaluated on the system introduce additional integrability conditions, leading to further simplifications.

3. Judicious choice of the normalisation equations can lead to substantial simplification of the input system when it is invariantised. In particular, nonlinearities may be removed.

Thus, the choice of normalisation equations is crucial. Not only must they be compatible with the input system, but they should be chosen to simplify the input system as much as possible. Further, their order should be as small as possible to minimize the calculations of the `Inner` procedure.

8.1 Comparison with the method due to Lisle

In [21], Lisle introduced a "moving frame" of operators which corresponds to the set of invariant operators $\{\mathcal{D}_j \,|\, j=1,\dots,p\}$ used here. The over determined systems considered were rewritten as polynomials in terms of the form $\mathcal{D}_K I^\alpha$ and a completion to standard form algorithm was given, denoted "frame standard form algorithm". With hindsight, in terms of the concepts used in this article, the examples discussed had normalisation equations which ensured that none of the difficulties with defining diffSpolynomials and pseudo-reductions arose. A major efficiency problem with the Lisle method is that extensive use of the commutation formulae for the $\mathcal{D}_j$ is needed. This problem is completely avoided in the procedure `IKolRitt` since it uses differential polynomials composed of the I^α_K which are invariant under permutations in the index K, whereas the $\mathcal{D}_K I^\alpha$ are not.

Lisle's application was the classification of Lie symmetries of differential equations containing arbitrary functions of the dependent and independent variables. The over determined system is for the infinitesimals, and the symmetry group of the over determined system is the equivalence group of the equation [33]. Indications are that Lisle's ideas will lead to large gains in efficiency in computing the classification [27]. The use of the methods described in the current article should lead to further gains in efficiency.

8.2 Implementation

The procedures given in §6.3 have been implemented as a package `Indiff` in MAPLE V5[1]. The implementation assumes that the above conjectures concerning the applicability of Buchberger's second criterion are true. We give here two small examples with brief explanations, to show how the package may be used. The group prolongation code called by `Indiff` is that of `Desolv`, written by Khai The Vu [41] who was inspired by the remarkable muMATH programme `LIE` by Alan Head [18].

8.2.1 Example 1: invariant differentiation

This first example (taken from [29]) shows invariant differentiation and derivation of the moving frame for the equiaffine group acting on the variables $(x, u(x))$ as

$$(x, u(x)) \mapsto \begin{cases} (x+a, u(x)) \\ (x, u(x)+b) \\ (\alpha x, u(x)/\alpha) \\ (x+\beta u(x), u(x)) \\ (x, u(x)+\delta x) \end{cases}$$

[1]See the URL `http://www.ukc.ac.uk/ims/maths/people/E.L.Mansfield.html` for information on how to obtain the code and manual.

The first step is to load the package `Indiff` and enter the names of all variables. The independent variables are denoted `vars`, the dependent variables `ukns`, and the group parameter names by `GroupP`. This last is not necessary from a programming point of view, but is a necessary aid for the human user. The symmetry group action is given in infinitesimal form. The (κ, i)th element of `XiPhis` is the infinitesimal of the κth group parameter on the ith variable of the concatenated list `vars, ukns`. The invariant derivative term I^{α}_{K} is denoted `In[`u^{α}`,K]`, while the operator $\mathcal{D}_j$ is calculated using the function `f` $\mapsto$ `Idiff(f,j);`.

```
> restart:with(Indiff);
```

$$[HNI,\ IKolRitt,\ IdSpoly,\ Idiff,\ Idiffparse,\ Invariantize,\ Iorthreduceall,\ Ireduce,\ Ireduceall,\ Kmat]$$

```
> vars:=[x]:ukns:=[u]:GroupP:=[a,b,alpha,beta,delta]:
> XiPhis:=matrix([[1,0],[0,1],[x,-u(x)],[u(x),0],[0,x]]);
```

$$XiPhis := \begin{bmatrix} 1 & 0 \\ 0 & 1 \\ x & -\mathrm{u}(x) \\ \mathrm{u}(x) & 0 \\ 0 & x \end{bmatrix}$$

The normalisation equations, `Neqs`, are the invariantised moving frame equations. The function `HNI` calculates the set $\mathcal{HNI}$, denoted by the global variable `HNIlist`, and information to calculate the map Norder is collected in the global variable `DegDiffTable`. The function `Kmat` calculates $-\mathbf{K}$.

```
> Neqs:=[In[x],In[u,[]],In[u,[1]],In[u,[1,1]]-1,In[u,[1,1,1]]]:
> HNI([[1],[u]],ttdeg):
> Kmat();
```

$$\begin{bmatrix} 1 & 0 & 0 & -\frac{1}{3}\, In_{u,\,[1,1,1,1]} & 1 \end{bmatrix}$$

We show the invariant differentiations of each of $I_{111}, \ldots, I_{111111}$. Since I_{111} is a normalised invariant, its derivative is zero.

```
> for j from 3 to 6 do Idiff(In[u,[1$j]],1) od;
```

$$
\begin{gathered}
0 \\
In_{u,[1,1,1,1,1]} \\
In_{u,[1,1,1,1,1,1]} - 5\, {In_{u,[1,1,1,1]}}^2 \\
In_{u,[1,1,1,1,1,1,1]} - 7\, In_{u,[1,1,1,1]}\, In_{u,[1,1,1,1,1]}
\end{gathered}
\tag{55}
$$

To actually see what the invariants and the moving frame equations are in terms of derivatives of u, and to verify that `Neqs` determines a bona fide moving frame, the MAPLE library package `PDETools[dchange]` written by E. Cheb-Terrab is useful. We first enter the variable transformation,

$$
tr := \{y = \frac{(1+\beta\,\delta)\,(x-a)}{\alpha} - \frac{\beta\,(\mathrm{u}(x)-b)}{\alpha},\ \mathrm{v}(y) = -\delta\,\alpha\,(x-a) + (\mathrm{u}(x)-b)\,\alpha\}
$$

The translation parameters are normalised to be $(a, b) = (x, u)$. Calculating the first three lifted derivatives to find suitable equations for the rest of the moving frame, the first is,

```
> normal(dchange(tr,diff(v(y),y),[x,u(x)]));
```

$$
-\frac{\alpha^2\,(\delta - (\frac{\partial}{\partial x}\,\mathrm{u}(x)))}{1+\beta\,\delta - \beta\,(\frac{\partial}{\partial x}\,\mathrm{u}(x))}
$$

If we take $v_y = 0$ as the first equation, we obtain $\delta = u_x$. Continuing, we see that the normalisation equations as given in `Neqs` yields a bona fide moving frame:

$$
\mathit{frame} := \delta = \frac{\partial}{\partial x}\,\mathrm{u}(x),\ \alpha = \frac{1}{(\frac{\partial^2}{\partial x^2}\,\mathrm{u}(x))^{(1/3)}},\ \beta = -\frac{1}{3}\,\frac{\frac{\partial^3}{\partial x^3}\,\mathrm{u}(x)}{(\frac{\partial^2}{\partial x^2}\,\mathrm{u}(x))^2}
$$

Using `dchange` and `frame`, one may now calculate the fourth order and higher invariants. For example,

```
> dchange(tr,diff(v(y),y$4),[x,u(x)]):subs(frame,%):In[1111]:=factor(%);
```

$$
In_{1111} := \frac{1}{3}\,\frac{-5\,(\frac{\partial^3}{\partial x^3}\,\mathrm{u}(x))^2 + 3\,(\frac{\partial^2}{\partial x^2}\,\mathrm{u}(x))\,(\frac{\partial^4}{\partial x^4}\,\mathrm{u}(x))}{(\frac{\partial^2}{\partial x^2}\,\mathrm{u}(x))^{(8/3)}}
$$

Similarly one may obtain the invariant derivative operator, which is ${u_{xx}}^{-1/3}\partial/\partial x$, and the formulae (55) may be confirmed.

8.2.2 Example 2: analysis of an invariantised system

This second example shows the output of `IKolRitt` on the system (7) with $f(u) = 2/u$. The same information as for Example 1 must be entered. The result of `Kmat()` is the negative of

(37). We verify a selection of the normalisation equations have zero derivatives, and enter the invariantised system directly, having first ascertained that I_{11} is zero on the system.

The argument `[[1,3,2],[u]]` of `HNI` and `IKolRitt` indicates that an ordering based on $x < z < y$ is required. The set `Xset` consists of those expressions by which an i.d.p. is multiplied in the course of a pseudo-reduction. These must be non-zero. The `NonZero` set are those expressions which may be removed if they occur as a factor.

```
> restart:with(Indiff):
> vars:=[x,y,z]:ukns:=[u]:GroupP:=[k1,k2,k3,theta1,theta2,theta3]:
> XiPhis:=matrix([[1,0,0,0],[0,1,0,0],[0,0,1,0],[-y,x,0,0],[-z,0,x,0],
    [0,-z,y,0]]):
> Neqs:=[In[x],In[y],In[z],In[u,[2]],In[u,[3]],
   2*In[u,[2,2]]-In[u,[3,3]]]:
> HNI([[1,3,2],[u]],ttdeg):
> Kmat():
> seq(factor(Idiff(Neqs[6],j)),j=1..3);
```

$$0, 0, 0$$

```
> g1:=In[u,[1]]-1:
> Idiff(g1,1);
```

$$In_{u,[1,1]}$$

```
> g2:=3*In[u,[]]*In[u,[2,2]]-2:
> IKolRitt([g1,g2],[[1,3,2],[u]],ttdeg,'C',info= {'Xset'},
   NonZero={In[u,[]]});
```

```
the Xset is generated by   {In[u,[]], In[u,[2, 3]]}
```

$$\begin{aligned}[&In_{u,[2]},\ In_{u,[3]},\ In_{u,[1]}-1,\ 2\,In_{u,[2,2]}-In_{u,[3,3]},\ 3\,In_{u,[]}\,In_{u,[3,3]}-4,\\ &16\,{In_{u,[2,3]}}^2+{In_{u,[3,3]}}^2,\ In_{u,[1,2]},\ In_{u,[1,3]},\ In_{u,[1,1]},\\ &In_{u,[2,2,2]}-3\,In_{u,[]}\,In_{u,[2,3]}\,In_{u,[2,2,3]},\ -In_{u,[2,2,2]}-In_{u,[2,3,3]},\\ &In_{u,[2,2,3]}+In_{u,[3,3,3]},\ 16\,In_{u,[1,2,2]}+3\,{In_{u,[3,3]}}^2,\\ &2\,In_{u,[1,2,3]}+3\,In_{u,[2,3]}\,In_{u,[3,3]},\ 16\,In_{u,[1,3,3]}+15\,{In_{u,[3,3]}}^2,\ In_{u,[1,1,2]},\\ &In_{u,[1,1,3]},\ In_{u,[1,1,1]}]\end{aligned}$$

The `IKolRitt` calculation took 22.64 seconds on a 500MHz Pentium III processor having 512Mb RAM, with MAPLE V5 running under Red Hat Linux 6.0.

Acknowledgements

The author is indebted to Peter Olver who explained the moving frame method in a series of informal seminars at the Mathematical Sciences Research Institute, Berkeley, answered many questions and supplied a much needed more general formula for the correction matrix than that given in the original papers. The hospitality of the Mathematical Sciences Research Institute, Berkeley, where the research for this this paper was carried out, is gratefully acknowledged. In particular the author is grateful to Michael Singer who co-chaired the special semester in "Symbolic Computation in Geometry and Analysis" at the MSRI during the fall of 1998, for his encouragement and support. The author also thanks Peter Clarkson, Karin Gatermann, Ed Green, Evelyne Hubert and Greg Reid, for their helpful discussions concerning the research described here. Finally the author thanks the referees and the Editor for extensive comments on the exposition.

References

[1] M.J. Ablowitz and P.A. Clarkson, *Solitons, Nonlinear Evolution Equations and Inverse Scattering* L.M.S. Lect. Notes Math., vol. **149** (C.U.P., Cambridge, 1991).

[2] T. Becker and V. Weispfenning, *Gröbner Bases*, Graduate Texts in Mathematics, Springer Verlag, New York, 1993.

[3] F. Boulier, D. Lazard, F. Ollivier and M. Petitot, Representation for the radical of a finitely generated differential ideal, in: A. Levelt, ed., *Proc. ISSAC '95* ACM Press, New York, 1995.

[4] B. Buchberger, Applications of Gröbner bases in non-linear computational geometry, in: J.R. Rice, ed., *Mathematical Aspects of Scientific Software*, IMA Volumes in Mathematics and its Applications, **14** (Springer-Verlag, New York, 1988) 52–80.

[5] G. Carrà-Ferro, Groebner bases and differential algebra in: L. Huguet and A. Poli, Eds., Proceedings AAECC, *Lect. Notes Comp. Sci.*, **356** (1989) 129–140.

[6] P.A. Clarkson and P.J. Olver, Symmetry and the Chazy Equation, *J. Diff. Eq.* **124** (1996) 225–246.

[7] C.B. Collins, Complex potential equations I, *Math. Proc. Cambridge Philos. Soc.* **80** (1976), 165–187.

[8] C.M. Cosgrove, private communication, 1991.

[9] R. Courant and D. Hilbert, *Methods of Mathematical Physics*, Interscience Publications, John Wiley, New York, 1962.

[10] D. Cox, J. Little and D. O'Shea, *Ideals, Varieties and Algorithms*, Undergraduate Texts in Mathematics, Springer Verlag, New York, 1992.

[11] M. Fels and P.J. Olver, Moving Coframes I, *Acta Appl. Math.* **51** (1998) 161–213.

[12] M. Fels and P.J. Olver, Moving Coframes II, *Acta Appl. Math.* **55** (1999) 127–208.

[13] W.I. Fushchich and I.A. Yegorchenko, Second order differential invariants of the rotation group $O(n)$ and of its extensions: $E(n)$, $P(1,n)$, $G(1,n)$, *Acta Appl. Math.* **28** (1992) 69–92.

[14] W.I. Fushchich, R.Z. Zhdanov, I.A. Yegorchenko, On the reduction of the nonlinear multi-dimensional wave equations and compatibility of the d'Alembert-Hamilton system, *J. Math. Anal. App.* **161** (1991) 352–360.

[15] K. Gatermann and F. Guyard, Gröbner bases, invariant theory and equivariant dynamics, *J. Symb. Comp.* **28** (1999) 275–302.

[16] K.O. Geddes, S.R. Czapor and G. Labahn, Algorithms for Computer Algebra, Kluwer Academic Publishers, Boston 1992.

[17] D. Hartley and R.W. Tucker, A constructive implementation of the Cartan Kahler theory of exterior differential systems, *J. Symb. Comp.* **12** (1991) 655–667

[18] A.K. Head, `LIE`, a PC program for LIE analysis of differential equations, *Comp. Phys. Commun.* **96** (1996) 311–313.

[19] E. Hubert, Factorisation free decomposition algorithms in differential algebra, *J. Symb. Comp.*, **29** (2000) 641–662.

[20] N.H. Ibragimov [Ed.], CRC Handbook of Lie Group Analysis of Differential Equations, Volumes 1–3, CRC press, Boca Raton 1995.

[21] I.G. Lisle, *Equivalence Transformations for Classes of Differential Equations*, PhD Thesis, University of British Columbia, 1992.

[22] E.L. Mansfield and P.A. Clarkson, Applications of the differential algebra package `diffgrob2` to classical symmetries of differential equations, *J. Symb. Comp.* **23** (1997) 517–533.

[23] E.L. Mansfield and E.D. Fackerell, Differential Gröbner bases, preprint **92/108**, Macquarie University (1992).

[24] E.L. Mansfield, G.J. Reid and P.A. Clarkson, Nonclassical reductions of a 3+1 cubic nonlinear Schrödinger system, *Comp. Phys. Comm.* **115** (1998) 460–488.

[25] E.L. Mansfield, `diffgrob2`: A symbolic algebra package for analyzing systems of PDE using MAPLE, preprint **M94/4**, Exeter University, 1994.

[26] E.L. Mansfield, The differential algebra package `diffgrob2`, *MapleTech* **3** (1996) 33–37.

[27] E.L. Mansfield, Computer algebra for the classification problem, in: N.H. Ibragimov, K.R. Naqvi, E. Straume, Eds, *Modern Group Analysis VII*, Mars Publishers, Trondheim, 1999. pp211–217.

[28] M.B. Monagan *et al*, *Maple Programming Guide*, Springer Verlag, New York, 1996. MAPLE and MAPLE V are registered trademarks of Waterloo Maple Inc.

[29] P.J. Olver, *Classical Invariant Theory*, London Mathematical Society Student Texts **44**, Cambridge University Press, Cambridge, 1999.

[30] P.J. Olver, *Equivalence, Invariants and Symmetry*, Cambridge University Press, Cambridge, 1995.

[31] P.J. Olver, *Applications of Lie Groups to Differential Equations*, Second Edition, Springer Verlag, New York, 1993.

[32] P.J. Olver, private communication, 1998.

[33] L.V. Ovsiannikov, Group Analysis of Differential Equations, [Trans. W.F. Ames] Academic Press, New York, 1982.

[34] G.J. Reid, Algorithms for reducing a system of PDEs to standard form, determining the dimension of its solution space and calculating its Taylor series solution, *Euro. J. Appl. Math.* **2** (1991) 293–318.

[35] G.J. Reid, A.D. Wittkopf and A. Boulton, Reduction of systems of nonlinear partial differential equations to simplified involutive forms, *Eur. J. Appl. Math.* **7** (1996) 604–635.

[36] J.F. Ritt, *Differential Algebra*, Dover Pub., New York, 1950.

[37] C.J. Rust and G.J. Reid, Rankings of partial derivatives, in: W. Küchlin, *Proc. ISSAC '97* (ACM Press, New York, 1997) 9–16.

[38] J. Schü, W.M. Seiler and J. Calmet, Algorithmic Methods for Lie Pseudogroups, in Proc. Modern Group Analysis: Advanced Analytical and Computational Methods in Mathematical Physics, [Ed.] N. Ibragimov, M. Torrisi and A. Valenti, Kluwer, Dordrecht 1993.

[39] W.M. Seiler, Applying AXIOM to Partial Differential Equations, Preprint, Universität Karlsruhe, 1995.

[40] A.M. Tresse, Sur les invariants différentiels des groupes continue de transformation, *Acta Math.* **18** (1894) 1–88.

[41] Khai T. Vu, *Computer Algebra and Symmetry: the package Desolv*, PhD Thesis, Monash University, 1998.

[42] D. Zwillinger, *Handbook of Differential Equations*, Second Edition, Academic Press, Boston, 1992.