Journal of Symbolic Computation 79 (2017) 482–498

Contents lists available at ScienceDirect

Journal of Symbolic Computation

www.elsevier.com/locate/jsc

Algorithmic calculus for Lie determining systems

Ian G. Lisle [1], S.-L. Tracy Huang

Faculty of Education, Science, Technology and Mathematics, University of Canberra, ACT 2601, Australia

ARTICLE INFO

Article history:
Received 11 March 2014
Accepted 27 January 2016
Available online 29 March 2016

MSC:
35-04
35N10
22-04
53-04

Keywords:
Determining equations
Lie symmetry
Lie algebra
Structure constants
Differential elimination
Algorithm

ABSTRACT

The infinitesimal symmetries of differential equations (DEs) or other geometric objects provide key insight into their analytical structure, including construction of solutions and of mappings between DEs. This article is a contribution to the algorithmic treatment of symmetries of DEs and their applications. Infinitesimal symmetries obey a determining system L of linear homogeneous partial differential equations, with the property that its solution vector fields form a Lie algebra $\mathcal{L}$. We exhibit several algorithms that work directly with the determining system without solving it. A procedure is given that can decide if a system specifies a Lie algebra $\mathcal{L}$, if $\mathcal{L}$ is abelian and if a system L' specifies an ideal in $\mathcal{L}$. Algorithms are described that compute determining systems for transporter, Lie product and Killing orthogonal subspace. This gives a systematic calculus for Lie determining systems, enabling computation of the determining systems for normalisers, centralisers, centre, derived algebra, solvable radical and key series (derived series, lower/upper central series). Our methods thereby give algorithmic access to new geometrical invariants of the symmetry action.

E-mail addresses: Ian.Lisle@canberra.edu.au (I.G. Lisle), Tracy.Huang@uni.canberra.edu.au (S.-L. Tracy Huang).

[1] The first author passed away on 27 September 2015 after he made the final versions to the paper and before it was accepted for publication. This work was part of the main research with his graduate student Tracy Huang. The entire project was carried while he battled cancer.

http://dx.doi.org/10.1016/j.jsc.2016.03.002

1. Introduction

The symmetries of a geometric object (differential equation, exterior differential system, Riemannian metric,...) are one of its most fundamental features. They arise in various ways: to capture physical invariance properties (Bluman and Anco, 2002, §1), or as a reflection of basic mathematical properties such as linearity (Bluman et al., 2010, §2). In the study of differential equations (DEs) they give powerful methods for constructing solutions (Bluman and Anco, 2002; Olver, 1993; Ovsiannikov, 1982), and form the basis of computer algebra solution techniques (Schwarz, 2008). In computer algebra applications, the DE or other object is given, and must be analysed for its symmetries. Typically this involves 'infinitesimal' methods: if the DE or other object is on a space with coordinates $x = (x^1, \dots, x^n)$, one seeks a vector field $\xi^i(x)\frac{\partial}{\partial x^i}$ which leaves the object invariant. This gives rise to a linear homogeneous system of 'defining' or 'determining' differential equations for the symmetry vector field components $\xi^i(x)$. Examples are Lie point and contact symmetries of DEs (Bluman and Anco, 2002), 'intrinsic' or Cartan symmetries (Krasil'shchik and Vinogradov, 1999, §3.7), and Killing equations for infinitesimal isometries of a Riemannian space.

Many packages are available in computer algebra systems for finding determining systems (Carminati and Vu, 2000; Cheviakov, 2007; Rocha Filho and Figueiredo, 2011). These packages then rely on solving the system to obtain symmetry vector fields explicitly. However there is great appeal in devising methods that can infer properties of a Lie algebra directly from the determining system without solving it. There are several reasons for this. First, the determining system is the immediately available, algorithmically constructible object, whereas the *solutions* of the determining system are available only via application of integration heuristics, which inevitably fail in some instances. Second, the determining system often has coefficients that live in a computable field (rational numbers, rational functions), whereas the solutions may involve algebraic or transcendental extensions (e^x, $\sqrt{1+x^2}$ etc.) which make computer algebra manipulations clumsier and less reliable. Finally, the very notion of a 'Lie pseudogroup' of transformations is *defined* in terms of satisfaction of a determining system (Pommaret, 1978; Singer and Sternberg, 1965; Stormark, 2000); it is therefore mathematically natural to cultivate techniques which stay close to the definition and work with the system rather than its solutions. An added attraction is that finite and infinite Lie pseudogroups can be treated in a unified way.

Some previous work has exploited differential reduction and completion methods to obtain properties of a Lie algebra directly from the determining system. Schwarz (1992a) noted that the dimension of a symmetry group could be inferred directly from the system, using the classical local existence–uniqueness (E–U) theorem of Riquier (1910). Subsequently, Reid et al. (1992) exhibited a method for finding the structure constants c_{ij}^k of a Lie algebra with respect to a certain basis characterised by the determining system, without knowing the solutions. Lisle et al. (2014) extended this work to extract c_{ij}^k in the case where the determining system has been integrated fully or partially. Reid et al. (1992) suggested using their c_{ij}^k as inputs to Lie algebraic algorithms such as are described in de Graaf (2000). Application to symmetry of DE additionally requires knowledge of how the Lie group acts on space. For example, Draisma (2001) used knowledge of Reid's c_{ij}^k, plus geometric information (transitivity, isotropy algebra) available at the level of determining systems (Lisle and Reid, 1998) to give an algorithm that identifies the 'symmetry type' of an ODE of order $k \geq 2$. An alternative approach to identifying the symmetry type of ODE is taken by Schwarz (2008), again working at the level of determining systems.

From the point of view of general Lie algebras, acting on general spaces, the above work is fragmentary, being restricted for example to action on 2-dim (x, y) space. Apart from the work of Reid et al. (1992) and Lisle et al. (2014) for finding structure constants c_{ij}^k, there is no systematic calculus for dealing with determining systems of arbitrary Lie algebras of vector fields. Our purpose is to fill this gap.

This paper is one of a sequence where we develop a toolkit of algorithms for determining systems, with the goal of extracting algebraic information (isomorphism invariants) about the Lie algebra, and geometric information (diffeomorphism invariants) about the vector fields that constitute it. As well as decision procedures, for example testing whether such a system really defines a Lie algebra $\mathcal{L}$, we show how to extract determining systems for the vector fields that make up various structural parts of $\mathcal{L}$, such as the derived algebra, solvable radical, centre, etc. This gives us access algorithmically to many more diffeomorphism invariants than have previously been available. Our primary tool is the local E–U result for linear homogeneous PDE that are differentially complete. None of our methods require solving the system, and all have been implemented in the Maple computer algebra language.

The remainder of this paper is organised as follows. First in §2, we provide the required definitions and establish our notation. Our main results, in the form of algorithms constituting our calculus, are presented in §3, and further illustrated and extended in §4. Finally, in §5 Discussion, we briefly discuss computer algebra implementation and indicate some further extension of the work.

2. Mathematical background

2.1. Lie algebras

A *Lie algebra* $\mathcal{L}$ is a vector space over a field (assumed of characteristic 0) endowed with a skew-symmetric bilinear operation (Lie bracket), $[V, V']$ (where $V, V' \in \mathcal{L}$) satisfying the Jacobi identity (Bourbaki, 1989; Jacobson, 1962). In this paper we consider only the case of $\mathcal{L}$ finite-dimensional. We establish the following notational conventions. The dimension of $\mathcal{L}$ is $r_{\mathcal{L}}$, while a typical member of $\mathcal{L}$ will be denoted by $V_{\mathcal{L}}$. A basis of $\mathcal{L}$ will be denoted by $\{X_{\mathcal{L}i}\}_{i=1}^{r_{\mathcal{L}}}$. We adopt the summation convention over repeated indices throughout. If the Lie algebra is clear from context, we drop the subscript (V rather than $V_{\mathcal{L}}$ etc.).

2.1.1. Structure coefficients

For the basis $\{X_{\mathcal{L}i}\}$ of $\mathcal{L}$ there are structure constants c_{ij}^k such that $[X_{\mathcal{L}i}, X_{\mathcal{L}j}] = c_{ij}^k X_{\mathcal{L}k}$. The c_{ij}^k determine $\mathcal{L}$ up to isomorphism, and are inputs to algorithmic procedures for many structural features of $\mathcal{L}$ (de Graaf, 2000). In the sequel, we will require the following generalisation. Let $\mathcal{L}$, $\mathcal{M}$, $\mathcal{N}$ be subspaces of some Lie algebra, such that $[\mathcal{L}, \mathcal{M}] \subseteq \mathcal{N}$. Then there are 'structure coefficients' f_{ij}^k such that

$$[X_{\mathcal{L}i}, X_{\mathcal{M}j}] = f_{ij}^k X_{\mathcal{N}k} \tag{1}$$

2.1.2. Transporter

The following general concept will be of great use to us:

Definition 2.1 *(Transporter (Bourbaki, 1989)).* Let $\mathcal{L}$, $\mathcal{M}$, $\mathcal{N}$ be subspaces of some Lie algebra. The *transporter* of $\mathcal{M}$ to $\mathcal{N}$ in $\mathcal{L}$ is the subspace

$$T_{\mathcal{L}}(\mathcal{M}, \mathcal{N}) = \{V_{\mathcal{L}} \in \mathcal{L} \mid [V_{\mathcal{L}}, V_{\mathcal{M}}] \in \mathcal{N}, \text{ for all } V_{\mathcal{M}} \in \mathcal{M}\} \tag{2}$$

If $\mathcal{M}$ and $\mathcal{N}$ are ideals in $\mathcal{L}$, then $T_{\mathcal{L}}(\mathcal{M}, \mathcal{N})$ is an ideal in $\mathcal{L}$.

2.1.3. Lie product

We note the following definitions (Bourbaki, 1989; Jacobson, 1962). Let $\mathcal{L}$, $\mathcal{M}$ be subspaces of a Lie algebra. The *Lie product* $[\mathcal{L}, \mathcal{M}]$ is the space generated by the brackets $\{[V_{\mathcal{L}}, V_{\mathcal{M}}] \mid V_{\mathcal{L}} \in \mathcal{L}, V_{\mathcal{M}} \in \mathcal{M}\}$. A subalgebra $\mathcal{M} \subseteq \mathcal{L}$ is a subspace satisfying $[\mathcal{M}, \mathcal{M}] \subseteq \mathcal{M}$. An ideal is a subalgebra satisfying $[\mathcal{L}, \mathcal{M}] \subseteq \mathcal{M}$. If $\mathcal{M}, \mathcal{N}$ are ideals in $\mathcal{L}$ then their Lie product $[\mathcal{M}, \mathcal{N}]$ is an ideal in $\mathcal{L}$.

2.1.4. Adjoint representation, Killing form

The *adjoint action* $\operatorname{ad} V$ (Bourbaki, 1989; de Graaf, 2000; Jacobson, 1962) of a Lie algebra $\mathcal{L}$ on itself is the linear map from $\mathcal{L}$ to $\mathcal{L}$ defined by $\operatorname{ad} V(V') = [V, V']$. With respect to a basis of $\mathcal{L}$, the operator $\operatorname{ad} V$ is represented by a square matrix. The *Killing form* K is the symmetric bilinear form on $\mathcal{L}$ defined by $K(V, V') = \operatorname{tr}(\operatorname{ad} V . \operatorname{ad} V')$.

2.1.5. Killing orthogonal

If $\mathcal{M}$ is a subspace of $\mathcal{L}$, then its *K-orthogonal subspace* $\mathcal{M}_K^\perp$ is

$$\mathcal{M}_K^\perp = \{V_\mathcal{L} \in \mathcal{L} \mid K(V_\mathcal{L}, V_\mathcal{M}) = 0 \text{ for all } V_\mathcal{M} \in \mathcal{M}\} \tag{3}$$

If $\mathcal{M}$ is an ideal in $\mathcal{L}$ then $\mathcal{M}_K^\perp$ is an ideal.

Various other constructions arise as special cases of Transporter, Lie product and K-orthogonal space; our algorithms and implementations will reflect this. Simple algorithms for some Lie algebra entities are described in de Graaf (2000). They take as their starting point the structure constants c_{ij}^k of $\mathcal{L}$ with respect to some basis; their goal is to exhibit a basis for the desired subspace or ideal. In the current paper, our goal is different: we work with annihilators rather than the subspace itself, as these are the quantities we can explicitly construct, and we seek to avoid bases altogether where possible.

2.2. Linear homogeneous PDE

We are concerned with Lie algebras $\mathcal{L}$ that arise in symmetry analysis (e.g. of DEs), so that the elements of $\mathcal{L}$ are vector fields. These vector fields may not be explicitly known, but are to be solutions of a system of linear homogeneous partial differential equations (the determining system). Our goal is to find properties of $\mathcal{L}$ directly from these LHPDEs.

Let there be $n \geq 1$ independent variables $(x^1, x^2, \dots, x^n)$, with associated (commuting) derivations $\boldsymbol{\partial} = (\frac{\partial}{\partial x^1}, \dots, \frac{\partial}{\partial x^n})$ and m dependent variables $U = (u^1, u^2, \dots, u^m)$. Let $J = (j_1, j_2, \dots, j_n) \in \mathbb{N}^n$ be an n-tuple, and let $|J| = j_1 + \cdots + j_n$. Then we shall write ∂^J for the $|J|$-th order derivative operator $\partial^J := \left(\frac{\partial}{\partial x^1}\right)^{j_1} \cdots \left(\frac{\partial}{\partial x^n}\right)^{j_n}$.

We consider coefficients from a differential field $(\mathbb{F}, \boldsymbol{\partial})$ of characteristic 0, which is to be a subfield of the field of complex meromorphic functions in $(x^1, \dots, x^n)$. We suppose that $\mathbb{F}$ is computable, that is, zero recognition is possible in $\mathbb{F}$, and arithmetic operations and differentiation can be effectively performed. Typically $\mathbb{F}$ will be the rational function field $\mathbb{Q}(x^1, \dots, x^n)$, the constant field $\mathbb{Q}$, or some computable extension of these.

A linear homogeneous PDE (LHPDE) is an expression of the form

$$\sum_{i=1}^{m} \sum_{J \in \mathbb{N}^n} a_{iJ} \partial^J u^i \tag{4}$$

where the coefficients $a_{iJ} \in \mathbb{F}$. A system of LHPDEs is a finite collection of such expressions (4).

The above theory can be put on a secure algebraic footing (Blinkov et al., 2003b; Castro-Jiménez and Moreno-Frías, 2001; Schwarz, 2008): it is analogous to computation in the free module of rank m over a polynomial ring $\mathbb{F}[x^1, \dots, x^n]$ (Adams and Loustaunau, 1994; Cox et al., 2005). Indeed if $\mathbb{F}$ is a constant field like $\mathbb{Q}$, the polynomial theory applies unchanged, but even in the more general case, the noncommutativity noted above is minor enough as to barely disturb the usual proofs.

The principal aims of differential algebraic treatment of LHPDE systems are, first, *completion* to a more favourable basis for the differential module generated by the system and second, *reduction* modulo such a basis. The reduction process relies on ranking the derivatives (Reid et al., 1996; Adams and Loustaunau, 1994; Rust and Reid, 1997) by a linear order relation $\preceq$ satisfying (i) $\partial^J u \preceq \partial^I(\partial^J u)$ and (ii) if $\partial^{J_1} u \preceq \partial^{J_2} v$ then $\partial^I \partial^{J_1} u \preceq \partial^I \partial^{J_2} v$. The highest-ranked derivative $\mathrm{Ld}(f)$ in an LHPDE f with respect to $\preceq$ is called its leading derivative. We always suppose that each LHPDE e_i in system $\mathrm{Sys} = \{e_1, \dots, e_s\}$ is arranged monic, i.e. the coefficient of $\mathrm{Ld}(e_i)$ is 1. Reduction of LHPDE f modulo system Sys amounts to repeatedly finding a derivative in f that is of form $\partial^J\big(\mathrm{Ld}(e_i)\big)$ and replacing it by the 'tail' of $\partial^J e_i$. The resulting reduced form of f consists entirely of terms that are not a derivative of any leader $\mathrm{Ld}(e_i)$ of $e_i \in \mathrm{Sys}$. A detailed algorithm is given in Schwarz (1992b); we henceforth assume availability of a procedure

$$\mathrm{Reduce}(f) \quad \mathrm{mod}\ \mathrm{Sys}, \preceq$$

which reduces f modulo system Sys with ranking $\leq$. If the ranking is of no importance, we suppress explicit reference to it.

As in the polynomial case, the reduced form of $f \mod \mathrm{Sys}$ is not necessarily unique, unless Sys has additional properties. The following notion is of central importance.

Definition 2.2 *(Riquier basis).* A system $\mathrm{Sys} = \{e_1, \ldots, e_s\}$ of monic LHPDE is a *Riquier basis* (Rust et al., 1999) with respect to ranking $\leq$ if for all equations $e_i, e_j \in \mathrm{Sys}$ and all differentiations ∂^I, ∂^J such that $\mathrm{Ld}(\partial^I e_i) = \mathrm{Ld}(\partial^J e_j)$, the integrability condition $\partial^I e_i - \partial^J e_j$ reduces to 0 mod Sys.

A *reduced Riquier basis* is one in which each LHPDE $e_i \in \mathrm{Sys}$ is in reduced form modulo the others. This is the case for example for Reid's *rif* algorithm (Reid et al., 1996) and other computer algebra packages (Gerdt, 1999; Blinkov et al. 2003a, 2003b). Our examples will always use reduced Riquier basis, but this is not essential to the theory. By a usual abuse of notation, we shall write LHPDE as equations, and if a derivative is isolated on the left-hand side, it is understood that this is the leading derivative with respect to a given ranking.

Example 2.3. The LHPDE system with independent variables (x, y), dependent variables (ξ, η) and system

$$\xi_{xx} = \frac{-2}{y-x}\xi_x + \frac{2}{(y-x)^2}(\eta - \xi), \quad \xi_y = 0, \quad \eta_x = 0, \quad \eta_y = -\xi_x + \frac{2}{y-x}(\eta - \xi) \tag{5}$$

is a reduced Riquier basis with respect to a ranking which (i) ranks by total order of differentiation, then (ii) ranks $\eta > \xi$, then (iii) ranks $\partial_y > \partial_x$. The derivatives on the left are higher ranked than derivatives on the right; and all integrability conditions (e.g. $\partial_x(-\xi_x + \frac{2}{y-x}(\eta - \xi)) - \partial_y(0)$) reduce to triviality modulo the original system.

Riquier bases for LHPDE are the differential analogue of Gröbner bases for submodules of free modules over a polynomial ring (Adams and Loustaunau, 1994; Cox et al., 2005). 'Janet bases' (Schwarz, 1992b; Blinkov et al., 2003b) and 'involutive bases' (Gerdt, 1999) are closely related. Because determining systems are linear, the difficulties that arise with 'differential Gröbner bases' (Boulier et al., 1995; Mansfield, 1991) in the nonlinear case are absent here.

A key result is the following (see Rust et al., 1999 for a proof):

Proposition 2.4. *Let* $\mathrm{Sys} = \{e_1, \ldots, e_s\}$ *be a Riquier basis and* f *an LHPDE. Then* f *has a unique reduced form modulo* Sys.

In particular this implies that f is a consequence of Sys (in the sense that $f = \sum_i \Delta_i(e_i)$ for some $\Delta_i \in \mathcal{R}$) if and only if f reduces to 0 mod Sys.

A Riquier basis Sys partitions the set of derivatives into two classes: *principal* derivatives, which are leaders $\mathrm{Ld}(e_i)$ or derivatives of leaders; and the complementary set of *parametric* derivatives. We say Sys is of *finite type* if there are finitely many r parametric derivatives. In this paper we restrict ourselves to systems of finite type. Assigning values to the parametric derivatives at a regular point x_0 will be called a 'specification of initial data'.

Example 2.5. Continuing Example 2.3, the $r = 3$ parametric derivatives of the system are (ξ, η, ξ_x). If a_1, a_2, b are constants then a specification of initial data is $\xi(x_0, y_0) = a_1$, $\eta(x_0, y_0) = a_2$, $\xi_x(x_0, y_0) = b$; this can be done at any regular point where $y_0 \neq x_0$.

Let the solution space of LHPDE system Sys in a neighbourhood of x_0 be denoted by $\mathcal{S}_{x_0}$. We define functionals $\{\mathrm{Par}^k_{x_0}\}_{k=1}^r$ on $\mathcal{S}_{x_0}$ that evaluate the k-th parametric derivative of a solution at x_0.

Theorem 2.6 *(Riquier existence–uniqueness). Let* Sys *be a Riquier basis for a LHPDE system of finite type, with* r *parametric derivatives. Let* $x_0 \in \mathbb{C}^n$ *be a regular point for* Sys*, and let* $(a^1, \ldots, a^r) \in \mathbb{C}^r$ *be initial data* *values. Then there is a unique formal power series solution $u(x)$ of* Sys *at x_0 such that* $\mathrm{Par}^k_{x_0}(u) = a^k$ *for each $k = 1, \ldots, r$. Also, every solution to* Sys *at x_0 may be obtained in this way for some initial data values.*

For a proof see Rust et al. (1999). With some care it can be shown that the formal power series are convergent, so that the space of initial data is isomorphic to $\mathcal{S}_{x_0}$. The classical proof is in Riquier (1910); great care must be exercised with rankings, as shown by the counterexample of Lemaire (2003).

By Theorem 2.6, the dimension r of $\mathcal{S}_{x_0}$ is equal to the number of parametric derivatives in a Riquier basis for the system. Moreover $\{\mathrm{Par}^k_{x_0}\}_{k=1}^r$ is a basis of the dual space $\mathcal{S}^*_{x_0}$.

Example 2.7. Continuing Examples 2.3 and 2.5, the Riquier basis (5) is of finite type, having 3 parametric derivatives and hence a 3-dimensional solution space. If $P = (x_0, y_0)$ is a regular point (so $y_0 \neq x_0$) then the parametric derivative evaluation functionals Par^k_P are

$$\mathrm{Par}^1_P(\xi,\eta) = \xi(x_0, y_0), \quad \mathrm{Par}^2_P(\xi,\eta) = \eta(x_0, y_0), \quad \mathrm{Par}^3_P(\xi,\eta) = \xi_x(x_0, y_0)$$

and form a basis for the dual space $\mathcal{S}^*_P$ to the solution space of system (5) near P. The induced basis of $\mathcal{S}_P$ consists of the solutions $(\xi_i, \eta_i)_{i=1}^3$ of the initial value problems for (5) with $\mathrm{Par}^k_P(\xi_i, \eta_i) = \delta^k_i$ (Kronecker-δ).

To each LHPDE system is naturally associated a linear homogeneous partial differential operator (LHPDO) Ξ.

Example 2.8. The LHPDO Ξ for system (5) is given by

$$(\xi,\eta) \overset{\Xi}{\mapsto} \left[\xi_{xx} - \tfrac{-2}{y-x}\xi_x - \tfrac{2}{(y-x)^2}(\eta - \xi), \quad \xi_y, \quad \eta_x, \quad \eta_y + \xi_x - \tfrac{2}{y-x}(\eta-\xi)\right]$$

The local solution space $\mathcal{S}_{x_0}$ of LHPDE Sys near x_0 is a vector space, and algorithms are available for basic vector space operations:

- IsSubspace(Sys′, Sys), which decides if $\mathcal{S}'_{x_0} \subseteq \mathcal{S}_{x_0}$ for each x_0.
- Intersection(Sys, Sys′) (respectively VectorSpaceSum(Sys, Sys′)), which returns LHPDE whose solution space for each x_0 is the intersection $\mathcal{S}_{x_0} \cap \mathcal{S}'_{x_0}$ (respectively vector space sum $\mathcal{S}_{x_0} + \mathcal{S}'_{x_0}$).

For the last two see e.g. Algorithms 2.9, 2.10 of Schwarz (2008).

Note that because our focus is on solutions of LHPDE our IsSubspace, Intersection, VectorSpaceSum are relative to the *solution space* of the determining system. In differential algebra, these operations would be stated relative to the differential modules generated by the systems, so that $+$ and $\cap$ would be exchanged, and the direction of containment $\subseteq$ reversed.

2.3. Lie vector field systems

Consider a vector field V on a space with coordinates $x = (x^1, \ldots, x^n)$

$$V = \xi^i(x)\frac{\partial}{\partial x^i} \tag{6}$$

If V, V' are any two vector fields on $(x^1, \ldots, x^n)$, locally defined on some neighbourhood of x_0, then their Lie bracket or commutator (Olver, 1993) is the vector field

$$[V, V'] = \left(\xi^j \frac{\partial \xi'^i}{\partial x^j} - \xi'^j \frac{\partial \xi^i}{\partial x^j}\right)\frac{\partial}{\partial x^i} \tag{7}$$

An LHPDE system Sys whose dependent variables $(\xi^1, \ldots, \xi^n)$ represent the components of such a vector field will be called a *vector field system* (VFS). Note that specification of the vector field structure is extrinsic to the LHPDE themselves: we therefore denote such a system L by $L = \mathrm{VFS}(V, \mathrm{Sys})$. The LHPDE Sys specify the vector space attributes of solutions $\mathcal{S}_{x_0}$, while vector field V specifies how to take Lie bracket of solutions of the VFS, and how to change variable in the VFS. The local solutions for $(\xi^1, \dots, \xi^n)$ (i.e. $\mathcal{S}_{x_0}$) are more satisfactorily thought of as local vector fields according to (6); the space of local solution vector fields will be denoted by $\mathcal{L}_{x_0}$. If the Lie bracket (7) of any two solutions of the VFS is also a solution of the VFS, we call this a *Lie* VFS. The local solution spaces $\mathcal{L}_{x_0}$ are then Lie algebras of (local) vector fields (LAVF). Collectively these local vector field solutions constitute an *infinitesimal Lie pseudogroup* $\mathcal{L}$. For a more thorough geometric treatment of these ideas, see Stormark (2000), Pommaret (1978) or Singer and Sternberg (1965). In this paper, we restrict ourselves to finite-dimensional Lie algebras, so all our VFS are assumed to be of finite type.

Lie VFS arise naturally as the determining (or 'defining') systems for infinitesimal symmetries of geometric objects such as a differential equations (Bluman and Anco, 2002; Olver, 1993; Ovsiannikov, 1982). The reason for our using the term VFS is for precision: a VFS does include the vector field structure, but does not assume closure under Lie bracket. In the works cited above it can be unclear whether 'determining system' includes these conditions. Unlike Ovsiannikov (1982, §7.15) we do not insist on closure under bracket, as many of our algorithms do not require it.

Example 2.9. Consider the vector fields on $\mathbb{R}^2(x, y)$

$$\frac{\partial}{\partial x}, \frac{\partial}{\partial y}, x\frac{\partial}{\partial y}$$

These generate a Lie algebra $\mathcal{L}$, which can be specified by the Lie VFS

$$V = \xi\frac{\partial}{\partial x} + \eta\frac{\partial}{\partial y}, \quad \text{where } \xi_x = 0,\ \ \xi_y = 0,\ \ \eta_{xx} = 0,\ \ \eta_y = 0 \tag{8}$$

The parametric derivatives are

$$\xi, \eta, \eta_x \tag{9}$$

Example 2.10 *(A^2).* The Lie algebra of the general affine group in the plane consists of vector fields of the form

$$V = (ax + by + e)\frac{\partial}{\partial x} + (cx + dy + f)\frac{\partial}{\partial y} \tag{10}$$

where a, b, c, d, e, f are constants. This is specified by the Lie VFS for A^2

$$V = \xi\frac{\partial}{\partial x} + \eta\frac{\partial}{\partial y}, \quad \text{where } \xi_{xx} = \xi_{xy} = \xi_{yy} = 0,\ \ \eta_{xx} = \eta_{xy} = \eta_{yy} = 0 \tag{11}$$

The parametric derivatives are

$$\xi, \eta, \xi_x, \xi_y, \eta_x, \eta_y \tag{12}$$

We now aim to use the methods for VFS described above to develop a transformational calculus for Lie VFS, that is, working directly with determining systems for Lie algebras of vector fields.

3. Calculus for Lie vector field systems

We now proceed to the main thrust of this paper, which is to extract properties of a Lie algebra of vector fields (LAVFs) $\mathcal{L}$, by direct algorithmic manipulation of the Lie VFS L that they satisfy. First we present some basis-free decision procedures that algorithmically decide whether (the solutions of) a VFS L possess certain Lie-algebraic properties. We then exhibit algorithms which take in a Lie VFS L with solution LAVF $\mathcal{L}$ and return Lie VFS obeyed by important structural components of $\mathcal{L}$

(e.g. its solvable radical or derived algebra). We will avoid basis-dependent calculation as much as possible; and where intermediate calculations rely on resolution with respect to a basis, the VFS finally returned is basis-free. Our approach is to use a small number of general constructions, in terms of which most important Lie algebraic attributes can be expressed.

3.1. General considerations

We consider a VFS $L = \mathrm{VFS}(V_L, \mathrm{Sys}_L)$ to consist of a space with coordinates $x = (x^1, \ldots, x^n)$, a formal vector field $V_L = \xi_L^i \frac{\partial}{\partial x^i}$ on that space, and a system Sys_L of LHPDE with independent and dependent variables x^i, ξ_L^i respectively, which we shall always assume to be in the form of a Riquier basis (Definition 2.2) with respect to some ranking. The space of local solution vector fields near x_0 of VFS L will be denoted by $\mathcal{L}_{x_0}$. By the dimension of L we shall mean the dimension of $\mathcal{L}_{x_0}$ in a neighbourhood of a regular point $x = x_0$: this dimension will be denoted by r_L. The LHPDO associated with L will be denoted by Ξ_L, and we regard this as acting on vector fields V_L (rather than on components ξ_L^i). The r_L parametric derivative functionals Par^j_{L,x_0} again will be regarded as operating on vector fields V_L, so that $\mathrm{Par}^j_{L,x_0}(V_L)$ represents evaluation of the j-th parametric derivative of V_L at point $x = x_0$. If L is a Lie VFS then $\mathcal{L}_{x_0}$ is a Lie algebra of vector fields. If the VFS L is obvious from context, the subscript L may be dropped. For solutions, we may suppress explicit notation of the base point x_0, but it is always implied present.

Before proceeding with the main algorithms, we note that applying the remarks at the end of §2.2 to VFS L, M (with solution LAVFs $\mathcal{L}$, $\mathcal{M}$) yield the following:

- Decision procedure $\mathrm{IsSubspace}(M, L)$, which decides if $\mathcal{M} \subseteq \mathcal{L}$. This is differential submodule containment, and amounts to computing
$$\mathrm{Reduce}(\mathrm{Sys}_L) \quad \mathrm{mod}\ \mathrm{Sys}_M$$
and deciding whether it gives 0. It is essential that Sys_M be a Riquier basis.
- Availability of IsSubspace gives a decision procedure for equality $\mathcal{L} = \mathcal{M}$ of (the solution sets of) two VFS.
- Procedure $\mathrm{Intersection}(L, M)$ for constructing a VFS whose solution space is $\mathcal{L} \cap \mathcal{M}$. This is got by taking the union of Sys_L and Sys_M.

Having established the above, we now proceed to our main algorithms.

3.2. Test for commutativity

Let L, M, N be VFS on the same space, with solution spaces $\mathcal{L}$, $\mathcal{M}$, $\mathcal{N}$ respectively (near x_0, which is not explicitly denoted) and consider the question: does $\mathcal{L}$ commute with $\mathcal{M}$ modulo $\mathcal{N}$? That is, we seek to test whether

$$[V_L, V_M] \in \mathcal{N} \quad \text{for all } V_L \in \mathcal{L},\ V_M \in \mathcal{M} \tag{13}$$

which we abbreviate to $[L, M] \subseteq N$. If Ξ_N is the LHPDO associated with N, the question amounts to whether

$$\Xi_N\big([V_L, V_M]\big) = 0 \quad \text{for all } V_L \in \mathcal{L},\ V_M \in \mathcal{M}$$

The left-hand side of this expression when expanded is a tuple of expressions bilinear in various derivatives $\partial^I \xi_L^i \cdot \partial^J \xi_M^j$. Since V_L, V_M are to be solutions of the determining systems Sys_L, Sys_M, one can reduce these expressions mod Sys_L, Sys_M, to give expressions bilinear in parametric derivatives $\mathrm{Par}^l_L(V_L) \cdot \mathrm{Par}^m_M(V_M)$. Since Sys_L, Sys_M are supposed to be Riquier bases, these expressions vanish identically if and only if all coefficients of $\mathrm{Par}^l_L(V_L) \cdot \mathrm{Par}^m_M(V_M)$ vanish. This justifies Algorithm 1.

Essentially one replaces the condition of vanishing for all V_L, V_M by vanishing for all values of parametric derivatives of Sys_L, Sys_M.

Algorithm 1 Commutation of L, M mod N.

Algorithm COMMUTES(L, M, N)
INPUT: VFS L, M, N, all on the same space
OUTPUT: true if L commutes with M mod N, false otherwise
$R := \Xi_N\big([V_L, V_M]\big)$
$R_{\mathrm{red}} := \mathrm{Reduce}(R)\quad \mathrm{mod}\ \{\mathrm{Sys}_L, \mathrm{Sys}_M\}$
return boolean$(R_{\mathrm{red}} = 0)$

Algorithm 1 can answer the following questions:

Is $\mathcal{L}$ a Lie algebra? That is: is L a Lie VFS? Compute Commutes(L, L, L).
Is $\mathcal{L}$ abelian? Compute Commutes$(L, L, \{0\})$.
Is $\mathcal{M}$ an ideal in $\mathcal{L}$? Assuming L is a Lie VFS, evaluate the boolean

$$\mathrm{IsSubspace}(M,\ L)\ \textbf{and}\ \mathrm{Commutes}(L,\ M,\ M)$$

System Sys_N need not be a Riquier basis. Note that the method does not require resolution with respect to bases of $\mathcal{L}$, $\mathcal{M}$, $\mathcal{N}$.

Example 3.1. Let L be the VFS of Example 2.10 with the affine algebra A^2 as its solution vector fields. We use Algorithm 1 in the form Commutes(L, L, L) to check whether L is a Lie VFS. First, take two copies of vector fields $V := \xi^i \frac{\partial}{\partial x^i}$, $V' := \xi'^i \frac{\partial}{\partial x^i}$ and compute their Lie bracket $[V, V']$ (7)

$$[V, V'] = \big(\xi\xi'_x + \eta\xi'_y - \xi'\xi_x - \eta'\xi_y\big)\frac{\partial}{\partial x} + \big(\xi\eta'_x + \eta\eta'_y - \xi'\eta_x - \eta'\eta_y\big)\frac{\partial}{\partial y} \tag{14}$$

Applying operator $\Xi_L[V] = (\xi_{xx}, \eta_{xx}, \xi_{xy}, \eta_{xy}, \xi_{yy}, \eta_{yy})$ gives expressions R which when reduced mod (11), (11)$'$ gives $R_{\mathrm{red}} = [0, 0, 0, 0, 0, 0]$. Since all values in R_{red} are 0, L is indeed a Lie VFS. Thus, A^2 forms a Lie algebra, a fact which is obvious from the explicit form (10), but less so from VFS (11).

3.3. Transporter

We continue the notation of §3.2, and define the transporter $T_L(M, N)$ as being a VFS whose solution space is $T_{\mathcal{L}}(\mathcal{M}, \mathcal{N})$ (2). We exhibit an algorithm for finding $T_L(M, N)$. Thus we seek conditions on V_L such that $[V_L, V_M] \in \mathcal{N}$, that is, $\Xi_N\big([V_L, V_M]\big) = 0$ for all $V_M \in \mathcal{M}$. Reducing the expressions $\Xi_N\big([V_L, V_M]\big)$ modulo determining system Sys_M gives expressions $\Gamma_j^\alpha[V_L]\,\mathrm{Par}_M^j[V_M]$, with Γ_j^α explicitly known LHPDOs. These expressions are to vanish for all $V_M \in \mathcal{M}$, and since Sys_M is supposed a Riquier basis, this amounts to vanishing for all values of $\mathrm{Par}_M^j[V_M]$. Hence V_L must satisfy the LHPDEs $\Gamma_j^\alpha[V_L] = 0$. These are the additional conditions required so that V_L lie in $T_L(M, N)$; adjoining them to system Sys_L yields the required VFS. Note that systems Sys_L, Sys_N need not be Riquier bases.

Algorithm 2 Transporter in L of M to N.

Algorithm TRANSPORTER(L, M, N)
INPUT: VFS L, M, N, all on the same space
OUTPUT: VFS for transporter $T_L(M, N)$
$R := \Xi_N\big([V_L, V_M]\big)$
$R_{\mathrm{red}} := \mathrm{Reduce}(R)\quad \mathrm{mod}\ \mathrm{Sys}_M$
$\mathrm{NewEqs} := \mathrm{coeffs}(R_{\mathrm{red}}, \mathrm{Par}_M)$
return $\mathrm{VFS}(V_L, \mathrm{RiquierBasis}(\mathrm{Sys}_L \cup \mathrm{NewEqs}))$

Algorithm 2 can perform the following constructions, assuming L is a Lie VFS:

Centre of $\mathcal{L}$. That is: find a VFS for the ideal of vector fields in $\mathcal{L}$ that commute with $\mathcal{L}$. This amounts to finding a VFS for $T_L(L, \{0\})$. Compute Transporter$(L, L, \{0\})$.

Upper central series of $\mathcal{L}$. Let C_jL be a VFS with solution LAVF $C_j\mathcal{L}$. Find a sequence of VFS C_0L, $C_1L, \ldots, C_kL$ such that $C_0L = \{0\}$ and where $C_{j+1}\mathcal{L}$ commutes with $\mathcal{L}$ modulo $C_j\mathcal{L}$. (Note that C_1L is a VFS for the centre of $\mathcal{L}$.) Compute $C_{j+1}L := \text{Transporter}(L, L, C_jL)$.

Is $\mathcal{L}$ nilpotent? That is: decide if $C_k\mathcal{L}$ is equal to $\mathcal{L}$. Decide if $C_kL = L$.

Normaliser of $\mathcal{M}$ in $\mathcal{L}$. That is, find a VFS for the vector fields in $\mathcal{L}$ that commute with those in $\mathcal{M}$ modulo $\mathcal{M}$. Compute $\text{Transporter}(L, M, M)$.

Centraliser of $\mathcal{M}$ in $\mathcal{L}$. That is, find a VFS for the vector fields in $\mathcal{L}$ that commute with those in $\mathcal{M}$. Compute $\text{Transporter}(L, M, \{0\})$.

The standard definition (Jacobson, 1962; Bourbaki, 1989) of upper central series is phrased in terms of quotient Lie algebras, which is inconvenient to our purposes; the above construction is equivalent (Bourbaki, 1989). Normalisers are of interest in symmetry analysis (Ovsiannikov, 1982, §20.4 or Pommaret, 1978, §7.6).

Example 3.2. Let L be the VFS of Example 2.9 with solution LAVF $\mathcal{L}$. We construct VFS for the centre of $\mathcal{L}$ as $Z_1 := \text{Transporter}(L, L, \text{Triv})$, where Triv is the trivial VFS with determining system $\xi = 0$, $\eta = 0$ and differential operator $\Xi_0[V] = (\xi, \eta)$. Taking two vector fields V, V' (8), computing Lie bracket (7) and applying operator Ξ_0 gives the expressions

$$R = [\, \xi\xi'_x + \eta\xi'_y - \xi_x\xi' - \xi_y\eta', \;\; \xi\eta'_x + \eta\eta'_y - \eta_x\xi' - \eta_y\eta' \,]$$

Reduction of R mod (8)$'$ is here trivial, so $R_{\text{red}} = R$. Picking off coefficients of the parametric derivatives (9)$'$ (ξ', η', η'_x) gives the equations

$$\text{NewEqs} = [\, -\xi_x = 0, \;\; -\xi_y = 0, \;\; \xi = 0, \;\; -\eta_x = 0, \;\; -\eta_y = 0 \,]$$

Adjoining NewEqs to (8) and completing to a Riquier basis gives the VFS

$$V = \xi\frac{\partial}{\partial x} + \eta\frac{\partial}{\partial y}, \quad \text{where } \xi = 0, \;\; \eta_x = 0, \;\; \eta_y = 0 \tag{15}$$

satisfied by the centre. Thus, the centre is spanned by $\frac{\partial}{\partial y}$ as expected.

Example 3.3. Now we compute the second centre of this VFS, that is, $Z_2 := \text{Transporter}(L, L, Z_1)$. Again take Lie bracket $[V_L, V'_L]$, but now apply the operator Ξ_{Z_1} associated with system (15) for the centre Z_1. Reduction mod (8)$'$ gives the expressions R_{red}

$$-\xi_{xx}\xi' - \xi_{xy}\eta' - \xi_y\eta'_x, \;\; -\xi_{xy}\xi' - \xi_{yy}\eta', \;\; \xi_x\eta'_x - \eta_{xx}\xi' - \eta_{xy}\eta' - \eta_y\eta'_x, \;\; \xi_y\eta'_x - \eta_{xy}\xi' - \eta_{yy}\eta'$$

and picking off coefficients of (ξ', η', η'_x) gives LH conditions on ξ, η which however are consequences of (8). Thus no conditions beyond (8) arise for Z_2, which is therefore found to be all of L.

The upper central series for $\mathcal{L}$ is thus given by the sequence of VFS $\{0\} \subseteq (15) \subseteq (8)$, of dimensions 0, 1, 3 respectively. Because the upper central series terminates with L itself, we conclude that $\mathcal{L}$ is nilpotent.

The above method for finding the centre was essentially described by Boulton (1993, Appendix B). Note that the transporter Algorithm 2 has been devised so that it does not require resolution with respect to bases of $\mathcal{L}$, $\mathcal{M}$, $\mathcal{N}$, nor knowledge of c^k_{ij}. There is some latitude in the amount of reduction used in Algorithm 2: for instance, it is reasonable to additionally reduce $\Xi_N[V_L, V_M]$ modulo Sys_L.

3.4. Structure coefficients

Continuing the notation of the last two subsections, we give a generalisation of the method of Reid et al. (1992) for finding structure constants of a Lie algebra of vector fields from its Lie VFS. In fact we consider a more general question (compare §2.1.1), namely given VFS L, M, N such that $[L, M] \subseteq N$, find 'structure coefficients' f^k_{ij} satisfying (1). This and subsequent algorithms require calculation with respect to a basis $\{X_{Li}\}_{i=1}^{r_L}$ of the local solution space $\mathcal{L}_{x_0}$ of L near initial data point x_0. The dual basis of $\mathcal{L}^*_{x_0}$ will be denoted $\{\phi^i_L\}_{i=1}^{r_L}$, so that $\phi^i_L(X_{Lj}) = \delta^i_j$. Note that although our notations for basis vector fields and dual basis do not explicitly indicate base point x_0, this is always implied. It is convenient to rewrite (1) in terms of dual bases, as

$$\phi^k_N\big([V_L, V_M]\big) = f^k_{ij}(x_0)\,\phi^i_L(V_L)\,\phi^j_M(V_M), \quad \text{for all } V_L \in \mathcal{L}_{x_0}, V_M \in \mathcal{M}_{x_0} \tag{16}$$

Since the determining systems Sys_L, Sys_M, Sys_N are assumed to be Riquier bases, the parametric derivative evaluation functionals $\phi^i_L = \mathrm{Par}^i_{L,x_0}$ (respectively M, N) are suitable dual bases. The local solution bases (e.g. X_{Li}) induced by this choice of dual basis satisfy e.g. $\mathrm{Par}^k_{L,x_0}(X_{Li}) = \delta^k_i$. This gives Algorithm 3.

Algorithm 3 Structure coefficients f^k_{ij} for $[V_L, V_M]$ taking values in $\mathcal{N}$.

Algorithm StructureCoeffs(L, M, N)
 Input: VFS L, M, N, all on the same space satisfying Commutes(L, M, N)
 Output: Structure coefficients f^k_{ij} satisfying (16)
  $R := \mathrm{Par}_N\big([V_L, V_M]\big)$
  $R_{\mathrm{red}} := \mathrm{Reduce}(R) \mod \{\mathrm{Sys}_L, \mathrm{Sys}_M\}$
  for each (i, j, k)
   $f^k_{ij} := \mathrm{coeff}(R^k_{\mathrm{red}}, \mathrm{Par}^i_L(V_L) \cdot \mathrm{Par}^j_M(V_M))$
 return $\{f^k_{ij}\}$

Note that the 'structure coefficients' f^k_{ij} may vary as a function of base point x_0. The most important case is the structure constants c^k_{ij} of a Lie VFS L, which are found by StructureCoeffs(L, L, L). The resulting equation

$$\mathrm{Par}^k_{L,x_0}[V_L, V'_L] = c^k_{ij}(x_0)\,\mathrm{Par}^i_{L,x_0}(V_L)\,\mathrm{Par}^j_{L,x_0}(V'_L) \tag{17}$$

is essentially that of Reid et al. (1992). By careful management of a more general solution basis (16), Lisle et al. (2014) were able to extend these results to the case where the Lie VFS has been fully or partially integrated.

Example 3.4 *(A^2).* Continuing Example 3.1. Take two vector fields V, V' for the affine algebra A^2 (11) with parametric derivatives (12). The expressions for the parametric derivatives $\big\{\mathrm{Par}^k[V, V']\big\}_{k=1}^6$ after reduction (17) are respectively

$$\begin{aligned}
&\xi\xi'_x - \xi_x\xi' + \eta\xi'_y - \xi_y\eta', \quad \xi\eta'_x - \eta_x\xi' + \eta\eta'_y - \eta_y\eta', \\
&\eta_x\xi'_y - \xi_y\eta'_x, \quad \xi_y\xi'_x - \xi_x\xi'_y + \eta_y\xi'_y - \xi_y\eta'_y, \\
&\xi_x\eta'_x - \eta_x\xi'_x + \eta_x\eta'_y - \eta_y\eta'_x, \quad \xi_y\eta'_x - \eta_x\xi'_y
\end{aligned} \tag{18}$$

Reading off coefficients of the bilinear products $(\xi\xi', \xi\eta', \ldots)$ of parametrics (12) as per Algorithm 3 gives commutation relations

$$\begin{aligned}
&[X_1, X_3] = X_1, \quad [X_1, X_5] = X_2, \quad [X_2, X_4] = X_1, \quad [X_2, X_6] = X_2, \\
&[X_3, X_4] = -X_4, \quad [X_3, X_5] = X_5, \quad [X_4, X_5] = -X_3 + X_6, \\
&[X_4, X_6] = -X_4, \quad [X_5, X_6] = X_5
\end{aligned} \tag{19}$$

3.4.1. Adjoint representation, Killing form

With structure constants c^k_{ij} available, the matrix of the adjoint representation §2.1.4 with respect to the initial data basis are immediately available from (17) by picking off coefficients of $\mathrm{Par}_L(V'_L)$. Similarly, the matrix $K_{ij} = K(X_i, X_j)$ of the Killing form §2.1.4 is $K_{ij} = c^k_{il}c^l_{jk}$, giving the value at base point x_0

$$K(V,V') = K_{ij}(x_0)\,\mathrm{Par}^i_{x_0}(V)\,\mathrm{Par}^j_{x_0}(V') \tag{20}$$

Although the matrix entries K_{ij} are basis-dependent, the value $K(V,V')$ is not; in particular it does not depend on base point x_0, despite appearances.

Example 3.5. Continuing Example 3.4, the value of the Killing form is found to be

$$K(V,V') = 3\xi_x\xi'_x + 3\eta_y\eta'_y - 2\xi_x\eta'_y - 2\eta_y\xi'_x + 5\xi_y\eta'_x + 5\eta_x\xi'_y \tag{21}$$

This is to be interpreted as follows: if V, V' are two local solutions of (11), carrying out the indicated differentiations on the components of these vector fields gives the value of the Killing form.

3.5. Lie product

Continuing the notation above, we again suppose L, M, N to be VFS such that $[L,M] \subseteq N$ (13). We seek a VFS for the Lie product $[L,M]$ by finding a basis for the annihilator of $[L,M]$ in N. More precisely, at each x_0, we seek the annihilator of $[\mathcal{L}_{x_0}, \mathcal{M}_{x_0}]$ in $\mathcal{N}_{x_0}$. Thus let $\phi = \alpha_k\,\mathrm{Par}^k_{N,x_0}$ be a linear functional on the local solution space $\mathcal{N}_{x_0}$, and require that ϕ annihilate all commutators $[V_L, V_M]$. This amounts to the conditions $\alpha_k\,\mathrm{Par}_{N,x_0}[X_{Li}, X_{Mj}] = 0$ for each i, j, which by (1) gives the linear equations

$$f^k_{ij}(x_0)\alpha_k = 0, \qquad i = 1,\ldots,r_L, \quad j = 1,\ldots,r_M \tag{22}$$

Picking a basis for the nullspace then gives the required annihilator at x_0. The conditions to be in the annihilator at each x_0 are additional LHPDE to be satisfied beyond those in Sys_N; adjoining them to Sys_N and completing will give a VFS for $[L,M]$. This gives Algorithm 4.

Algorithm 4 Lie product $[L,M]$ of L, M in N.

Algorithm LieProduct(L,M,N)
Input: VFS L, M, N, all on the same space satisfying Commutes(L,M,N)
Output: VFS for Lie product $[L,M]$ in N
$f^k_{ij} :=$ StructureCoeffs(L,M,N)
Let $\{v^1,\ldots,v^l\}$ basis of solution space of $f^k_{ij}\alpha_k = 0$
NewEqs $:= \{v^i_k\,\mathrm{Par}^k_N(V_N) = 0, i = 1..l\}$
return VFS(V_N, RiquierBasis($\mathrm{Sys}_N \cup$ NewEqs))

Algorithm 4 can perform the following constructions, assuming L is a Lie VFS:

Derived algebra of $\mathcal{L}$. That is: find a VFS DL for the ideal of vector fields in $\mathcal{L}$ generated by $[\mathcal{L},\mathcal{L}]$.
Compute $DL :=$ LieProduct(L,L,L).
Is $\mathcal{L}$ abelian? Decide if $DL = \{0\}$.
Derived series of $\mathcal{L}$. Iterate the above construction. Find the sequence of VFS $L, D^1L,\ldots,D^kL$, until it stabilises.
Compute $D^{j+1}L :=$ LieProduct(D^jL, D^jL, D^jL).
Is $\mathcal{L}$ solvable? Decide if $D^kL = \{0\}$.
Lower central series of $\mathcal{L}$. This is the series $C^1\mathcal{L} := \mathcal{L},\ldots,C^k\mathcal{L} := [\mathcal{L}, C^{k-1}\mathcal{L}]$. It is easy to show that $[\mathcal{L}, C^j\mathcal{L}] \subseteq C^j\mathcal{L}$. Let C^jL be a VFS whose solution space is $C^j\mathcal{L}$. Compute $C^{j+1}L :=$ LieProduct(L, C^jL, C^jL).
Is $\mathcal{L}$ nilpotent? Decide if $C^kL = \{0\}$.

We note that although our method relies on choosing bases for the solution spaces of L, M, N, this is only for the purpose of intermediate calculation. The VFS for $[L,M]$ that results is basis-free.

Example 3.6 *(A^2).* Continuing Example 3.4, let L be the VFS for the affine algebra A^2, and seek its derived algebra $[L,L]$ as a subspace of L. Letting $\phi = \sum_{k=1}^{6}\alpha_k\,\mathrm{Par}^k_{x_0}$ (with $\mathrm{Par}^k_{x_0}$ the functionals implied by (12)), and using the structure constants exhibited in (18) or (19), equations (22) become $\alpha_1 = \alpha_2 = \alpha_4 = \alpha_5 = 0$, $\alpha_6 - \alpha_3 = 0$. The nullspace is thus 1-dimensional, and we find one additional equation $\mathrm{NewEqs} := \{\xi_x + \eta_y = 0\}$. Adjoining this to determining system (11) and completing to a Riquier basis gives the derived algebra of A^2 as the VFS

$$V = \xi\frac{\partial}{\partial x} + \eta\frac{\partial}{\partial y} \quad \text{where } \xi_{xx} = 0,\ \xi_{xy} = 0,\ \xi_{yy} = 0,\ \eta_{xx} = 0,\ \eta_y = -\xi_x \tag{23}$$

The derived algebra of A^2 is thus of dimension 5, as expected from the explicit form (10) of the operators.

3.6. Killing orthogonal space

Consider a Lie VFS L with solution LAVF $\mathcal{L}$, let M be a VFS (with solution space $\mathcal{M}$) that specifies a subspace of $\mathcal{L}$ and suppose the Killing form of $\mathcal{L}$ (20) has been found. We show how to compute a VFS for the subspace Killing orthogonal to $\mathcal{M}$ (3). Reducing (20) modulo the determining system for $V_M \in \mathcal{M}$ gives

$$K(V_L, V_M) = h_{ij}(x)\,\mathrm{Par}^i_L(V_L)\,\mathrm{Par}^j_M(V_M)$$

where the coefficients h_{ij} are explicitly known. This is to vanish for all values of $\mathrm{Par}^j_M(V_M)$, giving the differential equations

$$h_{ij}(x)\,\mathrm{Par}^i_L(V_L) = 0$$

Adjoining these to the original determining system Sys_L gives the determining system for the Killing orthogonal subspace, justifying Algorithm 5 below.

Algorithm 5 K-orthogonal space to M in L.

```
Algorithm KOrthogonalSpace(M, L)
  Input: VFS M, Lie VFS L, satisfying IsSubspace(M, L)
  Output: VFS for the subspace Killing orthogonal to M in L
  K := KillingForm(L)
  K_red := Reduce(K(V_L, V_M)) mod Sys_M
  NewEqs := coeffs(K_red, Par_M(V_M))
  return VFS(V_L, RiquierBasis(Sys_L ∪ NewEqs))
```

Algorithm 5 allows the following constructions:

Solvable radical of $\mathcal{L}$. According to Cartan's criterion (Jacobson, 1962; Bourbaki, 1989) the solvable radical of $\mathcal{L}$ (i.e. its largest solvable ideal) is the K-orthogonal of the derived algebra $D\mathcal{L}$. Compute $\mathrm{KOrthogonal}(DL, L)$ where $DL := \mathrm{LieProduct}(L, L, L)$.

Nilpotent radical of $\mathcal{L}$. Let $SR(\mathcal{L})$ be the solvable radical of $\mathcal{L}$ and $D\mathcal{L}$ its derived algebra. Then according to Bourbaki (1989), the nilpotent radical of $\mathcal{L}$ can be found by $SR(\mathcal{L}) \cap D\mathcal{L}$. Compute $\mathrm{Intersection}(SR, DL)$ where $SR := \mathrm{KOrthogonal}(DL, L)$ and $DL := \mathrm{LieProduct}(L, L, L)$.

Example 3.7. Continuing Example 3.6, we find the solvable radical of the affine algebra A^2 (11), as the space Killing-orthogonal to its derived algebra DA^2 (23). Reducing the value $K(V, V')$ (21) of the Killing form of A^2 modulo system $(23)'$ for DA^2 (which amounts to substituting $\eta'_y = -\xi'_x$), gives

$$K(V, V') = 5\xi_x\xi'_x - \eta_y\xi'_x + 5\eta_x\xi'_y + 5\xi_y\eta'_x$$

Picking off coefficients of the parametric derivatives $(\xi', \eta', \xi'_x, \xi'_y, \eta'_x)$ of DA^2 (23) gives the new equations

$$\xi_x - \eta_y = 0, \quad \eta_x = 0, \quad \xi_y = 0$$

Appending these to the determining system for A^2 (11) and completing gives the VFS for the solvable radical $SR(A^2)$:

$$V = \xi \frac{\partial}{\partial x} + \eta \frac{\partial}{\partial y}, \quad \text{where } \xi_{xx} = 0, \ \ \xi_y = 0, \ \ \eta_x = 0, \ \ \eta_y = \xi_x$$

The solvable radical is thus 3-dimensional, as expected from the explicit form (10); it is spanned by the vector fields $\frac{\partial}{\partial x}$, $\frac{\partial}{\partial y}$, $x\frac{\partial}{\partial x} + y\frac{\partial}{\partial y}$.

4. Examples and applications

4.1. Example

Consider the VFS L for the point symmetries

$$V = \tau(t)\frac{\partial}{\partial t} + \rho(r,\theta,t)\frac{\partial}{\partial r} + \phi(r,\theta,t)\frac{\partial}{\partial \theta} + \eta(r,\theta,t,u)\frac{\partial}{\partial u}$$

of the $2+1$ linear heat equation $u_t = \frac{1}{r}(ru_r)_r + \frac{1}{r^2}u_{\theta\theta}$ in cylindrical polar coordinates. Suppressing the infinite 'superposition' symmetries, these obey determining system Sys_L with Riquier basis

$$\begin{aligned}
&\tau_{ttt} = 0, \ \ \rho_{tt} = 0, \ \ \rho_{t\theta} = r\phi_t, \ \ \rho_{\theta\theta} = \tfrac{1}{2}r\tau_t - \rho, \ \ \rho_r = \tfrac{1}{2}\tau_t, \\
&\phi_{tt} = 0, \ \ \phi_r = -\tfrac{1}{r^2}\rho_\theta, \ \ \phi_\theta = -\tfrac{1}{r}\rho + \tfrac{1}{2}\tau_t, \\
&\eta_t = -\tfrac{1}{2}u\tau_{tt}, \ \ \eta_r = -\tfrac{1}{2}u\rho_t, \ \ \eta_\theta = -\tfrac{1}{2}r^2u\phi_t, \ \ \eta_u = \tfrac{1}{u}\eta
\end{aligned} \tag{24}$$

The parametric derivatives are $(\tau, \tau_t, \tau_{tt}, \rho, \phi, \rho_t, \phi_t, \rho_\theta, \eta)$, giving a 9-dim symmetry algebra $\mathcal{L}$. Using Algorithm 4, the derived algebra $D\mathcal{L}$ is found to satisfy additionally $\rho_\theta = r\phi$, so that it is 8-dim. The Killing form of $\mathcal{L}$ is found to be (20)

$$K(V, V') = 3\tau_t\tau_t' - 3(\tau\tau_{tt}' + \tau_{tt}\tau') - 4\phi\phi' + \tfrac{4}{r}(\phi\rho_\theta' + \rho_\theta\phi') - \tfrac{4}{r^2}\rho_\theta\rho_\theta'$$

The solvable radical $SR(\mathcal{L})$ is the K-orthogonal of $D\mathcal{L}$, and satisfies (24) and the additional condition $\tau = 0$, so that it is the 6-dim solution space of

$$\begin{aligned}
&\tau = 0, \ \ \rho_{tt} = 0, \ \ \rho_{t\theta} = r\phi_t, \ \ \rho_{\theta\theta} = -\rho, \ \ \rho_r = 0, \ \ \phi_{tt} = 0, \ \ \phi_r = -\tfrac{1}{r^2}\rho_\theta, \\
&\phi_\theta = -\tfrac{1}{r}\phi, \ \ \eta_t = 0, \ \ \eta_r = -\tfrac{1}{2}u\rho_t, \ \ \eta_\theta = -\tfrac{1}{2}r^2u\phi_t, \ \ \eta_u = \tfrac{1}{u}\eta
\end{aligned} \tag{25}$$

In turn, the derived algebra of $SR(\mathcal{L})$ enforces the additional equation $\rho_\theta = \phi$ and is 5-dim. Indeed the derived series of $SR(\mathcal{L})$ gives VFS of dimensions $[6, 5, 1, 0]$. The penultimate 1-dim VFS turns out to be the same as the centre of $\mathcal{L}$ (and of $SR(\mathcal{L})$), and satisfies determining system

$$\tau = 0, \ \ \rho = 0, \ \ \phi = 0, \ \ \eta_t = 0, \ \ \eta_r = 0, \ \ \eta_\theta = 0, \ \ \eta_u = \tfrac{1}{u}\eta$$

Note that answers such as above can be extracted even though the heat equation is written in a somewhat adverse coordinate system, as they require only differentiation and elimination. Moreover, no field extension is needed, whereas solving systems (24), (25) requires introduction of transcendental functions, for example $-\sin\theta\, t\frac{\partial}{\partial r} - \frac{\cos\theta}{r}t\frac{\partial}{\partial \theta} + \frac{1}{2}r\sin\theta\, u\frac{\partial}{\partial u}$.

4.2. Integrals of determining system

With the methods described in this paper (see also Reid et al., 1992), the structure constants c_{ij}^k and thus the matrix of the adjoint representation of VFS L may depend on the base point x_0 about which local solutions are constructed. This is however an artifact of the method: the Lie algebras of local solutions $\mathcal{L}_{x_0}$ have the same algebraic structure for all x_0; it is our choice of *basis* for $\mathcal{L}_{x_0}$ which is varying. A quantity that is independent of basis will therefore be constant for all x_0. This fact can lead to construction of integrals of a determining system without in fact integrating.

Example 4.1. Consider the Lie VFS

$$V = \eta(y)\frac{\partial}{\partial y} \quad \text{where } \eta_{yy} = \frac{4y}{y-3}\eta_y - \frac{2}{(y-3)^2}\eta, \tag{26}$$

a system slightly modified from ‘Example A’ of Reid et al. (1992). The parametric derivatives are η, η_y, and with a standard initial data basis one finds the commutation relation

$$[X_1, X_2] = -X_1 - \frac{4y}{y-3}X_2$$

so the Lie algebra is 2-dim non-abelian. In this basis the matrix of ad V is

$$\begin{bmatrix} -\eta_y & \eta \\ \frac{-4y}{y-3}\eta_y & \frac{4y}{y-3}\eta \end{bmatrix}$$

and is y-dependent. However the trace of this ad-matrix is independent of basis, and is therefore a constant. We thus conclude that

$$\eta_y - \frac{4y}{y-3}\eta = c$$

is an integral of (26), obtained without applying any integration methods.

This observation obviously applies to any Lie VFS with a 2-dim non-abelian structure, and indeed to any Lie VFS where the adjoint representation has non-zero trace.

More generally, we could proceed as follows. From the Lie VFS L, find the Lie VFS R for the solvable radical of $\mathcal{L}$. Classical results (Gilmore, 2005; Schwarz, 2008) show that the characteristic polynomial of a solvable Lie algebra splits into linear factors. The roots of the characteristic polynomial of R are then basis-independent expressions, linear homogeneous in parametric derivatives; they are therefore integrals of the determining system. The number of integrals obtained is the number of such roots that are linearly independent, i.e. the ‘rank’ of R (Gilmore, 2005). For example, the characteristic polynomial of the solvable radical for the symmetry algebra of the $2+1$ heat equation (25) has root $\frac{1}{r}\rho_\theta - \phi$; it follows that $\frac{1}{r}\rho_\theta - \phi = c$ is a first integral of both (24), (25).

5. Discussion

The algorithms provided here give access to two kinds of information about a Lie symmetry algebra. Algorithm 3 gives structure constants c_{ij}^k and thence isomorphism invariants as described by de Graaf (2000). It is also possible to extract geometric information (i.e. diffeomorphism invariants) from a Lie VFS L: quantities such as orbit dimension, transitivity, and structure of isotropy algebra are available (Lisle and Reid, 1998; Draisma, 2001; Schwarz, 2008). Since our Algorithms 2, 4, 5 are able to extract Lie VFS for structural parts of L, this means that one now has access to a wider range of diffeomorphism invariants. For instance, the orbit dimension of the derived algebra, or the structure of the isotropy algebra of the solvable radical are now algorithmically available invariants.

Algebraic and geometric structure of Lie symmetries are required in a number of applications. For example, Lie's method of integrating ODE (Bluman and Anco, 2002; Olver, 1993) uses solvable symmetry algebras. Computer algebra implementation of such methods requires as one step the identification of the ‘group type’ of the symmetry algebra, i.e. its classification up to diffeomorphism. Draisma (2001) gives one approach to finding group type, which is, however limited to transitive symmetry group actions. Schwarz (2008) gives a method based on detailed analysis of the determining system, and which applies also to the intransitive case; however it is specific to transformations of the (x, y) plane. The toolkit provided in this paper gives more general machinery for attacking this problem.

The fact that we are computing with vector fields gives our treatment a different emphasis to algebraic algorithms for Lie algebras (de Graaf, 2000). There, structure constants c_{ij}^k with respect to a basis abstractly given are the starting point, and a commutator bracket is effectively defined by the c_{ij}^k. For us, the starting point is a vector field system, and a bracket operation is available a priori as (7). Algorithms 1 and 2 (checking commutation mod N, transporter) work directly with Lie brackets and determining systems, and do not resolve the solution space with respect to a basis at any stage. These algorithms are therefore quite distinct to the algebraic approach above. On the other hand, Algorithm 3 calculates structure constants c_{ij}^k with respect to a basis of solution vector fields which is implicitly determined, via a choice of dual basis. As observed by Reid et al. (1992), its output c_{ij}^k provides the inputs to standard Lie-algebraic algorithms. Our Algorithms 4, 5 for Lie product and Killing orthogonal subspace rely on these structure constants (and hence choice of basis), but only for purposes of intermediate calculation: the answers they return are basis-free. Nevertheless the internal working of these algorithms mirrors that described in de Graaf (2000). Our Algorithms 4, 5 are, however, phrased in terms of annihilators rather than bases, because annihilator conditions are realised as explicitly constructable LHPDE, whereas bases for the solution space are not explicitly constructable.

We have already extended the work described in this paper in certain directions. First, we have implemented our algorithms in a general purpose computer algebra package for dealing with vector fields, LHPDE and VFS. The implementation is engineered using the object-oriented design model of Maple 16 and later (Bernardin et al., 2013) and will be described elsewhere.[2] Note that our approach differs from existing Maple packages such as `DifferentialGeometry` (Anderson and Torre, 2012). Second, as has frequently been noted (e.g. Olver, 1993), Lie determining systems can often be integrated (or part-integrated) explicitly, so it is desirable to have methods that apply equally to VFS containing constants (or functions) of integration. Lisle et al. (2014) describe such a generalisation of Algorithm 3 for structure coefficients. In fact our Maple package implements generalisations of all Algorithms 1–5 to the part-integrated case.

Further extensions of our work may be possible. First, our treatment does not deal with quotient algebras. Although quotients are clearly defined algebraic objects, the notion of a quotient pseudogroup is more subtle. Essentially one should quotient not by an ideal, but by an invariant foliation (Kuranishi and Rodrigues, 1964; D'Atri, 1965). Such a process requires methods associated with infinite Lie pseudogroups. Second, it is desirable that tools for Lie pseudogroups work not just for systems of finite type, but also for infinite Lie pseudogroups (Singer and Sternberg, 1965; Stormark, 2000; Lisle and Reid, 1998; Olver et al., 2009), although care must be taken with the relationship between an infinite Lie pseudogroup and its infinitesimal determining system (Pommaret, 1978, §7). Note that as Algorithms 1 and 2 (commutation mod N, transporter) do not use structure constants, and are not referred to a basis of the solution Lie algebra, they can be carried out for VFS of both finite and infinite type. Thus centre, normalisers etc. are algorithmically constructible for infinite Lie pseudogroups.

Acknowledgements

IGL wishes to thank Alan Boulton and Greg Reid for discussions which stimulated this work. TH wishes to thank her supervisor IGL for passing down his wisdom and research. Despite his illness and severe side effects from treatments, Ian always worked extremely hard on this research. His determination has truly shown that it is possible to achieve anything. The authors would like to dedicate this paper to the memory of Mary Hewett, who provided us encouragement and support over many years.

References

Adams, W., Loustaunau, P., 1994. An Introduction to Gröbner Bases. AMS.

Anderson, I.M., Torre, C.G., 2012. New symbolic tools for differential geometry, gravitation, and field theory. J. Math. Phys. 53. http://dx.doi.org/10.1063/1.3676296.

[2] For code, help pages and demonstration sheets, see http://www.canberra.edu.au/research/faculty-research-centres/msrc/projects/lavf.

Bernardin, L., et al., 2013. Maple Programming Guide. Maplesoft.
Blinkov, Y.A., Cid, C.F., Gerdt, V.P., Plesken, W., Robertz, D., 2003a. The MAPLE package Janet: I. Polynomial systems. In: Proc. 6th Int. Workshop on Computer Algebra in Scientific Computing, pp. 31–40.
Blinkov, Y.A., Cid, C.F., Gerdt, V.P., Plesken, W., Robertz, D., 2003b. The MAPLE package Janet: II. Linear partial differential equations. In: Proc. 6th Int. Workshop on Computer Algebra in Scientific Computing, pp. 41–54.
Bluman, G., Anco, S., 2002. Symmetry and Integration Methods for Differential Equations. Springer.
Bluman, G., Cheviakov, A., Anco, S., 2010. Applications of Symmetry Methods to Partial Differential Equations. Springer.
Boulier, F., Lazard, D., Ollivier, F., Petitot, M., 1995. Representation for the radical of a finitely generated differential ideal. In: Proc. ISSAC '95. ACM, pp. 158–166.
Boulton, A., 1993. New symmetries from old: exploiting Lie algebra structure to determine infinitesimal symmetries of differential equations. Master's thesis. Dept. of Mathematics, University of British Columbia, Canada.
Bourbaki, N., 1989. Lie Groups and Lie Algebras: Chapters 1–3. Springer.
Carminati, J., Vu, K., 2000. Symbolic computation and differential equations: Lie symmetries. J. Symb. Comput. 29, 95–116.
Castro-Jiménez, F., Moreno-Frías, M., 2001. An introduction to Janet bases and Gröbner bases. In: Hermida, J., Verschoren, A., Granja, A. (Eds.), Ring Theory And Algebraic Geometry. CRC Press, pp. 133–145.
Cheviakov, A., 2007. GeM software package for computation of symmetries and conservation laws of differential equations. Comput. Phys. Commun. 176, 48–61. http://dx.doi.org/10.1016/j.cpc.2006.08.001.
Cox, D., Little, J., O'Shea, D., 2005. Using Algebraic Geometry, 2nd ed. Springer.
D'Atri, J., 1965. Homomorphisms of continuous pseudogroups. Nagoya Math. J. 25, 143–163.
Draisma, J., 2001. Recognizing the symmetry type of O.D.E.s. J. Pure Appl. Algebra 164, 109–128.
Gerdt, V.P., 1999. Completion of linear differential systems to involution. In: Ganzha, V., Mayr, E., Vorozhtsov, E. (Eds.), Computer Algebra in Scientific Computing. CASC '99. Springer-Verlag, pp. 115–137.
Gilmore, R., 2005. Lie Groups, Lie Algebras, and Some of Their Applications. Dover.
de Graaf, W., 2000. Lie Algebras: Theory and Algorithms. North-Holland.
Jacobson, N., 1962. Lie Algebras. Dover.
Krasil'shchik, I., Vinogradov, A. (Eds.), 1999. Symmetries and Conservation Laws for Differential Equations of Mathematical Physics. AMS.
Kuranishi, M., Rodrigues, A., 1964. Quotients of pseudo groups by invariant fiberings. Nagoya Math. J. 24, 109–128.
Lemaire, F., 2003. An orderly linear PDE system with analytic initial conditions with a non-analytic solution. J. Symb. Comput. 35, 487–498. http://dx.doi.org/10.1016/S0747-7171(03)00017-8.
Lisle, I., Reid, G., 1998. Geometry and structure of Lie pseudogroups from infinitesimal defining systems. J. Symb. Comput. 26, 355–379.
Lisle, I.G., Huang, S.L.T., Reid, G.J., 2014. Structure of symmetry of PDE: exploiting partially integrated systems. In: Proceedings of the 2014 Symposium on Symbolic-Numeric Computation. ACM, pp. 61–69.
Mansfield, E., 1991. Differential Gröbner bases. Ph.D. thesis. University of Sydney.
Olver, P., 1993. Application of Lie Groups to Differential Equations, 2nd ed. Springer-Verlag, New York.
Olver, P., Pohjanpelto, J., Valiquette, F., 2009. On the structure of Lie pseudo-groups. SIGMA 5, 77–90. http://dx.doi.org/10.3842/SIGMA.2009.077.
Ovsiannikov, L., 1982. Group Analysis of Differential Equations. Academic Press.
Pommaret, J.F., 1978. Systems of Partial Differential Equations and Lie Pseudogroups. Gordon and Breach.
Reid, G., Lisle, I., Boulton, A., Wittkopf, A., 1992. Algorithmic determination of commutation relations for Lie symmetry algebras of PDEs. In: Proc. ISSAC '92. ACM Press, pp. 63–68.
Reid, G., Wittkopf, A., Boulton, A., 1996. Reduction of systems of nonlinear partial differential equations to simplified involutive forms. Eur. J. Appl. Math. 7, 604–635.
Riquier, C., 1910. Les Systémes d'Équations aux dÉrivées Partielles. Gauthier-Villars.
Rocha Filho, T., Figueiredo, A., 2011. [SADE]: A Maple package for the symmetry analysis of differential equations. Comput. Phys. Commun. 182, 467–476. http://dx.doi.org/10.1016/j.cpc.2010.09.021.
Rust, C., Reid, G., 1997. Rankings of partial derivatives. In: Proc. ISSAC '97, pp. 9–16.
Rust, C., Reid, G., Wittkopf, A., 1999. Existence and uniqueness theorems for formal power series solutions of analytic differential systems. In: Proc. ISSAC '99. ACM, pp. 105–112.
Schwarz, F., 1992a. An algorithm for determining the size of symmetry groups. Computing 49, 95–115. http://dx.doi.org/10.1007/BF02238743.
Schwarz, F., 1992b. Reduction and completion algorithms for partial differential equations. In: Proc. ISSAC '92. ACM Press, pp. 49–56.
Schwarz, F., 2008. Algorithmic Lie Theory for Solving Ordinary Differential Equations. Chapman and Hall/CRC Press.
Singer, I., Sternberg, S., 1965. The infinite groups of Lie and Cartan Part I, (The transitive groups). J. Anal. Math. 15, 1–114. http://dx.doi.org/10.1007/BF02787690.
Stormark, O., 2000. Lie's Structural Approach to PDE Systems. Cambridge Univ. Press.