

Symbolic Computation of Local Symmetries of Nonlinear and Linear Partial and Ordinary Differential Equations

Alexei F. Cheviakov,^{a)}

Department of Mathematics and Statistics, University of Saskatchewan, Saskatoon, S7N 5E6 Canada

September 19, 2011

Abstract

The paper illustrates the use of a symbolic software package `GeM` for `Maple` to compute local symmetries of nonlinear and linear differential equations (DE). In the cases when a given DE system contains arbitrary functions or parameters, symbolic symmetry classification is performed.

Special attention is devoted to the computation of point symmetries of linear PDE systems. Routines are available that effectively eliminate infinite obvious symmetries of linear differential equations.

1 Introduction

A symmetry of a system of ordinary or partial differential equations (ODE, PDE) is any transformation of its solution manifold into itself, i.e., a symmetry maps any solution of a DE system to another solution of the same system. Continuous symmetries of DE systems are hence defined topologically, and in principle, any nontrivial DE system has symmetries. The practical problem consists in finding algorithmic ways of computation of such symmetries. In the latter part of the 19th century, Sophus Lie showed that the problem of finding the Lie group of point transformations (*point symmetries*) leaving invariant an equation (algebraic, or ODE, or PDE), reduced to solving related linear systems of determining equations for components of its infinitesimal generators. Lie's algorithm for finding point symmetries of a DE system can be extended to find more general *local symmetries* admitted by DEs, where one allows the symmetry components to depend on derivatives of dependent variables.

The structure of local symmetry groups admitted by different ODE and PDE systems contains essential information about the given DE system, and thus can vary greatly. For example, every first-order scalar ODE has an infinite-parameter Lie group of point symmetries. A second-order scalar ODE has at most an eight-parameter Lie group of point symmetries; a scalar ODE of order $n \geq 3$ has at most $n + 4$ independent point symmetries (see [1], Section 3.7). PDEs and PDE systems can have finite- or infinite-parameter Lie groups of point transformations. In particular, linear PDE systems and PDE systems that can be mapped into linear PDE systems through an invertible transformation admit infinite-parameter Lie groups of point symmetries.

Symmetries of ordinary differential equations are used for reduction of order and complete integration, as well as for the construction of invariant solutions ([1], Chapter 3). Symmetries of partial differential equations yield reductions of order and/or number of variables. Invariant (in particular, self-similar) solutions that arise from reduced systems often have transparent physical meaning. Many appropriate examples are found in [2]. Infinite-dimensional symmetry groups are

^{a)}Corresponding author. Electronic mail: chevaikov@math.usask.ca

used for the construction of families of new exact solutions from known ones (e.g. [3]). For a nonlinear PDE system, from its admitted symmetry group, one can determine whether or not it can be mapped into a linear system by an invertible transformation, and find the explicit form of that transformation ([4], Chapter 2).

Many extensions of the notion of local symmetries are known, including nonlocal symmetries, approximate symmetries, the nonclassical method, etc. For a detailed review, the reader is referred to [2, 4, 5] and references therein.

Application of Lie's symmetry method and its extensions to nontrivial DE systems often requires extensive algebraic manipulation. For many contemporary DE models, especially those that do possess non-trivial symmetry structure, such analysis presents a significant computational challenge. In particular, symmetry determining equations split into overdetermined systems of linear PDEs for the unknown symmetry components; such linear overdetermined systems can commonly contain hundreds or thousands of determining equations. To treat such huge systems, symbolic computer software is normally used.

A symbolic package for computations of symmetries naturally consists of two parts. The first part contains user routines that interpret equations, use a specified ansatz for symmetry components or conservation law multipliers, generate determining equations, and split them into an overdetermined linear PDE system. The second part contains routines for effective symbolic reduction and solution of large overdetermined PDE systems. Since DE systems often involve arbitrary constitutive functions and/or parameters, it is highly beneficial when the reduction algorithm allows for case splitting, i.e., the isolation of special forms of constitutive functions and/or parameters, for which additional symmetries arise.

Several methods for reduction of large overdetermined systems of partial differential equations have been developed (for review, see e.g. [8–11].) Many of these methods have been implemented in various symbolic software packages. In particular, the excellent `rif` package of `Maple` based on the work in [17–19] is one of the core components of `Maple`'s symbolic differential equations toolbox.

To date, a number of packages for the computation of symmetries has been developed for different computer algebra systems (CAS). For example, one can name a set of programs `LiePDE`, `ApplySym` [6] that provide a user interface for local symmetry computation in CAS `REDUCE`; a set of programs for symmetry computations in `Mathematica` is described in [7]. For reviews of symbolic software for symmetry analysis and related problems, the reader is referred to [8–11] and [4], Chapter 5.

The package `GeM` for `Maple` described in this work offers a set of simple and general routines that generate and split the systems of determining equations for the computations of local conservation laws, symmetries and approximate symmetries for wide classes of differential equations and DE systems. It subsequently uses `rif` for case splitting and solution of overdetermined systems, and contains additional of routines for the symmetry output, as well as other important routines described in this paper.

The current paper explains and illustrates the use of the latest version of `GeM` package to perform symbolic computations of local symmetries of linear and nonlinear differential equations.

In Section 2, the notation and general algorithms for local symmetry analysis are briefly reviewed. A special computational difficulty is presented by systems of *linear PDEs*, since they admit an infinite set of obvious point symmetries, corresponding to addition of arbitrary solutions of the corresponding linear homogeneous system. In Section 2.3, theorems and a general conjecture pertaining to point symmetries of linear equations are presented and discussed.

In Section 3, the principal sequence of symmetry analysis using `GeM` package for `Maple` is reviewed, and some features of the package structure and its routines are highlighted.

Section 4 contains run examples for point symmetry analysis of a third-order nonlinear ODE and higher-order symmetry computations for the Korteweg-de Vries equation [25].

To study symmetries of DE systems involving arbitrary function(s) and/or constant parameter(s), a symmetry classification must be performed. Section 5 discusses this problem, the related problem of finding and using equivalence transformations, and their symbolic implementations. As an example, equivalence transformations are computed and point symmetries are classified for a family of a nonlinear diffusion equations.

Section 6 contains an example of symmetry analysis of a *linear* PDE system. The example employs GeM package routines which use a special ansatz discussed in Section 2.3, in order to automatically exclude obvious infinite symmetries of linear equations.

Section 7 concludes the paper with some discussion.

2 Local symmetries of differential equations

Consider a system of N differential equations of order k , with n independent variables $x = (x^1, \dots, x^n)$ and m dependent variables $u(x) = (u^1(x), \dots, u^m(x))$, given by

$$R^\sigma(x, u, \partial u, \dots, \partial^k u) = 0, \quad \sigma = 1, \dots, N. \quad (2.1)$$

[For ODE systems, $N = 1$.] The notation

$$\partial u \equiv \partial^1 u = \left(u_1^1(x), \dots, u_n^1(x), \dots, u_1^m(x), \dots, u_n^m(x) \right)$$

denotes the set of all first-order partial derivatives;

$$\begin{aligned} \partial^p u &= \left\{ u_{i_1 \dots i_p}^\mu \mid \mu = 1, \dots, m; \quad i_1, \dots, i_p = 1, \dots, n \right\} \\ &= \left\{ \frac{\partial^p u^\mu(x)}{\partial x^{i_1} \dots \partial x^{i_p}} \mid \mu = 1, \dots, m; \quad i_1, \dots, i_p = 1, \dots, n \right\} \end{aligned}$$

denote higher-order derivatives.

Summation in any pair of repeated indices is assumed throughout the paper.

2.1 Point symmetries

Consider a one-parameter Lie group of point transformations

$$\begin{aligned} (x^*)^i &= f^i(x, u; \varepsilon) = x^i + \varepsilon \xi^i(x, u) + O(\varepsilon^2), \quad i = 1, \dots, n, \\ (u^*)^\mu &= g^\mu(x, u; \varepsilon) = u^\mu + \varepsilon \eta^\mu(x, u) + O(\varepsilon^2), \quad \mu = 1, \dots, m, \end{aligned} \quad (2.2)$$

with the corresponding infinitesimal generator

$$X = \xi^i(x, u) \frac{\partial}{\partial x^i} + \eta^\mu(x, u) \frac{\partial}{\partial u^\mu}. \quad (2.3)$$

The k th extension (prolongation) of (2.3) is given by

$$\begin{aligned} X^{(k)} &= \xi^i(x, u) \frac{\partial}{\partial x^i} + \eta^\mu(x, u) \frac{\partial}{\partial u^\mu} + \eta_i^{(1)\mu}(x, u, \partial u) \frac{\partial}{\partial u_i^\mu} \\ &\quad + \dots + \eta_{i_1 \dots i_k}^{(k)\mu}(x, u, \partial u, \dots, \partial^k u) \frac{\partial}{\partial u_{i_1 \dots i_k}^\mu}, \end{aligned} \quad (2.4)$$

where the prolonged components $\eta_i^{(1)\mu}, \dots, \eta_{i_1 \dots i_k}^{(k)\mu}$ are defined in terms of $\{\xi^i(x, u), \eta^\mu(x, u)\}$ by

$$\eta_i^{(1)\mu} = D_i \eta^\mu - (D_i \xi^j) u_j^\mu, \quad (2.5)$$

and

$$\eta_{i_1 \dots i_k}^{(k)\mu} = D_{i_k} \eta_{i_1 \dots i_{k-1}}^{(k-1)\mu} - (D_{i_k} \xi^j) u_{i_1 \dots i_{k-1} j}^\mu, \quad (2.6)$$

for $\mu = 1, \dots, m$, and $i, i_j = 1, \dots, n$ for $j = 1, \dots, k$, and

$$D_i = \frac{\partial}{\partial x^i} + u_i^\mu \frac{\partial}{\partial u^\mu} + u_{ii_1}^\mu \frac{\partial}{\partial u_{i_1}^\mu} + u_{ii_1 i_2}^\mu \frac{\partial}{\partial u_{i_1 i_2}^\mu} + \dots, \quad (2.7)$$

$i = 1, \dots, n$.

Definition 2.1. A one-parameter Lie group of point transformations (2.2) leaves the DE system (2.1) invariant if and only if its k th extension (2.4) leaves invariant the solution manifold of (2.1) in $(x, u, \partial u, \dots, \partial^k u)$ -space, i.e., it maps any family of solution surfaces $u = u(x)$ of the DE system (2.1) into another family of solution surfaces $u^* = u^*(x^*)$ of DE system (2.1). In this case, the one-parameter Lie group of point transformations (2.2) is called a *point symmetry* of the DE system (2.1).

Lie's algorithm to find the point symmetries of a given DE system (2.1) is given by the following theorem.

Theorem 2.1. Let (2.3) be the infinitesimal generator of a one-parameter Lie group of point transformations (2.2). Let (2.4) be its k th extension. Then the transformation (2.2) is a point symmetry of the DE system (2.1) if and only if for each $\alpha = 1, \dots, N$,

$$X^{(k)} R^\alpha(x, u, \partial u, \dots, \partial^k u) = 0, \quad (2.8)$$

when

$$R^\sigma(x, u, \partial u, \dots, \partial^k u) = 0, \quad \sigma = 1, \dots, N. \quad (2.9)$$

The proof appears in [12], with the restriction that the given system (2.1) can be written in a solved form in terms of a set of leading derivatives.

In order to find point symmetries admitted by a given DE system (2.1), one needs to determine the unknown symmetry components ξ^i, η^μ that appear in the symmetry generator (2.3). The algorithm proceeds in the following steps.

1. Obtain *determining equations* by substituting the given DEs (2.9) (as well as, possibly, differential consequences of (2.9)) into the invariance condition (2.8) (in order to restrict to the solution manifold).
2. Obtain *the split system of determining equations*, using the fact that ξ^i, η^μ do not depend on derivatives of u , i.e., setting coefficients at all independent combinations of derivatives of dependent variables in determining equations to zero.

In practice, split systems of determining equations can be rather large. For example, for the system of adiabatic compressible plasma equilibrium equations, one obtains 188 linear PDEs for 13 unknown functions [13]. Split systems containing over a thousand determining equations are also not uncommon.

The evolutionary form

The point symmetry generator (2.3) can be equivalently rewritten in the *evolutionary (characteristic) form*

$$\hat{X} = \hat{\eta}^\mu(x, u) \frac{\partial}{\partial u^\mu} = [\eta^\mu(x, u) - u_i^\mu \xi^i(x, u)] \frac{\partial}{\partial u^\mu}. \quad (2.10)$$

Note that in the evolutionary form, the independent variables are invariant, and the action of a group of point transformations is strictly an action on the dependent variables of the DE system, so that solutions are directly mapped into other solutions under the group action.

In evolutionary form, the components of point symmetry infinitesimal generators for dependent variables have at most a linear dependence on the first derivatives of the dependent variables (the coefficients of the first derivatives are the components of the independent variables when not in evolutionary form).

2.2 Local symmetries

The notion of evolutionary (characteristic) form allows one to readily extend Lie's algorithm to seek *higher-order symmetries* (order q) by letting the infinitesimal generators in evolutionary form to depend on derivatives of dependent variables to any finite order $s \geq 1$:

$$\hat{X} = \hat{\eta}^\mu(x, u, \partial u, \dots, \partial^s u) \frac{\partial}{\partial u^\mu}. \quad (2.11)$$

This corresponds to local transformations

$$\begin{aligned} (x^*)^i &= x^i, \quad i = 1, \dots, n, \\ (u^*)^\mu &= u^\mu + \varepsilon \hat{\eta}^\mu(x, u, \partial u, \dots, \partial^s u) + O(\varepsilon^2), \quad \mu = 1, \dots, m. \end{aligned} \quad (2.12)$$

In *contact symmetries*, the components of infinitesimal generators for dependent variables can depend at most on the first derivatives of the dependent variable of a given scalar DE. Contact symmetries preserve the contact condition $du^* = u_j^* dx_j^*$. A contact transformation is equivalent to a point transformation acting on the space of the given independent variables, the dependent variable and its first derivatives, and, through this, can be naturally extended to point transformations acting on the space of the given independent variables, the dependent variable and its derivatives to any finite order greater than one. In the case of one dependent variable ($m = 1$), contact symmetries are first-order symmetries. When $m = 2$, contact symmetries are point symmetries.

It turns out that higher-order symmetries are not equivalent to point transformations acting on a finite-dimensional manifold including the independent variables, the dependent variables and their derivatives to some finite order. However, they are local symmetries in the sense that the components of the dependent variables in their infinitesimal generators depend at most on a finite number of derivatives of the given DE system's dependent variables so that their calculation only depends on the local behaviour of solutions of the given DE system.

The notion of *local symmetries* includes point symmetries, contact symmetries and higher-order symmetries. Local symmetries are uniquely determined when infinitesimal generators are represented in evolutionary form.

The infinitesimal criterion of invariance under a one-parameter local transformation is provided by the following theorem.

Theorem 2.2. *Let (2.11) be the infinitesimal generator of a one-parameter local transformation, and let $\hat{X}^{(k)}$ be its k th extension. Then the transformation (2.12) is a local (point, contact or higher-order) symmetry of the DE system (2.1) if and only if for each $\alpha = 1, \dots, N$,*

$$\hat{X}^{(k)} R^\alpha(x, u, \partial u, \dots, \partial^k u) = 0, \quad (2.13)$$

when

$$\begin{aligned} R^\sigma(x, u, \partial u, \dots, \partial^k u) &= 0, & \sigma = 1, \dots, N, \\ \partial^l R^\sigma(x, u, \partial u, \dots, \partial^k u) &= 0, & l = 1, \dots, s, \quad \sigma = 1, \dots, N. \end{aligned} \quad (2.14)$$

[Note that if a particular DE $R^\sigma(x, u, \partial u, \dots, \partial^k u) = 0$ has order $k' < k$, then all its differential consequences $\partial^l R^\sigma(x, u, \partial u, \dots, \partial^{k'} u) = 0$, $l = 1, \dots, s + k - k'$, must be computed and used in substitutions.]

The local symmetry analysis algorithm therefore consists in determining the symmetry components $\hat{\eta}^\mu(x, u, \partial u, \dots, \partial^s u)$, and involves the following steps.

1. Obtain *determining equations* by substituting the given DEs (2.14) *and their differential consequences* into the invariance condition (2.13) (in order to restrict to the solution manifold).
2. Obtain *the split system of determining equations*, using the fact that $\hat{\eta}^\mu$ only depend on derivatives of order up to s , i.e., setting coefficients at all independent combinations of derivatives of dependent variables of order greater than s to zero in determining equations.

2.3 Point symmetries of linear DEs

Consider a linear DE system of order k , with n independent variables $x = (x^1, \dots, x^n)$, $n \geq 2$, and m dependent variables $u(x) = (u^1(x), \dots, u^m(x))$, $m \geq 1$. Such system can be written as

$$L^\sigma(x, u, \partial u, \dots, \partial^k u) = F^\sigma(x), \quad \sigma = 1, \dots, N, \quad (2.15)$$

where each $L^\sigma(x, u, \partial u, \dots, \partial^k u)$ is a linear homogeneous differential expression in $u(x)$. It is obvious that if $u(x)$ is a solution of the linear system (2.15), and $w(x)$ is any solution of the linear homogeneous system

$$L^\sigma(x, w, \partial w, \dots, \partial^k w) = 0, \quad \sigma = 1, \dots, N, \quad (2.16)$$

then $\hat{u}(x) = u(x) + w(x)$ is also a solution of (2.15). The transformation $u(x) \rightarrow u(x) + w(x)$ corresponds to obvious infinite point symmetries

$$X_{tr} = w^\mu(x) \frac{\partial}{\partial u^\mu} \quad (2.17)$$

of the linear DE system (2.15). [Symmetries (2.17) may be called *trivial* symmetries.]

For partial differential equations, it follows that the set of point symmetries admitted by any linear PDE system (2.15) is infinite. In particular, the symmetry components $\eta^\mu(x, u)$ in (2.3) are arbitrary solutions of the linear homogeneous PDE system (2.16).

In practice, one is only interested in finding *nontrivial* point symmetries, i.e., performing the symmetry analysis modulo the obvious symmetries (2.17). Moreover, the presence of an infinite number of symmetries often makes the complete symbolic symmetry computation impossible, hence one needs to explicitly exclude them from computations.

For linear ODEs, one of course also has trivial point symmetries (2.17). Unlike the case for PDEs, here the dimension of Lie group is always finite, which is better from the point of view of symbolic computations. However, if the explicit form of the general solution of the corresponding linear homogeneous ODE is unknown, the software will still not be able to compute all point symmetries explicitly.

The following helpful theorems has been established in [14].

Theorem 2.3. Suppose (2.15) is a scalar linear PDE (i.e., $N = m = 1, n \geq 2$) of order $k \geq 2$. Then components ξ^i, η of all of its point transformations (2.2) satisfy

$$\frac{\partial \xi^i}{\partial u} = \frac{\partial^2 \eta}{\partial u^2} = 0, \quad i = 1, \dots, n. \quad (2.18)$$

Theorem 2.4. Suppose (2.15) is a scalar linear ODE (i.e., $N = m = n = 1$) of order $k \geq 3$. Then components ξ, η of all of its point transformations (2.2) satisfy

$$\frac{\partial \xi}{\partial u} = \frac{\partial^2 \eta}{\partial u^2} = 0. \quad (2.19)$$

Ovsianikov ([15], Chapter 6) states that the above result, i.e., independence of ξ 's on the dependent variables, and linear dependence of η 's on the dependent variables, holds for the ‘‘majority of linear DEs’’ (that is, PDE and ODE systems). In general, this ‘‘linear DE conjecture’’ is formulated as follows.

Conjecture 2.1. For a linear DE system (2.15), components ξ^i, η^μ of any its point transformation (2.2) satisfy

$$\frac{\partial \xi^i}{\partial u^\nu} = 0, \quad \frac{\partial^2 \eta^\mu}{\partial u^\nu \partial u^\lambda} = 0, \quad i = 1, \dots, n, \quad \mu, \nu, \lambda = 1, \dots, m. \quad (2.20)$$

If Conjecture 2.1 holds for a given DE system, it follows that its point symmetry generator has the form

$$X = \xi^i(x) \frac{\partial}{\partial x^i} + (f_\nu^\mu(x) u^\nu + g^\mu(x)) \frac{\partial}{\partial u^\mu}. \quad (2.21)$$

or alternatively, in the evolutionary form,

$$\hat{X} = (f_\nu^\mu(x) u^\nu + g^\mu(x) - u_i^\mu \xi^i(x)) \frac{\partial}{\partial u^\mu}. \quad (2.22)$$

The trivial linear symmetries (2.17) evidently correspond to $g^\mu(x) \neq 0$ in (2.21). Two cases arise.

Case 1. Linear homogeneous equations. If the given DE system (2.15) is linear homogeneous, i.e., all $F^\sigma(x) = 0$, then the symmetry determining equations (2.13), (2.14) yield

$$L^\sigma(x, g, \partial g, \dots, \partial^k g) = 0, \quad \sigma = 1, \dots, N,$$

where $g = (g^1(x), \dots, g^m(x))$. Thus to find *all nontrivial point symmetries* of a given linear homogeneous DE system satisfying Conjecture 2.1, one may use a *homogeneous simplified ansatz*

$$X_{s.hom.} = \xi^i(x) \frac{\partial}{\partial x^i} + f_\nu^\mu(x) u^\nu \frac{\partial}{\partial u^\mu} \quad (\Leftrightarrow \quad g^\mu(x) := 0). \quad (2.23)$$

Case 2. Linear non-homogeneous equations. If the given DE system (2.15) is linear non-homogeneous, then using the symmetry determining equations (2.13), (2.14), one obtains that the functions $g = (g^1(x), \dots, g^m(x))$ satisfy a non-homogeneous linear system

$$L^\sigma(x, g, \partial g, \dots, \partial^k g) = \tilde{F}^\sigma(x), \quad \sigma = 1, \dots, N, \quad (2.24)$$

where $\tilde{F}^\sigma(x)$ are generally nonzero expressions depending on functions $F^\sigma(x)$ from (2.15), their derivatives, and coefficients of L^σ . Thus to find *all nontrivial point symmetries* of a linear non-homogeneous DE system satisfying Conjecture 2.1, one needs to take *only one particular solution* $g = (g^1(x), \dots, g^m(x))$ of DEs (2.24); the trivial symmetries are contained in the general homogeneous solution of (2.24).

Remark 2.1. (When does Conjecture 2.1 hold?) In general, conditions on linear DE systems that satisfy Conjecture 2.1 are unknown. The experience of the author confirms that the Conjecture holds for almost all linear DE systems he examined. The Conjecture generally *does not* hold for scalar second-order linear ODEs. For linear PDEs, a simple counterexample is provided by the following PDE system. Consider the linear constant-coefficient wave equation $u_{tt} = u_{xx}$ for $u = u(x, t)$. The simplest of its potential systems is given by

$$v_t = u_x, \quad v_x = u_t. \quad (2.25)$$

One can show that the linear PDE system (2.25) has a point symmetry

$$Z = (u^2 + v^2) \frac{\partial}{\partial u} + 2uv \frac{\partial}{\partial v}, \quad (2.26)$$

which contradicts the Conjecture 2.1. [The symmetry (2.26) is a nonlocal (potential) symmetry of the PDE $u_{tt} = u_{xx}$.]

Remark 2.2. (Nontrivial infinite symmetries.) It is well-known that linear DE systems can have an infinite number of point symmetries, even modulo the trivial symmetries (2.17). This happens when a set of functions $\{\xi^i(x), f_v^\mu(x)\}$ involves arbitrary functions. In particular, all linear PDE systems with constant coefficients have this property. Moreover, if there exists an invertible transformation mapping a given variable coefficient linear PDE system to a constant coefficient linear PDE system, then the given linear PDE system has an infinite number of nontrivial point symmetries ([1], Section 6.5).

3 Local symmetry analysis using “GeM” package

A `Maple` – based package `GeM` has been recently developed by the author. The package routines are capable of finding local (point, contact and higher-order) symmetries, adjoint symmetries and conservation laws of any ODE/PDE system without significant limitations on orders of derivatives and numbers of variables.

The `GeM` package employs an efficient representation of the system under consideration and resulting determining equations: all dependent variables and derivatives are treated as `Maple` symbols, rather than functions or expressions: $\partial B_1 / \partial x \equiv B1x$, etc. For example, the resulting `Maple` expression for the divergence of a 3-vector $\mathbf{B}(x, y, z) = (B_1(x, y, z), B_2(x, y, z), B_3(x, y, z))$ becomes

$$\frac{\partial}{\partial x} B_1(x, y, z) + \frac{\partial}{\partial y} B_2(x, y, z) + \frac{\partial}{\partial z} B_3(x, y, z) = B1x + B2y + B3z. \quad (3.1)$$

This significantly speeds up the computation involving establishing, splitting and solution of symmetry and conservation law determining equations. The final overdetermined linear PDE systems are reduced using `Maple` `rifsimp` [19] routine.

The reduction of overdetermined systems of linear determining equations is usually the most resource-demanding task, in particular, in problems that involve classification. Reduced systems of determining equations are normally much simpler and can be subsequently integrated automatically (`Maple` `pdsolve`, `dsolve`) or even by hand.

After the determining equations are solved, a `GeM` routine is called that outputs all symmetry generators, thus completing the symmetry analysis.

The program sequence for local symmetry analysis includes the following steps.

1. Declaration of variables and the given DE system.
2. Construction of a set of split symmetry determining equations.

3. Simplification and reduction of the overdetermined set of determining equations.
4. Solution of the simplified set of determining equations.
5. Output of symmetry generators.

If the given DE system contains constitutive function(s) and/or constant parameter(s), a classification and case splitting need to be performed (see Section 5 below).

For *linear DE systems*, where one is only interested in finding nontrivial point symmetries, in step 4, special **GeM** routine can be called that verifies whether Conjecture 2.1 holds. Then one subsequently uses the symmetry form (2.21) (and moreover, for homogeneous linear systems, the simplified ansatz (2.23)) to exclude trivial point symmetries. These important routines are demonstrated in the example in Section 6.

For symmetry analysis, it is necessary that a given DE system is provided in a form solvable with respect to a set of leading derivatives. The expressions for these leading derivatives are automatically substituted into the symmetry determining equations by **GeM** package routines, so that the determining equations are considered on the solutions of the given DE system. [In the computation of higher-order symmetries, differential consequences of solved equations are also used.]

It is also important to note that if the differential orders of the equations in the given system are not the same, the **GeM** package routines automatically compute differential consequences of the lower-order DEs, up to the order equal to the maximal differential order of the DEs in the given system. For example, consider a PDE system in a solved form, given by two equations

$$\begin{aligned}u_{tt} &= xuv_x, \\v_t &= v^2 + uu_x.\end{aligned}$$

For this system, the maximal differential order is two. Hence for the second equation, all its differential consequences up to second order will be automatically computed:

$$\begin{aligned}v_{tt} &= 2v(v^2 + uu_x) + u_t u_x + uu_{tx}, \\v_{tx} &= 2vv_x + uu_{xx} + u_x^2.\end{aligned}$$

The full list of features of **GeM** package, with corresponding descriptions and examples, is available [20].

Some run examples are presented in the following sections.

4 Examples of symmetry analysis of nonlinear systems of differential equations using “GeM” package

4.1 Computation of point symmetries of a third-order nonlinear ODE

As a first example, we find point symmetries of a third-order nonlinear ODE given by

$$y'''(x) = Ax^{-3}(y''(x))^3, \quad A = \text{const} \neq 0. \quad (4.2)$$

First, the **GeM** package is initialized using the command

```
with(GeM):
```

Second, one defines variables and differential equations, as follows.

```
gem_decl_vars(indeps=[x], deps=[Y(x)]);
gem_decl_eqs([diff(Y(x),x,x,x)=A*x^(-3)*diff(Y(x),x,x)^3],
solve_for=[diff(Y(x),x,x,x)]);
```

Third, one generates the symmetry determining equations, using the command

```
det_eqs:=gem_symm_det_eqs([x, Y(x)]);
```

The arguments of the command define the dependence of symmetry components. Here point symmetry components will depend on x and $Y(x)$.

Next, the overdetermined system is simplified, as follows.

```
sym_components:=gem_symm_components();
simplified_eqs:=DEtools[rifsimp](det_eqs, sym_components, mindim=1);
```

In particular, the option `mindim=1` forces the output of the number of linearly independent solutions of equations `simplified_eqs`, i.e., the number of point symmetries of the ODE system (4.2). In this example, there are three point symmetries.

Finally, the determining equations are solved, e.g., using the internal Maple solver

```
symm_sol:=pdsolve(simplified_eqs[Solved]);
```

This yields symmetry components containing three arbitrary constants. The command

```
gem_output_symm(symm_sol);
```

prints the three point symmetries separately:

$$X_1 = \frac{\partial}{\partial y}, \quad X_2 = x \frac{\partial}{\partial y}, \quad X_3 = x \frac{\partial}{\partial x} + 3y \frac{\partial}{\partial y}.$$

4.2 Computation of point symmetries of a nonlinear PDE system of gas dynamics equations in one space dimension

Now consider the nonlinear PDE system of polytropic Euler planar gas dynamics equations [21,22], given by

$$\begin{aligned} \rho_t + (\rho v)_x &= 0, \\ \rho(v_t + vv_x) + p_x &= 0, \\ p_t + vp_x + \gamma pv_x &= 0, \end{aligned} \tag{4.3}$$

for the particular case of the polytropic exponent $\gamma = 3$. The Maple program using GeM routines is initialized as follows.

```
with(GeM):
gem_decl_vars(indeps=[x,t], deps=[V(x,t),P(x,t),R(x,t)]);
gem_decl_eqs([
diff(R(x,t),t)+diff(R(x,t)*V(x,t),x),
R(x,t)(diff(V(x,t),t)+V(x,t)*diff(V(x,t),x))
+diff(P(x,t),x)=0,
diff(P(x,t),t)+V(x,t)*diff(P(x,t),x)
+3*P(x,t)*diff(V(x,t),x)=0
],
solve_for=[diff(R(x,t),t), diff(V(x,t),t), diff(P(x,t),t)]);
```

The point symmetry determining equations are generated using the command

```
det_eqs:=gem_symm_det_eqs([x,t, R(x,t),V(x,t),P(x,t)]);
```

The variable `det_eqs` now contains the split overdetermined system of 27 determining equations for the unknown point symmetry components depending on x, t, ρ, v and p . Next, the overdetermined system is simplified, as follows.

```
sym_components:=gem_symm_components();
simplified_eqs:=DEtools[rifsimp](det_eqs, sym_components, mindim=1);
```

The option `mindim=1` forces the output of the dimension of the solution set, which is seven for the current example. The determining equations are solved, and symmetries are output using the commands

```
symm_sol:=pdsolve(simplified_eqs[Solved]);
gem_output_symm(symm_sol);
```

As a result, one obtains the seven point symmetry generators

$$\begin{aligned} X_1 &= \frac{\partial}{\partial x}, & X_2 &= \frac{\partial}{\partial t}, & X_3 &= t \frac{\partial}{\partial t} + x \frac{\partial}{\partial x}, \\ X_4 &= t \frac{\partial}{\partial x} + \frac{\partial}{\partial v}, & X_5 &= x \frac{\partial}{\partial x} + v \frac{\partial}{\partial v} + p \frac{\partial}{\partial p} - \rho \frac{\partial}{\partial \rho}, & X_6 &= p \frac{\partial}{\partial p} + \rho \frac{\partial}{\partial \rho}, \\ X_7 &= xt \frac{\partial}{\partial x} + t^2 \frac{\partial}{\partial t} + (x - vt) \frac{\partial}{\partial v}. \end{aligned}$$

4.3 Computation of local symmetries of the Korteweg-de Vries equation

As a third example, consider the KdV equation [25]

$$u_t + uu_x + u_{xxx} = 0. \tag{4.4}$$

We now use GeM software to compute point and higher-order symmetries (2.11) of the PDE (4.4) depending on x -derivatives of orders up to 5.

First, variables and equations are defined.

```
with(GeM):gem_decl_vars(indeps=[x,t], deps=[U(x,t)]);
gem_decl_eqs([diff(U(x,t),t)+U(x,t)*diff(U(x,t),x)+diff(U(x,t),x,x,x)=0],
solve_for=[diff(U(x,t),t)]);
```

Then, the symmetry determining equations are generated.

```
det_eqs:=gem_symm_det_eqs([x,t, U(x,t), diff(U(x,t),x),
diff(U(x,t),x,x), diff(U(x,t),x,x,x), diff(U(x,t),x,x,x,x),
diff(U(x,t),x,x,x,x,x)],
in_evolutionary_form=true);
```

[Note that derivatives containing t can be omitted since they are expressible in terms of x -derivatives from the PDE (4.4).] The option `in_evolutionary_form=true` enforces the computation in the evolutionary form (2.11). For uniqueness, all higher-order symmetries are normally computed in the evolutionary form.

Next, the overdetermined system is simplified, as follows.

```
sym_components:=gem_symm_components();
simplified_eqs:=DEtools[rifsimp](det_eqs, sym_components, mindim=1);
```

Finally, the determining equations are solved, and symmetries are output using the commands

```
symm_sol:=pdsolve(simplified_eqs[Solved]);
gem_output_symm(symm_sol);
```

which yield the five local symmetries

$$\begin{aligned} X_1 &= u_x \frac{\partial}{\partial u}, & X_2 &= (tu_x - 1) \frac{\partial}{\partial u}, & X_3 &= (uu_x + u_{xxx}) \frac{\partial}{\partial u}, \\ X_4 &= (2u + (x - 3tu)u_x - 3tu_{xxx}) \frac{\partial}{\partial u}, \\ X_5 &= \left(\left(\frac{1}{4}u^2u_x + u_{xx} \right) u_x + \frac{1}{2}uu_{xxx} + \frac{3}{10}u_{xxxx} \right) \frac{\partial}{\partial u}. \end{aligned}$$

5 Symmetry classification problems

5.1 Equivalence transformations and their symbolic computation. Procedure of symmetry classification

When a given DE system contains classifying (“constitutive”) functions and/or parameters, finding its symmetries leads to a *classification problem*, which consists in the isolation of particular forms of constitutive functions and/or parameter values for which the given system possesses an extended symmetry structure. Such classification problems often arise in applications.

For such families of DE systems, it is useful to consider their *equivalence transformations* – transformations that preserve the form of the equations in the system but may change the form of the constitutive functions and/or parameters. In particular, symmetry classification tables are usually presented modulo known equivalence transformations: symmetries are computed only for simplest forms of constitutive functions and/or parameters that are essentially different (i.e., not related by an equivalence transformation). For all other forms of constitutive functions and/or parameters, symmetries can be easily derived using equivalence transformations. Work on equivalence transformations was initiated in [16]. For further details and references, see, for example, [4] (Chapter 1).

Equivalence transformations

Consider the family \mathcal{F}_K of DE systems (2.1), involving L constitutive functions and/or parameters $K = (K_1, \dots, K_L)$. Such functions may depend on particular dependent and independent variables of the system, as well as derivatives of dependent variables.

Definition 5.2. A one-parameter Lie group of *equivalence transformations* of a family \mathcal{F}_K of DE systems is a one-parameter Lie group of transformations given by

$$\begin{aligned} \tilde{x}^i &= f^i(x, u; \varepsilon), & i &= 1, \dots, n, \\ \tilde{u}^\mu &= g^\mu(x, u; \varepsilon), & \mu &= 1, \dots, m, \\ \tilde{K}_l &= G_l(x, u, K; \varepsilon), & l &= 1, \dots, L, \end{aligned} \tag{5.5}$$

which maps a DE system (2.1) in \mathcal{F}_K into another DE system in the same family.

[Note: if constitutive functions/parameters K are not modified under the transformations (5.5), then the transformation (5.5) is simply a point symmetry of each DE system in the family \mathcal{F}_K .]

Simple one-parameter Lie groups of equivalence transformations (like translation and scaling of constitutive functions/parameters) can be often found by inspection.

Symbolic computation of equivalence transformations using the “GeM” package

The sequence for computation of equivalence transformations is rather similar to that for point symmetry computations (Section 3).

1. Declaration of variables and the given DE system, treating constitutive functions/parameters as new dependent variables.
2. Construction of a split system of symmetry determining equations.
3. Simplification and reduction of the determining equations.
4. Solution of the simplified set of determining equations (using, e.g., Maple internal solver `pdsolve`).
5. Output of infinitesimal generators (candidates for equivalence transformations).

Then the user integrates the infinitesimal generators, to obtain the finite form of equivalence transformations; this also can be done using Maple internal solvers.

As an example, consider the family of nonlinear diffusion equations [23]

$$u_t - (K(u)u_x)_x = 0 \quad (5.6)$$

involving a the constitutive function $K(u)$, $K'(u) \neq 0$. In order to find one-parameter Lie groups of equivalence transformations of the family (5.6), one applies the standard Lie symmetry algorithm to the general DE (5.6), treating the constitutive function $K(u)$ as a new dependent variable. As remarked above, one needs to assume that the symmetry components for t , x and u are independent of $K(u)$, following the definition (5.5).

The Maple code for the program is as follows.

```
with(GeM):
gem_decl_vars(indeps=[x,t], deps=[U(x,t), K(x,t)]);
gem_decl_eqs([diff(U(x,t),t)=diff(K(x,t)*diff(U(x,t),x),x)],
solve_for=[diff(U(x,t),t)]);
```

The symmetry determining equations are generated using

```
det_eqs:=gem_symm_det_eqs([x,t, U(x,t),K(x,t)]);
```

which yields 29 determining equations. Conditions of independence of symmetry components for t , x and u of $K(u)$ are appended using the command

```
det_eqs1:= [ det_eqs[], diff(xi_t(x,t,U,K),K)=0, diff(xi_x(x,t,U,K),K)=0,
diff(eta_U(x,t,U,K),K)=0]:
```

Then the reduction of the determining equations stored in `det_eqs1` is performed:

```
sym_components:=gem_symm_components();
simplified_eqs:=DEtools[rifsimp](det_eqs1, sym_components, mindim=1);
```

The simplified determining equations `simplified_eqs` are solved, and symmetries are output using the commands

```
symm_sol:=pdsolve(simplified_eqs[Solved]);
gem_output_symm(symm_sol);
```

The resulting six generators are given by

$$\begin{aligned} X_1 &= \frac{\partial}{\partial t}, & X_2 &= \frac{\partial}{\partial x}, & X_3 &= \frac{\partial}{\partial u}, & X_4 &= x \frac{\partial}{\partial x} + 2t \frac{\partial}{\partial t}, \\ X_5 &= u \frac{\partial}{\partial u}, & X_6 &= F(t) \frac{\partial}{\partial t} - KF'(t) \frac{\partial}{\partial K}. \end{aligned} \quad (5.7)$$

One can check that for the generators X_6 with the arbitrary function $F(t)$, when $F''(t) \neq 0$ the transformed function \tilde{K} is not a function of \tilde{u} only. Therefore these generators do not correspond to equivalence transformations of the family (5.6).

The finite form of the equivalence transformations of the family of PDEs (5.6) arises from the six generators X_1, \dots, X_6 ($F'(t) = \text{const}$) and therefore involves six arbitrary parameters. It is given by

$$\tilde{t} = a_4 t + a_1, \quad \tilde{x} = a_5 x + a_2, \quad \tilde{u} = a_6 u + a_3, \quad \tilde{K}(\tilde{u}) = \frac{a_5^2}{a_4} K(u), \quad (5.8)$$

where a_1, \dots, a_6 are arbitrary constants with $a_4 a_5 a_6 \neq 0$. In particular, under an equivalence transformation (5.8), a PDE system (5.6) with variables x, t, u and constitutive function $K(u)$ is mapped into a PDE system with variables $\tilde{x}, \tilde{t}, \tilde{u}$ with the constitutive function $\tilde{K}(\tilde{u}) = \frac{a_5^2}{a_4} K\left(\frac{\tilde{u} - a_3}{a_6}\right)$.

5.2 Symmetry classification procedure and its symbolic implementation

For DE systems involving constitutive functions and/or parameters, symmetry determining equations are obtained as usual, using the algorithm described in Section 2.

After determining equations (2.13), (2.14) are obtained and split, the further reduction of the overdetermined system essentially depends on the constitutive functions/parameters. Indeed, for their particular relations, such reduction may go differently and thus yield additional symmetries. Such relations or conditions for constitutive functions and/or parameters are called *pivots*, and formulated as $p_i(K, x, u) = 0$, $i = 1, 2, \dots$

Complete analysis of pivots and isolation of all different cases arising from different values of pivots (*case splitting*) is effectively implemented in **Maple's** `rif` package [17–19] and is used during for symmetry classification with **GeM** package.

The symmetry classification program that uses **GeM** package requires the execution of two program sequences.

(i) Computation of equivalence transformations (Section 5.1).

(ii) Symmetry classification.

1. Declaration of variables and the given DE system. Constitutive functions/parameters are declared using special options `freefunc=[...]` and `freeconst=[...]` (see example in Section 5.3 below).
2. Construction of a split system of symmetry determining equations.
3. Simplification and reduction of the overdetermined set of determining equations using `rifsimp`, with `casesplit` option for case splitting with respect to constitutive functions/parameters.
4. For each of the cases obtained in step 3:
 - (a) Choice of the simplest form of constitutive functions/parameters, using equivalence transformations obtained above (by hand).
 - (b) Solution of the simplified set of determining equations, for the specific (chosen) form of constitutive functions/parameters.
 - (c) Output of symmetry generators in the given case.

In the above program sequence, as illustrated in the example in Section 5.3 below, step 4 has to be programmed separately for each case arising in step 3 which yields a different symmetry structure. The main reason for this intentional lack of automation is the utility of equivalence transformations

for the most compact presentation of the symmetry classification table. Indeed, for each classification case, solving pivot equations yields general forms of constitutive functions/parameters for that case, which need to be put in the “simplest” form using equivalence transformations. Yet the notion of “simplest” is hard to make algorithmic, hence case studies were left to the user.

Here it is worth remarking that though `rifsimp`-generated case trees are complete by construction, they can be redundant, presenting non-distinct cases as distinct ones in a non-obvious way. An excellent attempt to correct this minor deficiency using equivalence transformations was made in [26].

5.3 An example of point symmetry classification

As an example, consider the classification of point symmetries of the nonlinear diffusion equation (5.6) with respect to the constitutive function $K(u)$ ($K'(u) \neq 0$).

First, one defines variables using the commands

```
with(GeM):
gem_decl_vars(indeps=[x,t], deps=[U(x,t)], freefunc=[K(U(x,t))]);
```

where the optional parameter `freefunc=[...]` is used to define arbitrary function(s). [Arbitrary constants can be specified using another optional parameter `freeconst=[...]`.]

The given equation (5.6) is defined using

```
gem_decl_eqs([diff(U(x,t),t)=diff(K(U(x,t)))*diff(U(x,t),x),x],
solve_for=[diff(U(x,t),t)]);
```

The symmetry determining equations are generated using the command

```
det_eqs:=gem_symm_det_eqs([x,t, U(x,t)]);
```

which yields 10 determining equations.

Next, one performs the automatic reduction and case splitting of the overdetermined linear PDE system stored in `det_eqs`.

```
sym_components:=gem_symm_components();
split_eqs:=DEtools[rifsimp](det_eqs, sym_components,
casesplit, mindim=1);
```

The `Maple` variable `split_eqs` now contains a table of different computed cases. The case tree can be plotted using the command

```
caseplot(split_eqs,pivots);
```

[Note that depending on the version of `Maple` that is used, case splitting can occur differently, and moreover, some cases can yield the same symmetries. However, the complete analysis of a tree always yields complete results.] For the PDE (5.6), the case tree is shown in Figure 1. The pivot expressions are given by

$$p_1 = K(u), \quad p_2 = K(u), \quad p_3 = K'(u), \quad p_4 = 4K(u)K''(u) - 7(K'(u))^2,$$

$$p_5 = K(u)K'(u)K'''(u) - 2K(u)(K''(u))^2 + (K'(u))^2K''(u).$$

In particular, at each pivot, the left branch of the case tree in Figure 1 corresponds to the case where the pivot expression vanishes, and the right branch – to the case when the pivot expression is nonzero. Numbers below the branches denote case numbers.

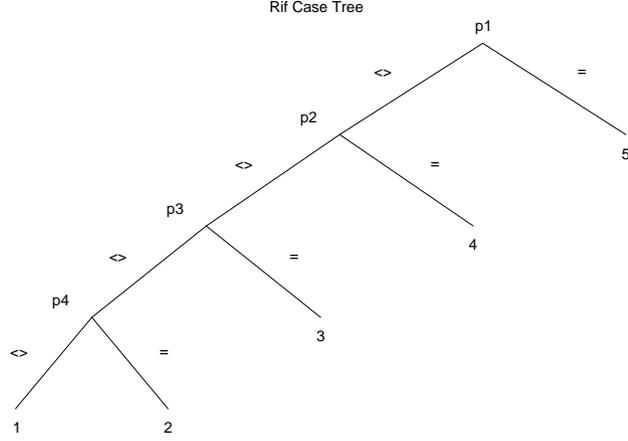


Figure 1: The tree of cases in the classification of point symmetries of the nonlinear diffusion equation (5.6).

For each case m in Figure 1 ($1 \leq q \leq 5$), the corresponding simplified set of determining equations is accessed by calling `split_eqs[q][Solved]`, and the number of independent solutions is given by `split_eqs[q][dimension]`.

Case 1. This is the most general case. Here one uses the commands

```
symm_sol:=pdsolve(split_eqs[1][Solved]);
gem_output_symm(symm_sol);
```

which yield the three symmetries

$$X_1 = \frac{\partial}{\partial x}, \quad X_2 = \frac{\partial}{\partial t}, \quad X_3 = x \frac{\partial}{\partial x} + 2t \frac{\partial}{\partial t}, \quad (5.9)$$

holding for an arbitrary constitutive function $K(u)$.

Case 2. In this case, the solution set has dimension four. This case is characterized by a restricted ODE satisfied by $K(u)$, contained in `split_eqs[2][Solved]`:

$$K'''(u) = \frac{2K(u)(K''(u))^2 - (K'(u))^2 K''(u)}{K(u)K'(u)}. \quad (5.10)$$

One can show that modulo equivalence transformations (5.8), the equation (5.10) has two different solutions: $K(u) = u^\nu$ ($\nu = \text{const}$) and $K(u) = e^u$.

Case 2a. For $K(u) = u^\nu$, one obtains the corresponding point symmetries using commands

```
case2a_symm_sol:=pdsolve(subs(K(U)=U^nu, split_system[2][Solved]));
gem_output_symm(case2a_symm_sol);
```

This yields the three generic symmetries (5.9), and the additional symmetry

$$X_4 = x \frac{\partial}{\partial x} + \frac{2}{\nu} u \frac{\partial}{\partial u}. \quad (5.11)$$

Case 2b. For $K(u) = e^u$, one uses commands

```
case2b_symm_sol:=pdsolve(subs(K(U)=exp(U), split_system[2][Solved]));
gem_output_symm(case2b_symm_sol);
```

This yields the three generic symmetries (5.9), and the additional symmetry

$$X_5 = x \frac{\partial}{\partial x} + 2 \frac{\partial}{\partial u}. \quad (5.12)$$

Case 3. In this case, the solution set has dimension five, and $K(u)$ is restricted to satisfying the ODE

$$K''(u) = \frac{7}{4} \frac{(K'(u))^2}{K(u)}. \quad (5.13)$$

The general second-order ODE (5.13) involves two integration constants, both of which can be removed using equivalence transformations (5.8). Thus modulo equivalence transformations, the only solution of the equation (5.13) is $K(u) = u^{-4/3}$. The corresponding point symmetries are computed using commands

```
case3_symm_sol:=pdsolve(subs(K(U)=U^(-4/3), split_system[3][Solved]));
gem_output_symm(case3_symm_sol);
```

This yields the three symmetries (5.9), the symmetry (5.11) (with $\nu = -4/3$), and the additional symmetry

$$X_6 = x^2 \frac{\partial}{\partial x} - 3xu \frac{\partial}{\partial u}. \quad (5.14)$$

Cases 4 and 5. These cases correspond to linear diffusion equations ($K(u) = \text{const}$ and $K(u) = 0$, respectively), and hence the PDE (5.6) has an infinite number of point symmetries in these cases. Indeed, `split_eqs[q][dimension] = ∞` for $q = 4, 5$. This completes the classification of point symmetries of the nonlinear diffusion equation (5.6).

Finally, the point symmetry classification of the nonlinear diffusion equation (5.6) is given in Table 1 [23].

Table 1: Point symmetry classification for the nonlinear diffusion equation (5.6).

$K(u)$	#	Point Symmetries
Arbitrary	3	$X_1 = \frac{\partial}{\partial x}, X_2 = \frac{\partial}{\partial t}, X_3 = x \frac{\partial}{\partial x} + 2t \frac{\partial}{\partial t}.$
u^ν	4	$X_1, X_2, X_3, X_4 = x \frac{\partial}{\partial x} + \frac{2}{\nu} u \frac{\partial}{\partial u}.$
e^u	4	$X_1, X_2, X_3, X_5 = x \frac{\partial}{\partial x} + 2 \frac{\partial}{\partial u}.$
$u^{-4/3}$	5	$X_1, X_2, X_3, X_4 \left(\nu = -\frac{4}{3} \right), X_6 = x^2 \frac{\partial}{\partial x} - 3xu \frac{\partial}{\partial u}.$

6 Example of point symmetry classification of linear systems of differential equations using “GeM” package

Consider the linear PDE system [24]

$$v_t = u_x, \quad u_t = c^2(x)v_x. \quad (6.15)$$

for with dependent variables $u(x, t), v(x, t)$, which is a potential system of a linear wave equation

$$u_{tt} = c^2(x)u_{xx} \quad (6.16)$$

with variable wave speed $c(x)$. We now use GeM package to classify nontrivial point symmetries of the system (6.15) with respect to the wave speed $c(x)$, in the assumption $c'(x) \neq 0$. The classification is done modulo the equivalence transformations of the system (6.15), which are computed in the same way as shown in the example of Section 5.1, and are given by

$$\begin{aligned} \tilde{t} &= a_1 a_5^{-1} t + a_3 v + a_6, & \tilde{x} &= a_1 a_5 x + a_3 u + a_7, \\ \tilde{u} &= a_2 a_5^2 u + a_4 x + a_8, & \tilde{v} &= a_2 v + a_4 t + a_9, & \tilde{c}(\tilde{x}) &= a_5^2 c(x), \end{aligned} \quad (6.17)$$

where a_1, \dots, a_9 are arbitrary constants with $a_1 a_2 a_5 \neq 0$.

The variables and the equations are defined using the command

```
with(GeM):
gem_decl_vars(indeps=[x,t], deps=[U(x,t),V(x,t)]);
gem_decl_eqs([diff(U(x,t),t)=C(x)^2*diff(V(x,t),x), ,
diff(V(x,t),t)=diff(U(x,t),x)],
solve_for=[diff(U(x,t),t), diff(V(x,t),t)]);
```

Then, the command

```
det_eqs:=gem_symm_det_eqs([ind, U(x,t), V(x,t)]);
```

is used to generate the ten determining equations (split with respect to derivatives v_t, v_x) and place them into the variable `det_eqs`.

Next, similarly to the previous classification example, one performs the automatic reduction and case splitting of the overdetermined linear PDE system stored in `det_eqs`.

```
sym_components:=gem_symm_components();
split_eqs:=DEtools[rifsimp](det_eqs, sym_components, casesplit, mindim=1);
```

The Maple variable `split_eqs` now contains a table of different computed cases. The case tree is plotted using the command

```
caseplot(split_eqs,pivots);
```

This yields four cases (Figure 2), with the three pivot expressions given in terms of $c = c(x)$ by

$$\begin{aligned} p_1 &= c, & p_2 &= c', \\ p_3 &= c^2[4(c'')^3 + (c')^2 c'''' - 5c' c'' c'''] + (c')^2[3CC' c'' - 4c(c'')^2 + (c')^2 c'']. \end{aligned} \quad (6.18)$$

Cases 3 and 4 in Figure 2 correspond to $c' = 0$ and $c = 0$, and are not considered, since we are only interested in variable-coefficient equations. We now study cases 1 and 2.

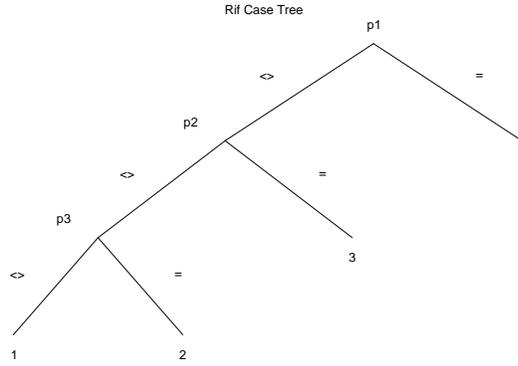


Figure 2: The tree of cases in the classification of point symmetries of the linear variable-coefficient wave equation potential system (6.15).

Before proceeding with the symmetry classification, we would like to verify whether the “linear conjecture” (Conjecture 2.1) holds for the variable-coefficient cases in general. We use the command

```
gem_linear_pointsymm_check_conjecture({det_eqs[ ], diff(C(x),x)<>0});
```

which verifies the conjecture and establishes that it does hold for $c' \neq 0$. As its argument, the above command takes a set of PDEs (determining equations). The implementation of the routine `gem_linear_pointsymm_check_conjecture` involves checking Conjecture 2.1 by contradiction. It is sequentially assumed that one of the conditions (2.20) is false (using Maple non-equality $\langle \rangle$). Such condition is appended to the argument of the routine, and then `rifsimp` is applied to the resulting PDE system. The Conjecture 2.1 holds if each such PDE system is found to be inconsistent.

Case 1. This is the most general case. Here one uses the command

```
syms_case1:=gem_linear_pointsymm_use_conjecture(rs1[1][Solved]);
```

which applies the *simplified ansatz* (2.23) to its argument, thus excluding the obvious symmetries of the linear homogeneous system (6.15).

Finally, symmetries are output using the command

```
gem_output_syms(symm_sol);
```

which yield the two symmetries

$$X_1 = \frac{\partial}{\partial t}, \quad X_2 = u \frac{\partial}{\partial u} + v \frac{\partial}{\partial v}, \quad (6.19)$$

holding for an arbitrary constitutive function $c(x)$.

Case 2. In [24] (see also [1], Section 7.2), it was shown that the pivot p_3 in (6.18) can be rewritten as

$$cc'(c/c')'' = \text{const} = K. \quad (6.20)$$

When $K = 0$, modulo the equivalence transformations (6.17), the ODE (6.20) has the following independent solutions: $c(x) = x$, $c(x) = x^M$ ($M = \text{const}$), and $c(x) = e^x$. When $K \neq 0$, the ODE

(6.20) reduces to first-order ODEs; these cases are omitted here for brevity (for details, see [1], Section 7.2).

Case 2a: $c(x) = x$. Here we again use the command that employs the simplified ansatz (2.23):

```
syms_case1:=gem_linear_pointsymm_use_conjecture(rs1[2][Solved],
  free_func_form = [C(x)=x]);
gem_output_symm(symm_sol);
```

which yields the two symmetries (6.19), and two additional symmetries

$$\begin{aligned} X_3 &= x \frac{\partial}{\partial x} + u \frac{\partial}{\partial u}, \\ X_4 &= 2 \log(x) \frac{\partial}{\partial t} + 2tx \frac{\partial}{\partial x} + (tu - xv) \frac{\partial}{\partial u} - \left(\frac{u}{x} + tv \right) \frac{\partial}{\partial v}. \end{aligned} \tag{6.21}$$

Case 2b: $c(x) = x^M$, $M \neq 0, 1$. The simplified ansatz (2.23) here us used by the command

```
syms_case1:=gem_linear_pointsymm_use_conjecture(rs1[2][Solved],
  free_func_form = [C(x)=x^M,M<>0]);
gem_output_symm(symm_sol);
```

which yields the two symmetries (6.19), and two additional symmetries

$$\begin{aligned} X_5 &= (1 - M)t \frac{\partial}{\partial x} + x \frac{\partial}{\partial x} + Mu \frac{\partial}{\partial u}, \\ X_6 &= \left((1 - M)^{-1} x^{2(1-M)} + (1 - M)t^2 \right) \frac{\partial}{\partial t} + 2tx \frac{\partial}{\partial x} \\ &\quad - (tu(1 - 2M) + xv) \frac{\partial}{\partial u} - (x^{1-2M}u + tu) \frac{\partial}{\partial v}. \end{aligned} \tag{6.22}$$

Case 2c: $c(x) = e^x$. Here one uses a similar command

```
syms_case1:=gem_linear_pointsymm_use_conjecture(rs1[2][Solved],
  free_func_form = [C(x)=exp(x)]);
gem_output_symm(symm_sol);
```

which yields the two symmetries (6.19), and two additional symmetries

$$\begin{aligned} X_7 &= t \frac{\partial}{\partial x} - \frac{\partial}{\partial x} - u \frac{\partial}{\partial u}, \\ X_8 &= (t^2 + e^{-2x}) \frac{\partial}{\partial t} - 2t \frac{\partial}{\partial x} + (v - 2tu) \frac{\partial}{\partial u} + e^{-2x} u \frac{\partial}{\partial v}. \end{aligned} \tag{6.23}$$

The symmetry classification results are summarized in Table 2.

Following [1] (Section 7.2), we note that symmetries X_4 , X_6 , X_8 yield nonlocal symmetries of the linear wave equation (6.16), since the u -component of each of these symmetries essentially depends on the nonlocal variable v . For details on nonlocally related PDE systems, see [4], Chapter 3.

Table 2: Partial point symmetry classification for the potential system (6.15) for the linear wave equation (6.16).

$c(x)$	#	Point Symmetries
Arbitrary	2	$X_1 = \frac{\partial}{\partial t}, X_2 = u \frac{\partial}{\partial u} + v \frac{\partial}{\partial v}.$
x	4	$X_1, X_2, X_3 = x \frac{\partial}{\partial x} + u \frac{\partial}{\partial u}, X_4 = 2 \log(x) \frac{\partial}{\partial t} + 2tx \frac{\partial}{\partial x} + (tu - xv) \frac{\partial}{\partial u} - \left(\frac{u}{x} + tv\right) \frac{\partial}{\partial v}.$
$x^M,$ $M \neq 0, 1$	4	$X_1, X_2, X_5 = (1 - M)t \frac{\partial}{\partial x} + x \frac{\partial}{\partial x} + Mu \frac{\partial}{\partial u},$ $X_6 = \left((1 - M)^{-1} x^{2(1-M)} + (1 - M)t^2 \right) \frac{\partial}{\partial t}$ $+ 2tx \frac{\partial}{\partial x} - (tu(1 - 2M) + xv) \frac{\partial}{\partial u} - (x^{1-2M}u + tu) \frac{\partial}{\partial v}.$
e^x	4	$X_1, X_2, X_7 = t \frac{\partial}{\partial x} - \frac{\partial}{\partial x} - u \frac{\partial}{\partial u}, X_8 = (t^2 + e^{-2x}) \frac{\partial}{\partial t} - 2t \frac{\partial}{\partial x} + (v - 2tu) \frac{\partial}{\partial u} + e^{-2x}u \frac{\partial}{\partial v}.$

7 Conclusions

Symbolic computations of symmetries of nonlinear and linear differential equations using the package **GeM** for **Maple** were presented. Using routines of the **GeM** package, one can produce, split, simplify, and completely solve local symmetry determining equations for nonlinear and linear ODE and PDE systems and scalar equations, and output the basis of symmetry generators in a separated form.

If a system of interest is a linear DE system, to perform a successful symbolic computation of its point symmetries, one often needs to exclude obvious point symmetries (2.17). [This is commonly the case for linear PDE systems, which admit an infinite number of such trivial symmetries.] For many linear DE systems, Conjecture 2.1 holds, which enables one to use the simplified symmetry form (2.21) which leads to effective exclusion of obvious point symmetries. In particular, for linear homogeneous DEs, ansatz (2.23) can be used. For linear DE systems, a **GeM** package routine can be used that explicitly verifies Conjecture 2.1 using symbolic computations. If the conjecture holds, another routine employing the ansatz (2.23) for linear homogeneous DEs is available, which returns only nontrivial point symmetries. [Even for DE systems for which the conjecture does *not* hold, the routine employing the simplified symmetry form still can be used, to provide a partial symmetry classification.]

If the given DE system contains constitutive function(s) / parameter(s), a classification and case splitting need to be performed. Symbolic implementations of symmetry classification and of the related problem of computation of equivalence transformations are discussed and illustrated by an example in Section 5.

It remains a practically important open question to formulate an explicit criterion to verify whether or not the statement of Conjecture 2.1 holds for a given linear DE system.

Acknowledgements

The author is grateful to NSERC for research support, and anonymous referees for helpful suggestions.

References

- [1] Bluman G.W. and Kumei S. *Symmetries and Differential Equations*. Applied Mathematical Sciences, No. 81. Springer-Verlag, New York, NY (1989).
- [2] *CRC Handbook of Lie Group Analysis of Differential Equations*. (N.H. Ibragimov, ed.) Vol. 1. CRC Press, Boca Raton, Florida (1993).
- [3] Bogoyavlenskij O. I. (2001). Infinite symmetries of the ideal MHD equilibrium equations. *Phys. Lett. A* **291** (4-5), 256-264.

- [4] Bluman G.W., Cheviakov A.F., and Anco S.C. *Applications of Symmetry Methods to Partial Differential Equations*. Applied Mathematical Sciences, No. 168. Springer, New York (2010).
- [5] Burde G.I. (2001). Potential symmetries of the nonlinear wave equation $u_{tt} = (uu_x)_x$ and related exact and approximate solutions. *J. Phys. A: Math. Gen.* **34**, 5355–5371.
- [6] Wolf T. (2002a). Investigating differential equations with CRACK, LiePDE, Applysymm and ConLaw. In: *Handbook of Computer Algebra, Foundations, Applications, Systems* (J. Grabmeier, E. Kaltofen, and V. Weispfenning, eds.). Springer, New York, NY, pp. 465–468.
- [7] Temuerchaolu. (2003). An algorithmic theory of reduction of differential polynomial systems. *Adv. Math.* **32**, 208–220 (in Chinese).
- [8] Hereman W. (1997). Review of symbolic software for Lie symmetry analysis, *Math. Comp. Mod.* **25**, 115–132.
- [9] Hereman W. (1996). Symbolic software for Lie symmetry analysis. In: *CRC Handbook of Lie Group Analysis of Differential Equations*, Vol. 3: *New Trends in Theoretical Developments and Computational Methods*, Chap. 13, pp. 367–413. (N.H. Ibragimov, ed.) CRC Press, Boca Raton, FL (1996).
- [10] Hereman W. (2005). Symbolic computation of conservation laws of nonlinear partial differential equations in multi-dimensions. *Int. J. Quant. Chem.* **106**, 278–299.
- [11] Butcher J., Carminati J., and Vu K.T. (2003). A comparative study of the computer algebra packages which determine the Lie point symmetries of differential equations. *Comput. Phys. Commun.* **155** (2), 92–114.
- [12] Olver P.J. *Applications of Lie Groups to Differential Equations*. Springer, New York (1986).
- [13] Cheviakov A. F. (2004). Bogoyavlenskij symmetries of ideal MHD equilibria as Lie point transformations. *Phys. Lett. A.* **321** (1), 34–49.
- [14] Bluman G.W. (1990). Simplifying the form of Lie groups admitted by a given differential equation. *J. Math. Anal. Appl.* **145**, 52–62.
- [15] Ovsianikov L.V. *Group Properties of Differential Equations*. Nauka, Novosibirsk (in Russian) (1962).
- [16] Ovsianikov L.V. *Group Analysis of Differential Equations*. Academic Press, New York (1982).
- [17] Reid G.J., Wittkopf A.D., and Boulton A. (1996) Reduction of Systems of Nonlinear Partial Differential Equations to Simplified Involutive Forms. *Eur. J. Appl. Math.* **7**, 604–635.
- [18] Reid G.J., Wittkopf A.D. (2000), Determination of Maximal Symmetry Groups of Classes of Differential Equations. In: *Proc. ISSAC 2000*, ACM Press, 272–280.
- [19] See Maple help system, sections on `rifsimp`, `maxdimsys`, and related commands.
- [20] The GeM package and documentation is available at <http://math.usask.ca/~cheviakov/gem/>.
- [21] Akhatov, I.S., Gazizov, R.K., and Ibragimov, N.H. (1987). Group classification of the equations of nonlinear filtration. *Sov. Math. Dokl.* **35**, 384–386.
- [22] Bluman, G.W., Cheviakov A. F., and Ivanova N.M. (2006). Framework for nonlocally related PDE systems and nonlocal symmetries: extension, simplification, and examples. *J. Math. Phys.* **47**, 113505.
- [23] Ovsianikov, L.V. (1959). Group properties of the nonlinear heat conduction equation. *Dokl. Akad. Nauk USSR* **125**, 492–495 (in Russian).
- [24] Bluman G.W., Kumei S., and Reid G.J. (1988). New classes of symmetries for partial differential equations. *J. Math. Phys.* **29**, 806–811.
- [25] Korteweg D. J. and de Vries G. (1895). On the Change of Form of Long Waves Advancing in a Rectangular Canal, and on a New Type of Long Stationary Waves. *Philosophical Magazine* **39**, 422–443.
- [26] Lisle I. and Huang T. S.-L. (2010) Algorithmic symmetry classification with invariance. *J. Eng. Math.* **66** (1-3), 201–216.