ALGORITHMIC METHODS FOR LIE PSEUDOGROUPS

JOACHIM SCHÜ, WERNER M. SEILER* and JACQUES CALMET
Institut für Algorithmen und Kognitive Systeme
Universität Karlsruhe, W-7500 Karlsruhe 1, Germany
Email: kg04@dkauni2.bitnet

Abstract. We present an algorithm to complete any given system of differential equations to an involutive system as needed e.g. for concrete applications of Lie pseudogroups. It is based on jet bundle formalism and formal theory. An implementation in the computer algebra system AXIOM is described.

1. Introduction

A central tool to study non-linear partial differential equations is provided by symmetry analysis [1, 8]. The usual procedure starts with setting up the determining equations for the infinitesimal symmetry generators (usually with one of the many programs dveloped for this purpose). Then this overdetermined system of linear partial differential equations has to be solved. To perform symmetry reductions, differential invariants must be constructed. This again requires to solve linear partial differential equations.

The ultimate goal of our work is to avoid solving differential equations during symmetry analysis. That this may be possible was recently indicated by Reid [12], who demonstrated that one can obtain the structure of the symmetry algebra without solving determining equations. For symmetry reductions this is, however, only of limited use, because there explicit expressions for the generators are needed. We hope that it will be possible to overcome some of these problems using Lie pseudogroups. They can be informally described as groups of transformations given as solutions of a system of partial differential equations [9]. These equations are, however, never explicitely solved!

As Pommaret [9] pointed out, it is very important for the application of Lie pseudogroups that both the system to be analysed and the system defining the group are involutive. Thus as a first step we need an algorithm to complete any given system of partial differential equations to an equivalent involutive one. In this paper, we will present such an algorithm and its implementation in the computer algebra system AXIOM.

The paper is organized as follows: The next section shortly introduces the notion of an involutive system and outlines the algorithm. Section 3 deals with some of the occuring computational problems, while Section 4 describes the implementation. After two examples in Section 5, finally some conclusions are given.

* Present address: Centre de recherches mathématiques, Université de Montréal, C.P. 6128 Succ. A, Montréal H3C 3J7, Canada

2. Involutive Systems

There exist several notions of involution in different frameworks. We will use a geometric approach to differential equations and define them as fibred submanifolds in a jet bundle. This formalism is the one usually used in symmetry theory. Other approaches involve e.g. differential forms (Cartan-Kähler theory [2]) or differential algebra (Riquier-Janet theory [5]). We can not give here an introduction to the underlying theory but must refer for all details to the literature [9].

A differential equation $\mathcal{R}_q$ is given in local coordinates by a set of equations $\Phi^\tau(x^i, u^\alpha, p^\alpha_\mu) = 0$ where the derivatives p^α_μ are of order less or equal q. Its prolongation $\mathcal{R}_{q+1}$ is obtained by formally differentiating the equations with respect to the independent variables x^i. It is well-known that during prolongation integrability conditions can arise, i.e. the projected system $\mathcal{R}^{(1)}_q$ has a smaller dimension than $\mathcal{R}_q$. This can happen at any prolongation order. A system without integrability conditions is called *formally integrable*, because it is possible to construct order by order a formal power series solution.

A formally integrable system is, however, not yet necessarily *involutive*. We request further that it has an involutive *symbol*. The symbol $\mathcal{M}_q$ of $\mathcal{R}_q$ is a system of linear (algebraic, not differential!) equations in some unknowns v^α_μ defined by:

$$\mathcal{M}_q \, : \sum_{\alpha,|\mu|=q} \frac{\partial \Phi^\tau}{\partial p^\alpha_\mu} \, v^\alpha_\mu = 0 \, . \tag{1}$$

Thus, for a quasi-linear equation the symbol is essentially the highest order part of the equation.

The intrinsic definition of an involutive symbol makes use of the Spencer cohomology [3, 9]. It is, however, possible to construct an algorithmic criterion to decide involution directly from the equations (1) using multiplicative variables as known from Riquier-Janet theory: $\mathcal{M}_q$ is involutive, if

$$\#(\text{multiplicative variables of } \mathcal{M}_q) = \operatorname{rank} \mathcal{M}_{q+1} \, . \tag{2}$$

The Cartan-Kuranishi theorem states that every differential equation $\mathcal{R}_q$ becomes involutive after a finite number of prolongations and projections (i.e. addition of integrability conditions). In other words: there exist two integers r, s such that $\mathcal{R}^{(s)}_{q+r}$ is an involutive system. The proof of this theorem yields an algorithm to construct r, s and thus the involutive system. It consists of two nested loops. The inner one prolongs till the symbol becomes involutive; the outer one adds the integrability conditions.

Two fundamental theorems form the basis of this algorithm: (i) Every symbol becomes involutive after a finite number of prolongations; (ii) $\mathcal{R}_q$ is involutive, iff $\mathcal{M}_q$ is involutive and $\mathcal{R}^{(1)}_q = \mathcal{R}_q$. Latter condition can be easily checked comparing the dimensions of the two submanifolds. The dimension of the projected submanifold is given by:

$$\dim \mathcal{R}^{(1)}_q = \dim \mathcal{R}_{q+1} - \dim \mathcal{M}_{q+1} \, . \tag{3}$$

Equations (2) and (3) lead to the simple algorithm depicted in Figure 1. The loop in [4.1] prolongs, until the symbol is involutive; [4.2] checks for integrability

```
[1]       r ← 0; s← 0
[2]       compute $\mathcal{R}_{q+1}$                          {*prolong*}
[3]       compute $\mathcal{M}_q$,$\mathcal{M}_{q+1}$                  {*extract symbols*}
[4]       until $\mathcal{R}^{(s)}_{q+r}$ involutive repeat
[4.1]         while $\#multVar(\mathcal{M}^{(s)}_{q+r}) < \mathrm{rank}\,\mathcal{M}^{(s)}_{q+r+1}$ repeat
[4.1.1]           r ← r+1                              {*counter for prolongations*}
[4.1.2]           compute $\mathcal{R}^{(s)}_{q+r+1}$                {*prolong*}
[4.1.3]           compute $\mathcal{M}^{(s)}_{q+r+1}$                {*extract symbol*}
[4.2]         if $\dim\mathcal{R}^{(s)}_{q+r+1} - \dim\mathcal{M}^{(s)}_{q+r+1} < \dim\mathcal{R}^{(s)}_{q+r}$ then
[4.2.1]           s ← s+1                              {*counter for projections*}
[4.2.2]           compute $\mathcal{R}^{(s)}_{q+r}$                  {*add integrability conditions*}
[4.2.3]           compute $\mathcal{R}^{(s)}_{q+r+1}$                {*prolong*}
[4.2.4]           compute $\mathcal{M}^{(s)}_{q+r}$,$\mathcal{M}^{(s)}_{q+r+1}$       {*extract symbols*}
[5]       return $\mathcal{R}^{(s)}_{q+r}$
```

Fig. 1. Completion algorithm from the Cartan-Kuranishi theorem.

conditions. Their construction in step [4.2.2] is a purely linear operation. Equation (3) shows that they can only occur, if $\mathcal{M}_{q+1}$ has not maximal rank. The same linear combinations that yield zero rows in the symbol applied to the full system generate the integrability conditions, because a zero row in the symbol denotes a cancelling of the leading terms of the full equation.

3. Computational Problems

Our algorithm contains mostly simple operations: taking derivatives and computing ranks of matrices. The only non-trivial operation is the determination of the dimensions of the submanifolds $\mathcal{R}_{q+r}$. But already the ranks make trouble. The symbol $\mathcal{M}_{q+r}$ is a matrix whose entries can again be viewed as differential equations. Hence its rank may change depending on whether some additional equations are satisfied or not. In general, a whole tree of such case distinctions may exist, which must all be treated separately.

Similar problems can arise during prolongations. We have assumed throughout this paper, that we start with a regular system $\mathcal{R}_q$, i.e. all prolongations are again fibred submanifolds. Usually, however, only a suitable chosen restriction of $\mathcal{R}_q$ satisfies this condition. Thus we should check after each prolongation, whether the prolonged equations still define a variety of constant dimension.

This brings us back to the problem of determining the dimension of a submanifold defined by a set of equations. There exists no general algorithm for this purpose. One can design, however, many different algorithms for special classes of equations. The first approach would be to compute the rank of the Jacobi matrix. This rank has, however, to be evaluated *on* the submanifold. In general, this is only possible, if the equations are in solved form (e.g. quasi-linear equations).

For algebraic equations one could use Gröbner bases. The huge dimension of higher order jet bundles makes this not very promising. For homogeneous equations, i.e. equations which vanish, if the dependent variables and all derivatives are set equal to zero, a much simpler approach exists. There, $\mathcal{R}_{q+r}$ always contains the origin. The dimension of the submanifold can be computed as the dimension of the tangent space at the origin, i.e. through the rank of a linear system with *numerical* entries.

We omit here for lack of space the question of δ-regularity of the coordinate system [9]. In our algorithm it affects the criterion (2) for an involutive symbol, but it seems to be a more general problem (see e.g. [4]). It can be handled algorithmically, but only at the prize of a considerably higher complexity of the calculation.

A practical problem comes from the combinatorical explosion of the size of the occuring matrices. If several prolongations are necessary, the computations become quite lengthy. Consequent simplification of the equations can partially counter this effect. To what extent we can simplify depends on the class of equations treated. Linear and quasi-linear systems offer of course the most possibilities.

One of the most expensive operations in the algorithm is the prolongation of the equations. In principle it is only necessary for the determination of the dimension. The symbol can be prolonged independently using pure index manipulations, as one can easily see from its definition (1). Only when integrability conditions arise, we need explicitely the concerned equations. An alternative method to compute the dimension would thus lead to a considerable optimization.

The complexity of the algorithm is unknown. There exists only a bound for the number of prolongations necessary to get an involutive symbol [9]. It yields, however, usually much too high values (10^3 – 10^6 even for small examples!).

4. Implementation

We implemented the algorithm in the computer algebra system AXIOM [6]. Its highly modular and object-oriented design allows to imitate very closely the mathematical structure in the program. We have implemented general data structures and procedures for computations within the geometric theory of differential equations. This allows for an easy extension of the program for many other applications.

Figure 2 shows the present structure of the program. A more detailed description can be found in [14]. Italics denote some planned extensions; the final goal will be the implementation of the category and the domains in the dashed box.

C,D,P abbreviate the AXIOM structures category, domain and package: A *package* contains just a collection of procedures. A *category* defines an abstract datatype, i.e. it defines what operations are allowed in this type. A *domain* is a concrete implementation of such a category. In our programm e.g. `JetBundleFunctionCategory` contains basically only the signatures of operations like formal differentiation. The actual implementation of these procedures depends usually on the representation of the concrete chosen class of functions (linear, quasi-linear etc.). Thus it is contained in the corresponding domain.

We see that the domain `DifferentialEquation` and consequently the package `CartanKuranishi` are based only on the *category* `JetBundleFunctionCategory`. They are independent of the concrete implementation of any specific class (domain)

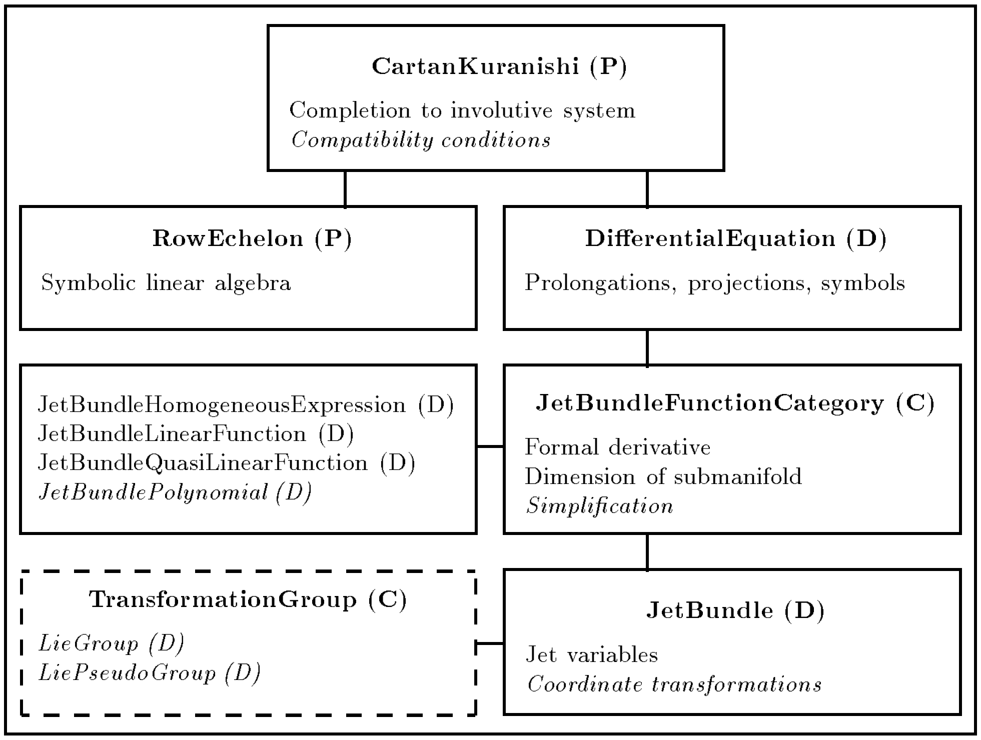
CartanKuranishi (P)
Completion to involutive system
Compatibility conditions
RowEchelon (P)
Symbolic linear algebra
DifferentialEquation (D)
Prolongations, projections, symbols
JetBundleHomogeneousExpression (D)
JetBundleLinearFunction (D)
JetBundleQuasiLinearFunction (D)
JetBundlePolynomial (D)
JetBundleFunctionCategory (C)
Formal derivative
Dimension of submanifold
Simplification
TransformationGroup (C)
LieGroup (D)
LiePseudoGroup (D)
JetBundle (D)
Jet variables
Coordinate transformations

Fig. 2. Overview of the structure of the program.

of differential equations and can be left unchanged, if one adds new domains or changes old ones within `JetBundleFunctionCategory`. As they use of course many operations which are implemented in these domains, their performance can change considerably when applied to different ones.

So far we have implemented three domains within `JetBundleFunctionCategory`: Homogeneous, linear and quasi-linear functions. The main reason is the before mentioned problem of computing the dimension of submanifolds. For the homogeneous functions we use the above outlined tangent space method, as it appears to be the most efficient one. Linear and quasi-linear systems provide of course no problems.

The procedures in the package `RowEchelon` compute all necessary information to detect case distinctions. They use a fairly primitive approach, namely applying Gaußian elimination and storing the pivots. Sit [17] pointed recently out, that this leads often to many redundant distinctions and described an alternative algorithm based on methods from algebraic geometry which might be more effective in detecting the degenerate cases. Presently, the completion procedure in `CartanKuranishi` ignores them anyway restricting itself to the generic case.

Due to its general ansatz, many extensions of the program are possible. Besides the application of an infinitesimal generator to the system, nearly all ingredients for a symmetry package are already there. Then the completion procedure could be used to analyse determining equations, e.g. to calculate the size of the symmetry group (see below). However, to apply this approach to realistic problems will request the implementation of simplification procedures to enhance the performance. Another straight-forward extension is the construction of compatibility conditions at least for linear systems.

5. Examples

First, we look at a standard example due to Janet [10]. It comprises two linear equations of second order in one dependent and three independent variables:

$$\mathcal{R}_2 : \begin{cases} u_{zz} + y\, u_{xx} &= 0\,, \\ u_{yy} &= 0\,. \end{cases} \tag{4}$$

Applying our algorithm we notice, that $\mathcal{M}_2$ is not involutive but $\mathcal{M}_3$. Since $\dim \mathcal{R}_3^{(1)} = 11 < \dim \mathcal{R}_3 = 12$, we find an integrability condition: $u_{xxy} = 0$. Because $\mathcal{M}_3^{(1)}$ is not involutive, we must prolong once to get the involutive symbol $\mathcal{M}_4^{(1)}$. But again an integrability condition occurs, namely $u_{xxxx} = 0$. Thus we have arrived at $\mathcal{R}_4^{(2)}$, who has, however, a non-involutive symbol. $\mathcal{M}_5^{(2)}$ vanishes and is therefore trivially involutive. As $\dim \mathcal{R}_5^{(3)} = 12 = \dim \mathcal{R}_5^{(2)}$, we are finished.

Because of the vanishing symbol, $\mathcal{R}_5^{(2)}$ is a finite type system and has a 12-dimensional solution space. This can also be seen by direct integration, which is easily performed in this simple example. $\mathcal{R}_4^{(2)}$ provides one of the few known examples of a formally integrable system that is not involutive.

To demonstrate some further possibilities of formal analysis, we consider the following linear system that arises in the theory of the KP equation [13].

$$\mathcal{R}_2 : \begin{cases} u_{xy} - u_t &= 0\,, \\ u_{xx} - u_y &= 0\,. \end{cases} \tag{5}$$

This second-order system is not yet involutive. We have to add the integrability condition $u_{yy} - u_{xt} = 0$ to get the involutive system $\mathcal{R}_2^{(1)}$.

A closer investigation of the symbol yields further information on the solution space [16]. We get the Cartan characters directly from its row echelon form. For (5) we find $\alpha_1 = 3, \alpha_2 = \alpha_3 = 0$. They allow to determine the arbitrariness of the general solution. Here it can be expressed by three functions each of one variable.

In our implementation, all these information are computed automatically at the end of the completion process. Figure 3 contains the input and the important part of the output of an AXIOM session analysing system (5). As AXIOM is a strongly typed system, it is sometimes necessary to declare types. Thus we introduce at the beginning abbreviations for some domains and packages. We use the ordering $x > y > t$. The `setOutput` command controls the amount of output generated.

6. Conclusions

We have presented an algorithm to complete a given system of partial differential equations to an involutive one within the framework of formal theory. There exist already several other, similiar algorithms using different approaches.

```
jbl := JetBundleLinearFunction('x,'u,'p,3,1)
lde := DifferentialEquation('x,'u,'p,3,1,jbl)
ck := CartanKuranishi('x,'u,'p,3,1,jbl)
eq1:jbl := P([0,0,2]) - P([0,1,0])
eq2:jbl := P([0,1,1]) - P([1,0,0])
sys:lde := generateSystem [eq1,eq2]
setOutput 12
complete sys
```

System $R_2^{(1)}$ involutive!

System without prolonged equations. Dimension: 7

$$
\begin{aligned}
-p_{[0,1,0]} + p_{[0,0,2]} &= 0,\\
-p_{[1,0,0]} + p_{[0,1,1]} &= 0,\\
p_{[1,0,1]} - p_{[0,2,0]} &= 0,
\end{aligned}
$$

Cartan characters: 3, 0, 0

Hilbert polynomial: 3

Number of arbitrary functions: 3, 0, 0

Fig. 3. In- and output for system (5).

Reid [11] implemented in Maple an algorithm to transform quasi-linear systems into a standard form based on Riquier-Janet theory. This algorithm yields in general only a formally integrable system. Another restriction is the necessity to have a quasi-linear systems in order to be able to solve for the leading derivatives. (The same is true for other implementations of Riquier-Janet theory [15, 18]). This allows on the other hand simplifications which improve the efficiency considerably and even lead to the definition of a standard form.

Mansfield [7] introduced differential Gröbner bases analogously to algebraic equations. The primary goal there is to obtain a normal form for bases of differential ideals. This requests to construct all integrability conditions of the system. They can be viewed as the differential analog to the S-polynomials in the Buchberger algorithm. Thus every differential Gröbner basis forms a formally integrable system. The converse does not hold in general, however, as the bases possess additional properties.

Hartley and Tucker [4] describe a REDUCE implementation of Cartan-Kähler theory, where they construct regular chains of involutive integral elements for exterior differential systems. This yields automatically the Cartan characters. Presently the program works only for involutive systems, as it does not yet contain a completion algorithm.

It is difficult to compare the different approaches, as for each one the class of equations for which it is applicable, the generated output and the complexity are different. For some applications, it might be more important to obtain a standard form; in others, one might prefer to get the Cartan characters. Our algorithm seems at present time to be the only one that is able to construct an equivalent involutive system for non-linear systems that are not in solved form. But it has e.g. compared with the algorithm of Reid a much higher complexity.

7. Acknowledgements

This work was partially supported by DFG. We thank J.-F. Pommaret for many helpful discussions. WMS is supported by Studienstiftung des deutschen Volkes and thanks P. Winternitz and the CRM for their hospitality.

References

1. G.W. Bluman, S. Kumei: *Symmetries and Differential Equations*, Springer, New York 1989.
2. R.L. Bryant, S.S. Chern, R.B. Gardner, H.L. Goldschmidt, P.A. Griffiths: *Exterior Differential Systems*, Springer, New York 1991.
3. H.L. Goldschmidt, J. Diff. Geom. 1 (1969) 269–307.
4. D. Hartley, R.W. Tucker, J. Symb. Comp. 12 (1991) 655–667.
5. M. Janet: *Leçons sur les Systèmes d'Équations aux Dérivées Partielles*, Gauthier-Villars, Paris 1929.
6. D. Jenks, R.S. Sutor: *AXIOM – The Scientific Computation System*, Springer, New York 1992.
7. E. Mansfield, E.D. Fackerell: *Differential Gröbner Bases*, Preprint 92-108, Macquarie University, Sydney 1992.
8. P.J. Olver: *Applications of Lie Groups to Differential Equations*, Springer, New York 1986.
9. J.F. Pommaret: *Systems of Partial Differential Equations and Lie Pseudogroups*, Gordon&Breach, London 1978.
10. J.F. Pommaret, A. Haddak: *Effective Methods for Algebraic Partial Differential Equations*, in: Proc. MEGA '90, T. Mora, C. Traverso (eds.), Birkhäuser, Boston 1991, pp. 411–426.
11. G.J. Reid, Eur. J. Appl. Math. 2 (1991) 293–318.
12. G.J. Reid, Eur. J. Appl. Math. 2 (1991) 319–340.
13. C. Rogers, W. Oevel, P.J. Vassiliou: *Invariance under Reciprocal Transformations in 1+1 and 2+1 Dimensions*, in: Proc. Int. Meeting on Nonlinear Diffusion Phenomena, Indian Institute of Science 1992.
14. J. Schü: *Implementierung des Cartan-Kuranishi-Theorems in AXIOM*, diploma thesis (in german), Karlsruhe 1992.
15. F. Schwarz: *Symmetries and Involution Systems: Some Experiments in Computer Algebra*, in: Proc. Oberwolfach Meeting on Nonlinear Evolution Equations, M. Ablowitz, B. Fuchssteiner, M. Kruskal (eds.), World Science Press, Singapore 1987.
16. W.M. Seiler: *On the Arbitrariness of the General Solution of an Involutive Partial Differential Equation*, Preprint CRM–1873, Montréal 1993.
17. W.Y. Sit, J. Symb. Comp. 13 (1992) 353–394.
18. V.L. Topunov, Acta Appl. Math. 16 (1989) 191–206.