Determination of Maximal Symmetry Groups of Classes of Differential Equations

Gregory J. Reid
Department of Applied Mathematics
University of Western Ontario
London, Ontario, Canada N6A 5B7
Greg.Reid@uwo.ca

Allan D. Wittkopf
CECM, Simon Fraser University
Burnaby, B.C. Canada V5A 1S6
wittkopf@cecm.sfu.ca

ABSTRACT

A symmetry of a differential equation is a transformation which leaves invariant its family of solutions. As the functional form of a member of a class of differential equations changes, its symmetry group can also change.

We give an algorithm for determining the structure and dimension of the symmetry group(s) of maximal dimension for classes of partial differential equations. It is based on the application of differential elimination algorithms to the linearized equations for the unknown symmetries. Existence and Uniqueness theorems are applied to the output of these algorithms to give the dimension of the maximal symmetry group.

Classes of differential equations considered include ODE of form $u_{xx} = f(x, u, u_x)$, Reaction-Diffusion Systems of form $u_t - u_{xx} = f(u, v), v_t - v_{xx} = g(u, v)$, and Nonlinear Telegraph Systems of form $v_t = u_x, v_x = C(u, x)u_x + B(u, x)$.

1. INTRODUCTION

The symmetries, or transformations leaving invariant a system of partial differential equations, are generally not known and have to be determined. Infinitesimal Lie symmetries, the linearized form of such transformations, satisfy an associated linear system of overdetermined partial differential equations. There are many computer algebra programs for automatically generating such infinitesimal symmetry determining systems (see the extensive review article by Hereman [9], and also see [11]).

Applications of symmetries include mapping known solutions to new solutions, reducing the order of ODE and the number of variables in PDE, construction of similarity solutions or solutions invariant under a symmetry, and mapping of a given PDE to a more tractable PDE (e.g. transformation of a nonlinear PDE to a linear PDE) [1, 15].

Algorithms have been developed for simplifying overdetermined systems of PDE using differentiation and elimination. Maple has implementations of the Rosenfeld-Gröbner algorithm [3], Mansfield's Differential Gröbner Basis algorithm [13] and our *RifSimp* algorithm [20, 24] which is available at our web sites, and is distributed as part of Maple 6. Reduce has Wolf's program Crack [32], Schwarz's programs [26], and an implementation of the Cartan Exterior Differential Systems approach by Hartley and Tucker [8]. Axiom has an implementation of the Cartan-Kuranishi Algorithm [28].

In this article we apply our *RifSimp* algorithm written in Maple, and the *DiffElim* algorithm written in C by ADW, to infinitesimal symmetry determining systems of classes of PDE. These algorithms give highly structured output, and *RifSimp* delivers an existence and uniqueness theorem for that output [22, 24]. The algorithmic application of this theory gives, in principle, the sizes [17, 25] and structure [18, 19] of the symmetry groups of each member of a class of PDE without any heuristic integration of the associated systems of determining equations for the symmetries.

Although the algorithms are in theory finite, in practice their poor complexity and tendency to explode in memory put many interesting physical problems out of reach. In this article we give a method for partially solving such problems. In particular a method is given for determining the size and structure of maximal dimension symmetry groups. The ease of this method, compared to the full determination of the size and structure of all symmetry groups, is illustrated by examples.

The Existence and Uniqueness Theorem underpinning this article gives an algorithm for determining initial data to uniquely specify formal power series solutions. For a given point $x = x^0$ in the space of independent variables, the initial data consists of a set of (so-called *parametric*) derivatives of dependent variables that can be chosen freely at $x = x^0$. Ascribing values to the parametric derivatives (i.e. specifying the initial data), uniquely determines the values of the remaining (or so-called *principal*) derivatives at $x = x^0$. Hence the formal power series solution is uniquely deter-

[1] Support from the Natural Sciences and Engineering Research Council of Canada is gratefully acknowledged. The assistance of Mark Hickman, with his package *Symmetry* is also gratefully acknowledged.

ISSAC 2000, St. Andrews, Scotland
©2000 ACM 1-58113-218-2/ 00/ 0008 $5.00

mined. To apply this theorem, the system of PDE must be in a certain form, which in this article is chosen to be reduced involutive form (*rif*-form) [22].

The method for classification of maximal symmetry groups exploits a weakened form of the Existence and Uniqueness Theorem, applicable to systems not yet in *rif*-form. It gives upper bounds on the number of free parameters in formal power series solutions of the infinitesimal determining systems for symmetries, and hence on the dimensions of the symmetry groups. In the process of performing differential elimination on a given system, this upper bound decreases, as additional relations restricting the initial set of parametric derivatives are determined. Finally at output the set of initial data is given which uniquely specifies formal power series solutions.

Many aspects of the methods of this article have a long history. In 1894 Tresse [29] showed that by ordering derivatives, and taking integrability conditions, systems of PDE could be brought to a certain form, from which the number of parameters in their solutions could be determined. Tresse gave what can be regarded as the first proof of what is now called Dickson's Lemma, and the barest of outlines of Gröbner Basis Theory can be discerned in his work. Tresse's work was not rigorous, and Riquier and Janet put his work on a surer footing [21, 12]. Apart from linear systems, with rational function coefficients, however, these works are still not algorithmic by modern standards.

Tresse [29] gave a system of invariants which determined necessary and sufficient conditions for two second order ODE of form $u_{xx} = f(x, u, u_x)$ to be equivalent via an analytic change of variables. A by-product of Tresse's classification, is that the maximal dimension of the symmetry group of such ODE is 8, and that this value is obtained if and only if the ODE is equivalent by change of variables to $u_{xx} = 0$ (see [27] for an alternative approach).

The descendant of Tresse's method is Cartan's powerful geometric method for determining when two differential systems are equivalent [16]. The classification of second order ODE using this method was carried out by Kamran and Hsu [10].

A number of examples of the determination of maximal symmetry groups of geometric objects have been carried out by Differential Geometers. An early example, closely related to our method, is Eisenhart's demonstration [6] that the maximal dimension of an isometry group of an n-dimensional Riemannian space is $n(n+1)/2$, and that this maximum only occurs in spaces of constant curvature.

Gröbner basis and differential elimination methods can, by finite step algorithms, determine information such as dimension of solution components. However, the complexity of these calculations can make their completion practically infeasible. For even moderately sized systems, the process of differential elimination can cause the equations of the system to grow exponentially during calculation, exhausting the resources of the machine, and causing the calculation to fail.

One advantage of the approach proposed in this paper is that it can allow a partial form of the calculation to succeed. It can also deliver information about the progress of the calculation, as it is running, to guide later attempts. For some problems, only the form of the class having the maximal symmetry group is needed. For example in searching for linearizations of PDE, only infinite symmetry groups are of interest. Our method can narrow the search space for such systems.

In Section 2, we give an informal introduction, by example, to concepts and methods of differential elimination algorithms. We also present the *MaxDim* algorithm, and briefly discuss the *RifSimp* and *DiffElim* algorithms. In Section 3, the *MaxDim* algorithm is applied to the determination of maximal symmetry group of second order ODE, reproducing a well-known classical result [29, 10, 27]. In Section 4, the *MaxDim* algorithm is applied to Nonlinear Reaction Diffusion Systems, and in Section 5 it is applied to a Nonlinear Telegraph System.

2. DIFFERENTIAL ELIMINATION, RIFSIMP, AND MAXDIM

An informal introduction, by example, is given for differential elimination algorithms (see [3, 22, 24] for formal treatments).

2.1 Rankings and Leading Derivatives

Differential elimination algorithms can be regarded as a generalization of the Gröbner Basis Algorithm, which in turn can be regarded as a generalization of the Gauss Algorithm. The *RifSimp* algorithm takes as input a polynomially nonlinear system of PDE, with coefficients which are polynomial functions in the independent variables, over the Rational Numbers (or some computable extension thereof). For effective completion it also requires a ranking of derivatives, which is a total ordering whose most important property is invariance under differentiation. We also assume that the ranking is a *positive ranking*, that is, it is a well-ordering. The classification of all rankings is given in [22, 23] (also see [5]).

For a system consisting of $u(x)$ and its derivatives, if $\prec$ is a positive ranking, then $u \prec u_x$ by positivity, and by invariance under differentiation only one positive ranking is possible, namely $u \prec u_x \prec u_{xx} \prec u_{xxx} \prec u_{xxxx} \prec \ldots$. Two (of many) possible rankings for one dependent variable η, and two independent variables, x and u are

$$\eta \prec \eta_u \prec \eta_x \prec \eta_{uu} \prec \eta_{xu} \prec \eta_{xx} \prec \ldots \tag{1}$$

$$\eta \prec \eta_u \prec \eta_{uu} \prec \ldots \prec \eta_x \prec \eta_{xu} \prec \eta_{xuu} \prec \ldots. \tag{2}$$

In any equation, the derivative or dependent variable highest in the ranking is called the *leading derivative* of that equation. An equation is *leading linear*, if it is linear in its highest derivative, otherwise it is *leading nonlinear*. For a given system (not necessarily in *rif*-form), the set of leading derivatives of all equations, and their derivatives are called the *principal derivatives* of the system. The complementary set of derivatives are called the *parametric derivatives* of the system.

2.2 Elimination

Differential elimination is illustrated by application to the simple system

$$\eta_{xuu} - \eta_{xxxu} = 0, \quad \eta_{xxu} - \eta = 0.$$

Assuming ranking (1), this system would be rewritten in the solved form with respect to its leading derivatives as

$$\eta_{xxxu} = \eta_{xuu}, \quad \eta_{xxu} = \eta,$$

since $\eta_{xuu} \prec \eta_{xxxu}$ and $\eta \prec \eta_{xxu}$. The first equation differentially reduces with respect to the second as

$$\eta_{xxxu} = (\eta_{xxu})_x = (\eta)_x = \eta_x.$$

The reduced equation is then solved for its leading derivative, η_{xuu}, and the system becomes

$$\eta_{xuu} = \eta_x, \quad \eta_{xxu} = \eta. \tag{3}$$

No further differential reductions are possible (for definitions of reduction, etc., see [22, 3]).

2.3 Initial Data

We determine the initial data required to build a formal power series solution for η at a point (x^0, u^0). The set of principal derivatives is the set of all derivatives of the leading derivative η_{xuu} of the first equation together with the set of all derivatives of the leading derivative η_{xxu} of the second equation, or

$$\begin{aligned} &\{\eta_{xuu}, \eta_{xxuu}, \eta_{xuuu}, \eta_{xxxuu}, \eta_{xxuuu}, \ldots\} \\ \cup\ &\{\eta_{xxu}, \eta_{xxxu}, \eta_{xxuu}, \eta_{xxxxu}, \eta_{xxxuu}, \ldots\}. \end{aligned} \tag{4}$$

The parametric derivatives, which are the derivatives not belonging to the above set of principal derivatives, are

$$\{\eta(x^0,u^0), \eta_x(x^0,u^0), \eta_{xx}(x^0,u^0), \ldots\} \cup \\ \{\eta_u(x^0,u^0), \eta_{uu}(x^0,u^0), \ldots\} \cup \{\eta_{xu}(x^0,u^0)\}.$$

An equivalent finite representation for the above set of parametric derivatives is given by

$$\eta(x,u^0) = f_1(x), \quad \eta_u(x^0,u) = f_2(u), \quad \eta_{xu}(x^0,u^0) = c_1.$$

This is called the *initial data* of a system. In this case we have two arbitrary functions of one variable, and a constant. Algorithms for describing initial data go back to [12, 21]. Efficient versions are given in [17] (also see [26]), and versions in the presence of constraints are described in [20, 22].

2.4 Integrability Conditions

The condition for equality of mixed partials from (3):

$$(\eta_{xuu})_x - (\eta_{xxu})_u = (\eta_x)_x - (\eta)_u = \eta_{xx} - \eta_u = 0 \tag{5}$$

is called an integrability condition for (3). It is new in that it does not reduce to zero with respect to (3).

It should be noted that for n equations, solved for derivatives of the same dependent variable, the number of conditions to be checked by equality of mixed partials is $O(n^2)$. More elaborate and efficient algorithms for avoiding redundant integrability conditions are discussed in [17, 22, 2].

2.5 Completing the Example

Solving the new PDE resulting from (5) for its leading derivative yields $\eta_{xx} = \eta_u$ which is appended to the system. The first equation of (3) is reduced to $\eta_{uu} - \eta = 0$ by this equation, which in turn is solved for η_{uu} and reduces the second equation of (3) to zero. Thus the system becomes

$$\eta_{xx} = \eta_u, \quad \eta_{uu} = \eta. \tag{6}$$

It can be checked that all the integrability conditions of this system are identically satisfied, and that the system satisfies the conditions for *rif*-form given in [22].

The parametric derivatives, (all of which are not derivatives of η_{xx} or η_{uu} from (6)) are $\eta, \eta_x, \eta_u, \eta_{xu}$ so the initial data has dropped to 4 arbitrary constants:

$$\eta(x^0,u^0) = c_1, \quad \eta_x(x^0,u^0) = c_2, \\ \eta_u(x^0,u^0) = c_3, \quad \eta_{xu}(x^0,u^0) = c_4.$$

New equations appended to the system give constraints on the parametric derivatives and reduce the initial data. This is the foundation on which the *MaxDim* algorithm is based.

2.6 Pivots and Binary Branching

In the above example, all coefficients are constant. In Lie symmetry classification problems, the coefficients of the system can contain unknown parameters or functions, which we call *class functions.* Algorithms such as our *RifSimp* algorithm and the Rosenfeld-Gröbner algorithm [3] execute binary branching on such coefficients.

For example, consider a calculation where the PDE $cu_{xxx} - u_x + u^2 = 0$ is encountered. The calculation will branch into two cases, specifically the case $c \neq 0$, and hence $u_{xxx} = (u_x - u^2)/c$, and the case $c = 0$. The coefficient c is a *Pivot*, and the $c \neq 0$ constraint is called an *inequation.*

2.7 The RifSimp and MaxDim Algorithms

The following is a description of the most relevant steps in the *RifSimp* algorithm with case-splitting options. It is to be employed as a sub-routine in the *MaxDim* algorithm described later in this section. A set-up call (RifSimp → CalcRifSimp) is needed due to the recursive nature of the algorithm.

Input: A system of ODE/PDE(Sys), a ranking, and an initial data constraint (minfree)
Output: A tree structure containing any completed cases, and indicators for the other cases where the initial data constraint was violated

```
RifSimp(Sys, minfree)
   UnC :=Sys (unclassified equations)
   Sol  :=∅   (solved form equations)
   NLin :=∅   (equations with a nontrivial pivot)
   Piv  :=∅   (all pivots currently in effect)
   Case :=[]  (current branch on binary case tree)
   return CalcRifSimp (UnC, Sol, Nlin, Piv, Case, minfree)
end Rifsimp

CalcRifSimp(Sol, UnC, NLin, Piv, Case, minfree)
   while UnC ≠ ∅ and Nlin ≠ ∅ do
      neq:= select eqn from UnC that can be solved without
```

```
      introducing a new pivot (any eqn that require new
      pivots are placed in NLin)
    if neq is null then
      neq := select eqn from NLin
      piv := pivot for neq
      return CalcRifSimp(Sol, UnC ∪ {neq}, NLin,
              Piv ∪ {piv ≠ 0}, [Case,piv ≠ 0], minfree)
              ∪
              CalcRifSimp(Sol, UnC ∪ {neq, piv=0},
              NLin, [Case, piv = 0], minfree)
    end if
    neq := solve neq for highest ranked derivative
    Sys := self reduce Sys ∪ {neq}
    if initial data constraint on Sys is violated then
      return [Case, 'minfree violated', details]
    end if
    Piv := reduce Piv wrt Sys
    if pivots are violated then
      return [Case, 'system is inconsistent']
    end if
    UnC :=UnC ∪ {new integrability conditions on Sys}
    UnC := reduce UnC wrt Sys
  end loop;
  return ([Case, Sys, Piv])
end CalcRifSimp
```

In the applications considered in this article, the infinitesimal determining systems are linear and homogeneous in the infinitesimals. However the coefficients of these systems are generally polynomially nonlinear in derivatives of the functions occurring in the class of PDE being considered. A ranking is chosen, that orders all of the infinitesimals, and their derivatives, greater than any derivative of any class function. As splitting occurs, this generally results in a separate system of nonlinear differential equations and inequations for the class functions. For sake of brevity in the above algorithm, we have not included the subroutines in which these systems are also brought to *rif*-form, as this uses the fully nonlinear theory given in [22, 24]. For an example of such a fully nonlinear class system see (22).

The main algorithm of this article uses the modified *RifSimp* algorithm presented above, or another modified differential elimination algorithm (such as *DiffElim*) to detect maximal groups for the problem.

The MaxDim Algorithm:

1 Set the maximal dimension d to ∞.

2 The differential elimination algorithm traverses the binary tree, stopping on any branch k when the upper bound for the initial data d_k drops below d. After a finite number of steps, either some branches complete, and output *rif*-forms will have been found, or none will have been found.

3 If branches have completed, the algorithm terminates, returning the *rif*-forms with d free parameters in their formal series solutions, and a list of all failed branches with the values $d_k < d$, representing the upper bounds on the dimension of initial data for the branch.

4 If no branches have completed, the maximal dimension d is set to the maximal value found on the rejected branches, and the algorithm restarts at step 2.

The algorithms given in [19] are then applied to the output *rif*-forms to determine the structure of the maximal symmetry groups.

In addition to finding the structure of the maximal symmetry groups for a class of PDE, the *MaxDim* algorithm can also be used to find the symmetry group belonging to all members of a class of PDE. Consider application of the *MaxDim* algorithm, where the initial data restriction is placed on the classification functions instead of the infinitesimals. For the case where the classification functions are fully arbitrary (i.e. no restrictions are placed on their form), the algorithm will follow a single branch consisting only of the inequations $P_1 \neq 0, P_2 \neq 0, ..., P_k \neq 0$. Since the input system is homogeneous and linear in the infinitesimals, and the infinitesimals are of higher rank than the class functions, these inequations will only depend on the class functions. This means the selection of a $P_i = 0$ branch would decrease the initial data for the class functions, and hence be rejected. This will compute the *rif*-form of the defining system for the *minimal* dimensional symmetry group. Non-minimal dimensional symmetry groups require that $P_1 P_2 ... P_k = 0$.

2.8 The DiffElim Algorithm

The primary goal of this paper is give a strategy to obtain information about difficult systems, by exploring some particularly interesting (in this case maximal dimensional) parts of their solution spaces. Another strategy we are pursuing is the development of a stand alone implemenation of a flexible differential elimination algorithm in C, the *DiffElim* algorithm. Already, Faugére has pursued such a strategy in algebraic elimination with his development of stand alone Groebner Bases algorithms in C. He has shown that large gains in efficiency and speed can be made over traditional general-purpose computer algebra systems [7].

Advantages of such low level implementations include much tighter control over memory allocation, and the implementation of operations such as differentiation, GCDs, etc., that are specifically tailored to the given application. Despite *DiffElim* being a new and still developing program, the results in Section 4.3 show that it can give great gains over the very mature and well-developed implemenation of *RifSimp* in Maple (for a more detailed discussion see [31]). Of course, disadvantages, when compared to general purpose computer algebra systems, include the lack of ability to exploit updates of the quality and speed of algorithms made in these systems.

While possessing much of the functionality of *RifSimp*, the *DiffElim* algorithm also incorporates algorithmic approaches not available in *RifSimp*. Specifically, it incorporates a pure reduction mode (introducing no new pivots during reduction, and hence obtains all results in a single case). It is generally non-terminating on nonlinear systems in this mode (unlike the Groebner basis case [4]), but can, after a finite number of iterations, uncover simple relations which allows termination in its case splitting mode, or by the use of *RifSimp*, with fewer cases.

Finally, for systems in which no case splitting occurs (e.g. for linear systems), the algorithm yields equivalent output to *RifSimp*, and its predecessor *StandardForm*, as expected.

3. MAXIMAL GROUP OF $\mathbf{U_{XX} = F(X, U, U_X)}$

The *MaxDim* algorithm is illustrated by determining the conditions for the class of second order ODE of form

$$u_{xx} = f(x, u, u_x) \tag{7}$$

to have the maximal Lie symmetry group. This reproduces a classical result first derived by Tresse [29], and then others (e.g. see [10, 27]).

The symmetries of (7), are transformations of the form

$$(x, u) \mapsto (\hat{x}, \hat{u}) = (X(x, u), U(x, u)) \tag{8}$$

which leave its form invariant. The direct nonlinear approach of substituting the transformations was linearized in breakthrough work by Lie. Following this approach we seek linearized transformations about the identity or equivalently infinitesimal symmetries of the form

$$(x, u) \mapsto (\hat{x}, \hat{u}) = (x + \xi\epsilon + \mathcal{O}(\epsilon^2), u + \eta\epsilon + \mathcal{O}(\epsilon^2)) \tag{9}$$

where ξ, η, are unknown functions of x, u. Lie's method [15] yields the associated infinitesimal determining equation which is a linear, homogeneous PDE for ξ, η

$$\begin{aligned} &\eta_{xx} - 2f\xi_x + f\eta_u - f_u\eta - f_x\xi - f_{u_x}\eta_x \\ &+u_x(2\eta_{xu} - \xi_{xx} - 3f\xi_u - f_{u_x}\eta_u + f_{u_x}\xi_x) \\ &+u_x^2(\eta_{uu} - 2\xi_{xu} + f_{u_x}\xi_u) - u_x^3\xi_{uu} = 0, \end{aligned} \tag{10}$$

together with the equations that are *implied* by the fact that both ξ and η are independent of u_x

$$\xi_{u_x} = 0, \quad \eta_{u_x} = 0. \tag{11}$$

The key input to the differential elimination algorithm *RifSimp* is the determining equation (10), and a ranking of derivatives which is chosen to be

$$\xi \prec \eta \prec \xi_u \prec \eta_u \prec \xi_x \prec \eta_x \prec \xi_{uu} \prec \eta_{uu} \prec ... \tag{12}$$

Additionally all derivatives of the class function $f(x, u, u_x)$ are ranked lower than any derivative of ξ or η.

The *MaxDim* algorithm is executed using *RifSimp* by setting the infolevel (`infolevel[rifsimp]:=3`), to allow viewing of the state of the system as it is processed, then issuing the command

```
rifsimp(sys,[eta,xi],indep=[x,u],casesplit,
          minfree=infinity);
```

where `sys` is a list containing equation (10), and the inclusion of $[\eta, \xi]$ and $[x, u]$ in the argument list implies the derivative ranking is the one described in (12).

Initially the system consists of only equation (10) in the unclassified (yet to be processed) list UnC. The algorithm reports:

```
#OT: 0 #Sol:  0 #NL: 0 #UnC: 1 #NonP: 0 #Piv:  0
```

The leading derivative of (10) in the ranking (12) is η_{xx}. This equation is written in solved form as $\eta_{xx} = F_1$ (where the explicit form of F_1 easily follows from (10), and the algorithm reports:

```
#OT: 0 #Sol:  1 #NL: 0 #UnC: 0 #NonP: 0 #Piv:  0
```

The algorithm then calculates the following initial data

$$\eta(x^0, u) = h_1(u), \; \eta_x(x^0, u) = h_2(u), \; \xi(x, u) = h_3(x, u),$$

and the data is infinite, so the algorithm continues.

It is found that the new solved equation $\eta_{xx} = F_1$ has a non-trivial integrability condition with the implied equation $\eta_{u_x} = 0$, giving

$$0 = (\eta_{xx})_{u_x} - (\eta_{u_x})_{xx} = (F_1)_{u_x} = \xi_{xx} - F_2, \tag{13}$$

where F_2 is a function of derivatives of lower rank than ξ_{xx}. So at the end of the first iteration the algorithm reports:

```
#OT: 0 #Sol:  1 #NL: 0 #UnC: 1 #NonP: 0 #Piv:  0
```

In the next iteration (13) is placed in solved form with respect to its leading derivative ξ_{xx}, and the initial data is calculated giving

$$\begin{aligned} \eta(x^0, u) &= h_1(u), \quad \eta_x(x^0, u) = h_2(u), \\ \xi(x^0, u) &= h_3(u), \quad \xi_x(x^0, u) = h_4(u), \end{aligned} \tag{14}$$

so the initial data currently depends on four arbitrary functions of one variable. The complementary sets of parametric and principal derivatives of η and ξ are represented by the "dotplots" below. Each solid circle represents a principal derivative, and each empty circle represents a parametric derivative.

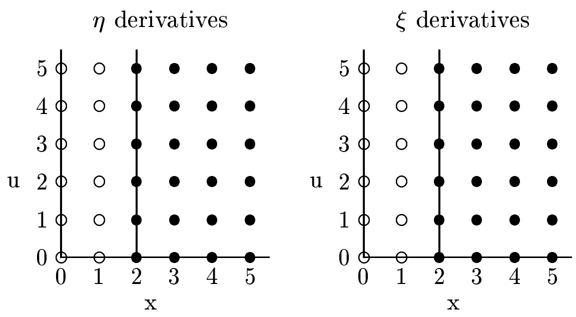

Figure 1: Derivative Dotplot

In the diagram on the left, the solid circle located at (3, 1) corresponds to the derivative η_{xxxu}, which is a principal derivative since it is a derivative of the leading derivative η_{xx}. In the diagram on the right the empty circle located at (1,1) corresponds to the derivative ξ_{xu}, which is a parametric derivative since it is not a derivative of either of the leading derivatives η_{xx}, ξ_{xx}. The infinite set of parametric derivatives of η can be split into two infinite subsets, which lie on the two vertical lines $x = 0, x = 1$. These are represented by the functions $h_1(u), h_2(u)$ in (14). A similar interpretation holds for the parametric derivatives of ξ.

The new solved equation has a non-trivial integrability condition with the implied equation $\xi_{u_x} = 0$, giving

$$0 = (\xi_{xx})_{u_x} - (\xi_{u_x})_{xx} = (F_2)_{u_x} = 2\xi_{xu} - 2F_3 \tag{15}$$

where F_3 is a function of derivatives of lower rank than ξ_{xu}. So at the end of the second iteration we have:

```
#OT: 0 #Sol:  2 #NL: 0 #UnC: 1 #NonP: 0 #Piv:  0
```

By the same process, the next iteration places the unsolved equation (15) with leading derivative ξ_{xu} into solved form, and the initial data drops to

$$\begin{aligned} \eta(x^0, u) &= h_1(u), \quad \eta_x(x^0, u) = h_2(u), \\ \xi(x^0, u) &= h_3(u), \quad \xi_x(x^0, u^0) = h_4. \end{aligned}$$

This does not restrict the dotplot for η, but restricts the dotplot for ξ, by introducing a vertical line of principal derivatives located at $(1,1),(1,2),(1,3),(1,4),\ldots$ into that diagram.

The new equation has an integrability condition with the implied equation $\xi_{u_x}=0$ giving

$$0=(\xi_{xu})_{u_x}-(\xi_{u_x})_{xu}=(F_3)_{u_x}. \tag{16}$$

It also has an integrability condition with the second solved form equation resulting from (13) giving

$$0=(\xi_{xu})_x-(\xi_{xx})_u=(F_2)_x-(F_1)_u. \tag{17}$$

At the end of the third iteration, the system state is:

```
#OT: 0 #Sol:  3 #NL: 0 #UnC: 2 #NonP: 0 #Piv:  0
```

Once these equations have been reduced with respect to the system, and each other, they become

$$\eta_{uuu}=F_4, \quad \xi_{uu}=F_5, \tag{18}$$

so our initial data is now given by

$$\begin{aligned}
\eta(x^0,u^0)&=h_1, & \eta_u(x^0,u^0)&=h_2, & \eta_{uu}(x^0,u^0)&=h_3,\\
\eta_x(x^0,u^0)&=h_4, & \eta_{xu}(x^0,u^0)&=h_5, & \eta_{xuu}(x^0,u^0)&=h_6,\\
\xi(x^0,u^0)&=h_7, & \xi_u(x^0,u^0)&=h_8, & \xi_x(x^0,u^0)&=h_9.
\end{aligned}$$

This initial data is represented in the following dotplot:

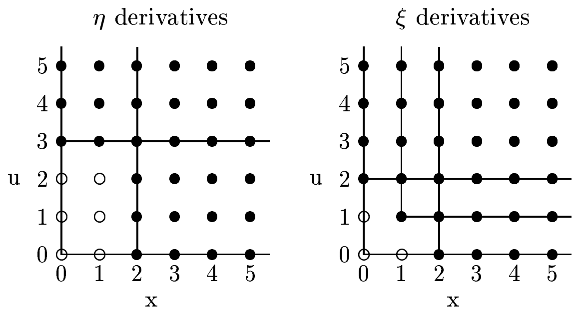

Figure 2: Derivative Dotplot

Since minfree was set to ∞, *RifSimp* terminates, as the initial data space has closed off to at most 9 parameters (the number of empty circles in the diagram above).

Following the *MaxDim* algorithm, the calculation is restarted with minfree=9. An additional relation of the form $\eta_{xuu}=F_6$ is found, so the the initial data space is at most dimension 8, and the algorithm again terminates.

Running the calculation with minfree=8 rapidly arrives at an 8-dimensional initial data space, but all of the remaining unsolved equations (arising from new integrability conditions) have leading derivatives with coefficients that are differential expressions in $f(x,u,u_x)$. *RifSimp* is then forced to choose an equation (from the 6 equations in the nonlinear (NL) list), and perform a binary case split on the coefficient of the leading derivative (the pivot). The chosen equation is of the form

$$\eta_x f_{u_x u_x u_x u_x}+F_7=0. \tag{19}$$

The case $f_{u_x u_x u_x u_x}\neq 0$ immediately results in a solved equation for η_x, which in turn drops the dimension of the initial data to 5, so this case is rejected. The calculation proceeds with $f_{u_x u_x u_x u_x}=0$, and this pattern repeats for three more pivots, and the calculation completes with a single 8-parameter case.

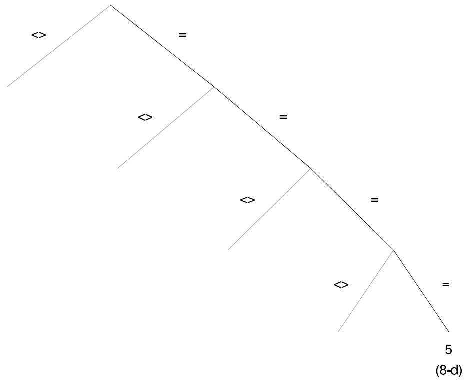

Figure 3: Second order ODE

The `caseplot` command can be used to view the binary case split structure, and results in Figure 3 where all the <> branches having no numbers attached are cases that have less than 8 free parameters, and case 5 (lower right of the figure) has exactly 8 parameters. There are two equations in case 5 containing only $f(x,u,u_x)$ and its derivatives:

$$\begin{aligned}
f_{u_x u_x u_x u_x} &= 0\\
f_{xxu_x u_x} &= 4f_{xuu_x}-u_x f_{u_x u_x u_x} f_u-4f_{u_x} f_{uu_x}\\
&\quad +3f_{uu_x u_x} f-f_{u_x u_x u_x} f_x-2u_x f_{xuu_x u_x}\\
&\quad -u_x^2 f_{uuu_x u_x}-2f_{xu_x u_x u_x} f+3f_{u_x u_x} f_u\\
&\quad +u_x f_{u_x} f_{uu_x u_x}-2u_x f_{uu_x u_x u_x} f\\
&\quad +f_{u_x} f_{xu_x u_x}-6f_{uu}+4u_x f_{uuu_x}.
\end{aligned}$$

These are exactly the conditions on f required for equation (7) to have a maximal symmetry group. The structure of the 8-dimensional Lie algebra can be determined from the *rif*-form of the determining equations and the class equations given above using [19].

4. MAXIMAL GROUP OF NONLINEAR REACTION DIFFUSION EQUATIONS

We consider coupled nonlinear reaction diffusion systems [14] of form

$$\mathbf{u}_t-\mathbf{u}_{xx}=\mathbf{f}(\mathbf{u}), \tag{20}$$

where $\mathbf{u}$ is an m-component vector. The components of $\mathbf{u}$ are, for example, concentrations of interacting chemicals in combustion or competing species in biological applications [14]. The group theoretic study of such systems was initiated in [33] (also see Vol. I, §10.13 of [11]).

4.1 Two Component Uncoupled System

We now consider the two component reaction-diffusion system where $\mathbf{u}=(u,v)$ which is given by

$$u_t-u_{xx}=f(u), \qquad v_t-v_{xx}=g(v)$$

where the nonlinearity is imposed by the additional inequation $f_{uu}\neq 0$.

Running *RifSimp* on this system, together with the inequation, results in 8 infinite dimensional cases as shown in Figure 4. Analysis of each of the infinite branches reveals that the equation $g_{vv} = 0$ is common to all. These are all cases in which the second equation becomes linear with respect to v, and admits an easily interpreted infinite dimensional group, which is not pursued further here.

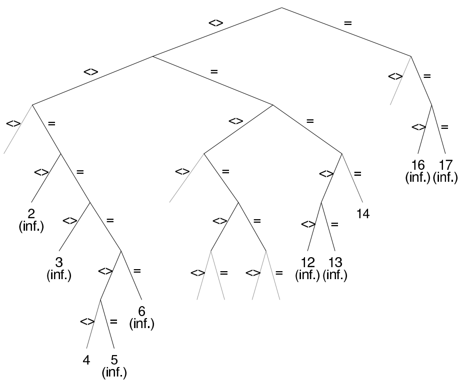

Figure 4: Uncoupled Reaction-Diffusion (infinite)

If we re-run the calculation adding the assumption $g_{vv} \neq 0$, we find that there are no infinite cases, and obtain an upper bound for the dimension of the initial data of 5.

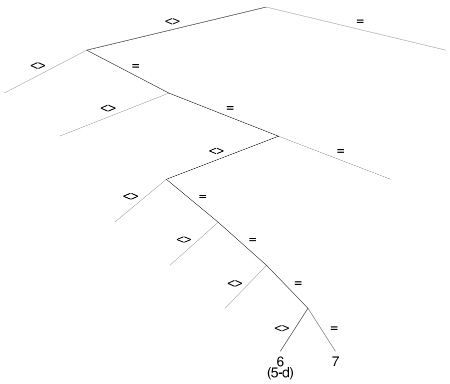

Figure 5: Uncoupled Reaction-Diffusion (finite)

Running *RifSimp* with `minfree=5` gives us exactly one case, as shown by the `caseplot` in Figure 5. We also obtain a system of 3 PDE, and 4 inequations for $f(u), g(v)$:

$$
\begin{aligned}
f_{uuu} = \frac{-f_{uu}g_{vv}g - f_{uu}f_u^2 + f_{uu}^2 f + f_u g_v f_{uu}}{f f_u - f g_v},& \\
g_{vvv} = \frac{g_{vv}^2 g_v - f_u g_{vv}^2}{f f_{uu} - g_{vv} g},& \\
-2 f f_{uu} g_{vv} g + g^2 g_{vv}^2 - f_u g_v f f_{uu} - g_{vv} f_u g_v g& \\
+ g_v^2 f f_{uu} + f^2 f_{uu}^2 + f_u^2 g_{vv} g = 0,& \\
f_{uu} \neq 0, g_{vv} \neq 0, f f_{uu} - g_{vv} g \neq 0.& \\
f_u^2 - f_u g_v - f f_{uu} + g_{vv} g \neq 0,&
\end{aligned}
$$

which can be solved using Maple's *dsolve* with assistance from *RifSimp* to obtain

$$
\begin{aligned}
f(u) &= a(u + b_1)\ln(c_1(u + b_1)), \\
g(v) &= a(v + b_2)\ln(c_2(v + b_2)).
\end{aligned}
$$

This is an predictable extension of the well-known result given in §10.2, Vol. I of [11], that the one component diffusion equation with (4 dimensional) maximal group is $u_t - u_{xx} = a(u + b_1)\ln(c_1(u + b_1))$.

4.2 Two Component Partially Coupled System

We now consider the two component reaction-diffusion system given by

$$u_t - u_{xx} = f(u), \qquad v_t - v_{xx} = g(u, v),$$

where the nonlinearity is imposed by the additional inequation $f_{uu} \neq 0$, and the coupling by the equation $g_u \neq 0$.

RifSimp gives a total of 67 cases for this problem, of which 18 are rejected as the initial data is finite, and 41 have infinite initial data. Investigation reveals that all infinite cases have two branching equations in common, specifically $g_{vv} = 0$ and $g_{uv} = 0$.

This means that $g(u, v)$ is of the form $g(u, v) = g_1 v + g_2(u)$, and with this restriction, one of our equations becomes linear in v. Once the nonlinear equation for u alone is solved, we have an inhomogeneous linear equation in v to solve. This is a predictable class with infinite groups, which we do not pursue further here.

To find pure nonlinear cases, we then examine the three separate cases given by Case (I): $g_{uv} \neq 0, g_{vv} = 0$; Case (II): $g_{uv} = 0, g_{vv} \neq 0$; Case (III): $g_{uv} \neq 0, g_{vv} \neq 0$.

Application of the *MaxDim* algorithm to Case (I) shows that there are no solutions with ∞, or 7 dimensional initial data, but there are three solutions having 6 dimensional initial data (see Figure 6). The first case (referenced by 16 in

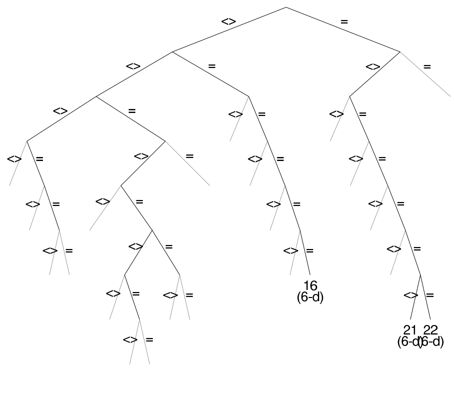

Figure 6: Partially Coupled Reaction-Diffusion

Figure 6) gives a straightforward set of differential equations for f, g, which can be readily solved to give

$$
\begin{aligned}
f(u) &= a(u + b)^2, \\
g(u, v) &= auv + d_1 u + d_2 v + c,
\end{aligned}
\tag{21}
$$

where a, b, c, d_1, and d_2 are arbitrary parameters.

The second case (21 in Figure 6) gives a somewhat more complex system of differential equations, involving four PDE, and a constraint

$$
\begin{aligned}
g_{uuu} &= \frac{2g_v g_{uu}^2 - 2f_u g_{uu}^2 - 2g_u g_{uv} g_{uu}}{f g_{uu} + g_{uv} g}, \\
g_{uuv} &= \frac{g_{uv} g_{uu} g_v - g_{uv}^2 g_u - g_{uv} g_{uu} f_u}{f g_{uu} + g_{uv} g}, \\
& g_{vv} = 0, f_{uu} = g_{uv}, \qquad (22) \\
-2g_v g_u g_{uv} f g_{uu} &+ g_v^2 g_{uu}^2 f + f^2 g_{uv} g_{uu}^2 + 2 f g_{uv}^2 g_{uu} g \\
&- f_u g_{uu}^2 g_v f + f_u g_{uu} g_v g_{uv} g + g_{uv}^3 g^2 + g_u^2 g_{uv}^2 f \\
&- g_u g_{uv}^2 f_u g + g_{uv} g_{uu} f_u f g_u - g_{uv} g_{uu} f_u^2 g = 0.
\end{aligned}
$$

Since $f(u), g(u,v)$ and all their derivatives are ranked lower than the infinitesimals, the *RifSimp* algorithm gives the f, g subsystem in *rif*-form. This also means that no new conditions arise through differentiation of the constraint (See [24]). These can also be explicitly solved to give

$$
\begin{aligned}
f(u) &= a_1(u+b_1)\ln(c_1(u+b_1)), \\
g(u,v) &= a_1(v+b_2)\ln(c_2(u+b_1)) + a_2(u+b_1),
\end{aligned}
$$

where a_1, a_2, b_1, b_2, c_1, and c_2 are arbitrary parameters.

The third case can be shown to be a subcase of the second, so we have two separate forms for $f(u), g(u,v)$ that give 6 dimensional initial data.

Exploration of Case (II) revealed that the initial data was at most 4 dimensional. Case (III), the most computationally challenging of the three, required the use of *DiffElim* to complete, and was shown to be at most 5 dimensional. So we have found the maximal finite cases.

4.3 Extension to Greater Number of Components

We explore the extension of one of the maximal cases for the partially coupled system (21) to a system involving a greater number of components.

Extensive calculations in the three component case, made possible by the *DiffElim* implementation, and based on the form of the two component system, led us to conjecture the following form for $\mathbf{f} = (f_1, ..., f_n)$:

$$
\begin{aligned}
f_1(u_1) &= a(u_1+b)^2, \quad a \neq 0, \\
f_i(u_1,...,u_n) &= a u_1 u_i + \sum_{j=1}^{n} d_{i,j} u_j + c_i, \quad i = 2,...,n
\end{aligned}
$$

For $n = 1, 2, ..., 7$ it was found that the dimension of the Lie group of symmetries of the n-component system is $n^2 + 2$. These calculations, which resulted in one case, were used as a test problem to compare the efficiency of the *RifSimp* implementation and the newly developed *DiffElim* implementation [31]. In the following table, memory is measured in Mbytes, and time in c.p.u. seconds running on a PII 333MHz Linux machine:

		RifSimp		DiffElim	
n	dim	Time	Mem	Time	Mem
1	3	2.41	2.56	–	0.69
2	6	4.63	2.69	0.01	0.71
3	11	9.55	3.00	0.04	0.74
4	18	19.93	3.12	0.09	0.81
5	27	42.72	3.56	0.26	0.92
6	38	91.28	4.12	0.53	1.03
7	51	202.99	4.50	0.81	1.43

Table 1: Run-time Comparison

In the above table, the *DiffElim* implementation has a running time between $\frac{1}{164}$ and $\frac{1}{250}$ of that of *RifSimp*, and requires less than $\frac{1}{3}$ the memory for these calculations.

5. NONLINEAR TELEGRAPH SYSTEM

Consider the nonlinear telegraph system

$$v_x = u_t, v_t = C(u,x)u_x + B(u,x),$$

where $C_u \neq 0$ and $B_u \neq 0$. Application of *MaxDim* with $d = \infty$, yields several infinite cases. Integrating the classifying conditions for one of these cases gives the class of nonlinear telegraph equations

$$v_x = u_t, v_t = \frac{p}{u} f(pu)\, u_x + p_x f(pu),$$

where p is an arbitrary function of x and f is an arbitrary function of its argument pu.

The type of infinite symmetry group admitted satisfies the conditions for Bluman and Kumei's Linearization Theorem [1]. That theorem shows that

$$X = \int \frac{dx}{p(x)}, T = t, U = p(x)u, V = v$$

transforms the system above to

$$V_X = U_T, V_T = f(U)\, U_X / U,$$

which is well-known to be linearizable by hodograph transformation. When $f(pu) = b/(pu)$ and $p(x) = \exp(ax)$, a, b constant, the above system is $v_x = u_t$, $v_t = bu_x/u^2 + ab/u$ and is linearizable as was shown in [30].

6. REFERENCES

[1] G. W. Bluman and S. Kumei, *Symmetries and Differential equations*, vol. 81 of Applied Mathematical Sciences, Springer, 1989.

[2] F. Boulier, *Some improvements of a lemma of Rosenfeld.* Preprint, 1998.

[3] F. Boulier, D. Lazard, F. Ollivier, and M. Petitot, *Representation for the radical of a finitely generated differential ideal*, in Proc. ISSAC 1995, ACM Press, 1995, pp. 158–166.

[4] G. Carrà Ferro, *Gröbner Bases and differential ideals*, in Proc. AAECC5, Menorca, Spain, 1997, pp. 129–140.

[5] G. CARRÀ FERRO AND W. SIT, *On term-orderings and rankings*, in Computational Algebra, vol. 151 of Lecture Notes in Pure and Applied Mathematics, Marcel Dekker, 1994, pp. 31–77.

[6] L. P. EISENHART, *Riemannian Geometry*, Princeton University Press, 1949.

[7] J. C. FAUGÈRE, *A new efficient algorithm for computing Gröbner bases (F4)*, in J. of Pure and Appl. Algebra, 139, 1999, pp. 61–88.

[8] D. HARTLEY AND R. TUCKER, *A Constructive Implementation of the Cartan–Kähler theory of Exterior Differential Systems*, J. Symb. Comp., 12 (1991), pp. 655–667.

[9] W. HEREMAN, *Review of symbolic software for the computation of Lie symmetries of differential equations*, Euromath Bull., 1 (1994), pp. 45–79.

[10] L. HSU AND N. KAMRAN, *Classification of second-order ordinary differential equations admitting Lie groups of fibre-preserving symmetries*, Proc. Lond. Math. Soc., 58 (1989), pp. 387–416.

[11] N. H. IBRAGIMOV, ed., *CRC Handbook of Lie Group Analysis of Differential Equations*, vol. I,II,III, CRC Press, Boca Raton, 1994,1995,1996.

[12] M. JANET, *Sur les systèmes d'équations aux dérivées partielles*, J. de Math. (8), 3 (1920), pp. 65–151.

[13] E. MANSFIELD, *Differential Gröbner Bases*, PhD thesis, Univ. of Sydney, 1991.

[14] J. D. MURRAY, *Mathematical Biology*, Springer-Verlag, 1989.

[15] P. OLVER, *Applications of Lie Groups to Differential Equations*, vol. 107 of Graduate Texts in Mathematics, Springer, 2nd ed., 1993.

[16] ———, *Equivalence, Invariants, and Symmetry*, Cambridge University Press, 1995.

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

[18] ———, *Finding abstract Lie symmetry algebras of differential equations without integrating determining equations*, European J. of Appl. Math., 2 (1991), pp. 319–340.

[19] G. REID, I. LISLE, A. BOULTON, AND A. WITTKOPF, *Algorithmic determination of commutation relations for Lie symmetry algebras of PDEs*, in Proc. ISSAC '92, Berkeley, P. Wang, ed., acm Press, 1992, pp. 63–68.

[20] G. REID, A. WITTKOPF, AND A. BOULTON, *Reduction of systems of nonlinear partial differential equations to simplified involutive forms*, European J. of Appl. Math., 7 (1996), pp. 635–666.

[21] C. RIQUIER, *Les systèmes d'équations aux dérivées partielles*, Gauthier-Villars, 1910.

[22] C. RUST, *Rankings of Derivatives for Elimination Algorithms and Formal Solvability of Analytic Partial Differential Equations*, PhD thesis, Univ. Chicago, 1998. www-address: www.cecm.sfu.ca/~rust.

[23] C. RUST AND G. REID, *Rankings of Partial Derivatives*, in Proc. ISSAC '97, Maui, W. Küchlin, ed., acm Press, 1997, pp. 9–16.

[24] C. RUST, G. REID, AND A. WITTKOPF, *Existence and Uniqueness Theorems for Formal Power Series Solutions of Analytic Differential Systems*, in Proc. ISSAC '99, Vancouver, S. Dooley, ed., acm Press, 1999, pp. 105–112.

[25] F. SCHWARZ, *An algorithm for determining the size of symmetry groups*, Computing, 49 (1992), pp. 95–115.

[26] ———, *Reduction and completion algorithms for Partial Differential Equations*, in Proc. ISSAC '92, Berkeley, P. Wang, ed., acm Press, 1992, pp. 49–56.

[27] ———, *Janet Bases of 2nd Order Differential Equations*, in Proc. ISSAC '96, Zurich, Y. N. Lakshman, ed., acm Press, 1996, pp. 179–188.

[28] W. SEILER, *Analysis and application of the formal theory of partial differential equations*, PhD thesis, Lancaster University, 1994.

[29] A. TRESSE, *Sur les invariants différentiels des groupes continus de transformations*, Acta Math., 18 (1894), pp. 1–88.

[30] E. VARLEY AND B. SEYMOUR, *Exact solutions for large amplitude waves in dispersive and dissipative systems*, Studies in Applied Math., 72 (1985), pp. 241–262.

[31] A. D. WITTKOPF AND G. J. REID, *Fast Differential Elimination in C: The DiffElim Algorithm*, to appear in Computer Physics Communications.

[32] T. WOLF, *A package for the analytic investigation and exact solutions of differential equations*, in Proc. EUROCAL '87, vol. 378, Springer, 1987, pp. 479–491.

[33] W. ZULEHNER AND W. F. AMES, *Group Analysis of a semilinear vector diffusion equation*, Nonlinear Analysis Theory Methods and Applications, 7 (1983), pp. 945–969.