Computer Physics Communications 220 (2017) 56–73

Contents lists available at ScienceDirect

Computer Physics Communications

journal homepage: www.elsevier.com/locate/cpc

Symbolic computation of equivalence transformations and parameter reduction for nonlinear physical models

Alexei F. Cheviakov

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

ARTICLE INFO

Article history:
Received 19 January 2017
Received in revised form 13 June 2017
Accepted 17 June 2017
Available online 18 July 2017

Keywords:
Nonlinear differential equations
Equivalence transformations
Parameter reduction
Symbolic computations

ABSTRACT

An efficient systematic procedure is provided for symbolic computation of Lie groups of equivalence transformations and generalized equivalence transformations of systems of differential equations that contain arbitrary elements (arbitrary functions and/or arbitrary constant parameters), using the software package `GeM` for `Maple`. Application of equivalence transformations to the reduction of the number of arbitrary elements in a given system of equations is discussed, and several examples are considered. The first computational example of generalized equivalence transformations where the transformation of the dependent variable involves an arbitrary constitutive function is presented.

As a detailed physical example, a three-parameter family of nonlinear wave equations describing finite anti-plane shear displacements of an incompressible hyperelastic fiber-reinforced medium is considered. Equivalence transformations are computed and employed to radically simplify the model for an arbitrary fiber direction, invertibly reducing the model to a simple form that corresponds to a special fiber direction, and involves no arbitrary elements.

The presented computation algorithm is applicable to wide classes of systems of differential equations containing arbitrary elements.

1. Introduction

The majority of contemporary mathematical models, in particular, physical models written in terms of partial or ordinary differential equations (PDEs, ODEs), involve constant or variable parameters. These parameters, also called arbitrary or constitutive elements, appear in the equations in the form of arbitrary constants (*constitutive parameters*) that assume values in certain ranges, and/or arbitrary functions (*constitutive functions*) belonging to certain classes. Equivalent formulations of a model at hand, involving fewer constitutive parameters and/or reduced forms of constitutive functions, can lead to a significant simplification of the analysis of the model, in particular, when such analysis involves classifications, such as local symmetry or conservation law classifications. Equivalent formulations of a class of models may be systematically sought using *equivalence transformations*.

Equivalence transformations of a family of DE (differential equation) systems are transformations of problem variables and arbitrary elements that map every DE system from the given family into another system from the same family. As a basic illustration, consider a set of $(1+1)$-dimensional nonlinear diffusion equations for $u(x,t)$, given by

$$u_t = c^2(u)u_{xx}, \tag{1.1}$$

where $c(u)$ is an arbitrary constitutive function. (In (1.1) and throughout the paper where appropriate, subscripts denote derivatives.) The family (1.1) admits, for example, an obvious discrete equivalence transformation $c^*(u) = -c(u)$; in particular, any solution $u(x,t)$ of (1.1) is mapped a solution of a PDE $u_{tt} = (c^*(u))^2 u_{xx}$. Scalings and translations are the simplest kinds of continuous equivalence transformations. In particular, transformations that re-scale variables using their typical values, and thus map a problem into a dimensionless one, or scaling transformations in general, have a long history, and are commonly used to simplify and non-dimensionalize DE problems. For example, the PDE family (1.1) clearly admits scaling and translation-type equivalence transformations

$$x^* = A_3x + A_1, \quad t^* = A_4t + A_2, \quad u^*(x^*,t^*) = A_5u(x,t), \quad c^*(u^*) = \frac{A_3^2}{A_4}c(u), \tag{1.2}$$

E-mail address: chevaikov@math.usask.ca.

http://dx.doi.org/10.1016/j.cpc.2017.06.013
0010-4655/© 2017 Elsevier B.V. All rights reserved.

$A_1, \ldots, A_5 \in \mathbb{R}, A_4 > 0$, so that a solution $u(x, t)$ of a PDE (1.1) with a constitutive function $c(u)$ is mapped into a solution $u^*(x^*, t^*)$ of a PDE

$$u^*_{t^*} = (c^*(u^*))^2 u^*_{x^*x^*},$$

a member of the family (1.1) with a different constitutive function, as long as $A_3^2 \neq A_4$.

The notion of equivalence transformations is closely related to that of *symmetries* of a system of differential equations. A symmetry is a transformation that leaves invariant the solution manifold of a given system, i.e., maps any solution of the system into a solution of the same system. Due to the general topological nature of this definition, a systematic way to calculate *all* symmetries of a given DE system, and similarly, *all* equivalence transformations of a given family of DE systems, is not available. However, equivalence transformations of particular kinds, such as Lie groups of local and nonlocal equivalence transformations, can indeed be computed systematically.

Work on equivalence transformations was initiated in [1]. Various examples, extensions, and applications of the notion of equivalence transformations appear in the literature, such as [2–9] and references therein. A broad review and work on equivalence transformations and some related topics, including both theoretical and computational aspects, is contained in [3]. A related problem, in a certain sense complementary to finding equivalence transformations of a given DE family, is known as the *Cartan equivalence problem* [10–12]; it consists in finding criteria for DEs within the family to be connected by a change of variables from a prescribed transformation group (see, e.g., [3] and references therein for a discussion and recent developments). Significant progress has recently been made in the application of the Cartan's method of equivalence (MoE) to differential equations (see, e.g., Refs. [13–19]). A symbolic implementation of routines for the automated MoE-related computations is contained in the standard `Maple` package `DifferentialGeometry` [20].

The current manuscript presents an algorithmic procedure to calculate Lie groups of local equivalence transformations of a given DE system using a symbolic symmetry and conservation law computation package `GeM` for `Maple` [21–23] developed by the author. A brief review of the notion of equivalence transformations, relations between Lie groups of point equivalence transformations and Lie point symmetries, and extended classes of equivalence transformations, is given in Section 2. The section also contains several illustrative examples, and ends with an outline of a systematic procedure of computation of Lie groups of generalized equivalence transformations, closely related to the algorithm of point symmetry computation. Section 3 presents a typical sequence of steps for complete symbolic computation of Lie groups of generalized equivalence transformations using the `GeM` `Maple` package.

Section 4 is devoted to examples of symbolic computation of equivalence transformations of PDE models. Detailed `Maple` code is included for each example. In the first example, equivalence transformations are computed for the well-known three-parameter family of dimensional Korteweg–de Vries (KdV) equations, and are used to invertibly reduce the family to a single canonical KdV equation involving no arbitrary parameters. In the second example, we compute nonlocal equivalence transformations of a family of nonlinear wave equations. In particular, nonlocal equivalence transformations are found as point transformations of a potential system, and do not correspond to local equivalence transformations of the model.

The final example of Section 4 is devoted to the computation of a Lie group of generalized equivalence transformations of the KdV–Burgers PDE family; it is demonstrated that the transformation component corresponding to the dependent variable explicitly depends on an arbitrary element that is an *arbitrary function* of the model. This is the first computational example of this situation in the literature (see also Remark 2.6).

An additional example considered in Section 5 presents a recent model of finite anti-plane shear displacements in nonlinear incompressible hyperelastic fiber-reinforced hyperelastic materials [24]. Such models arise in the description of multiple types of nonlinear media, including biological membranes [25]. The systematically computed local equivalence transformations are used to drastically simplify the model, by invertible transformations, through the complete elimination of arbitrary elements. Together with the classical example of the Korteweg–de Vries model in Section 4, the example of Section 5 addresses the questions of primary importance for the analysis of nonlinear equations arising in physics and applied science—the identification of essential parameters of the model, and the possibility of invertible parameter reduction.

Section 6 concludes the paper with a discussion of the presented algorithms, their efficiency, results, related classification problems, and an overview of open questions and possible work directions in the area.

Throughout the paper, summation in repeated indices is assumed where appropriate.

2. Equivalence transformations, their computation, and extensions

2.1. Lie groups of equivalence transformations

Consider a family $\mathcal{F}_K$ of DE systems $\mathbf{R}\{x\,;u;K\}$:

$$R^\sigma(x, u, \partial u, \ldots, \partial^k u, K) = 0, \quad \sigma = 1, \ldots, N, \tag{2.1}$$

involving n independent variables $x = (x^1, \ldots, x^n)$, m dependent variables $u(x) = (u^1(x), \ldots, u^m(x))$, and L arbitrary elements (constitutive functions and/or parameters) $K = (K^1, \ldots, K^L)$. In (2.1), partial derivatives are denoted by $u^\mu_i = \partial u^\mu(x)/\partial x^i$. The symbol

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

is used to denote the set of all partial derivatives of order p, $p = 1, 2, \ldots$. The arbitrary elements K in (2.1) may be either arbitrary constants or arbitrary functions within some class. Constitutive functions are assumed to be sufficiently smooth functions of dependent and/or independent variables and/or derivatives of the dependent variables, or combinations thereof.

In general, equivalence transformations for a DE family $\mathcal{F}_K$ (2.1) can be defined as follows.

Definition 2.1. An *equivalence transformation* for a DE family $\mathcal{F}_K$ is a change of variables and arbitrary elements $(x, u, K) \to (x^*, u^*, K^*)$ which maps every DE system $\mathbf{R}\{x\,; u; K\} \in \mathcal{F}_K$ into a DE system $\mathbf{R}\{x^*\,; u^*; K^*\} \in \mathcal{F}_K$.

Definition 2.1 is rather general, and there is no systematic way to compute explicit forms of transformations described therein. For the practical purposes, the following definition describing a special subclass of equivalence transformations is often used [1,3].

Definition 2.2. A *one-parameter Lie group of point equivalence transformations* of a family $\mathcal{F}_K$ of DE systems (2.1) is a one-parameter Lie group of transformations acting in the space of (x, u, K), given by

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

which maps a PDE system $\mathbf{R}\{x\,; u; K\} \in \mathcal{F}_K$ into another PDE system $\mathbf{R}\{x^*\,; u^*; K^*\}$ in the same family.

In formulas (2.2), the arguments Q^ℓ of the corresponding transformation G^ℓ and the tangent vector field component κ^ℓ depend on the nature of the arbitrary element K^ℓ, $\ell = 1, \dots, L$. In particular, if K^ℓ is a constant parameter, Q^ℓ may be the set of all constant parameters of the DE system family (2.1). If K^ℓ is a arbitrary (constitutive) function, Q^ℓ may also involve independent/dependent variables on which K^ℓ depends, other constitutive functions with compatible dependencies, etc. Within Definition 2.2, transformations (2.2) form a group on the (x, u) space; the transformations for x, u do not involve arbitrary elements.

Remark 2.3. The notion of a Lie group of equivalence transformations (2.2) of a DE system is intimately related to that of a Lie group of point symmetries; the appropriate notation, including that for infinitesimal generators, and further details, can be found in Section 2.2.

Multiple examples of equivalence transformations are known in the literature that go beyond Definition 2.2. Some of such examples are considered in Section 2.3.

Example 2.4. The transformations (1.2) involve five one-parameter groups of equivalence transformations. Indeed, they can be written in the global group form as

$$\begin{aligned}
&x^* = e^{\varepsilon_3}x + \varepsilon_1, \quad t^* = e^{\varepsilon_4}t + \varepsilon_2, \quad u^*(x^*, t^*) = e^{\varepsilon_5}u(x, t),\\
&\quad c^*(u^*) = e^{\varepsilon_3 - \varepsilon_4/2}c(u),
\end{aligned} \tag{2.3}$$

with group parameters ε_i, $i = 1, \dots, 5$. The corresponding infinitesimal generators are given by

$$\mathrm{X}_1 = \frac{\partial}{\partial x}, \quad \mathrm{X}_2 = \frac{\partial}{\partial t}, \quad \mathrm{X}_3 = x\frac{\partial}{\partial x} + c\frac{\partial}{\partial c}, \quad \mathrm{X}_4 = t\frac{\partial}{\partial t} - \frac{1}{2}c\frac{\partial}{\partial c}, \quad \mathrm{X}_5 = u\frac{\partial}{\partial u}. \tag{2.4}$$

Example 2.5 (*Equivalence Transformations and Parameter Reduction*). As a second example, consider a classical family of "dimensional Korteweg–de Vries" equations for the unknown function $u(x, t)$, given by

$$u_t + au_x + buu_x + qu_{xxx} = 0, \tag{2.5}$$

involving three constant parameters $a, b, q \in \mathbb{R}$, $b, q \neq 0$. The model (2.5) arises in multiple physical contexts [26], for example, it describes the evolution of the fluid surface elevation $u(x, t)$ in a limit considering small-amplitude long waves in a shallow channel. We now recollect the well-known equivalence transformations of the PDE family (2.5). (An example of symbolic software computations of these equivalence transformations is given below in Section 4.1.)

A basic set of equivalence transformations of the PDE family (2.5) is given by infinitesimal generators

$$\begin{aligned}
&\mathrm{Y}_1 = \frac{\partial}{\partial x}, \quad \mathrm{Y}_2 = \frac{\partial}{\partial t}, \quad \mathrm{Y}_3 = \frac{\partial}{\partial u} - b\frac{\partial}{\partial a}, \quad \mathrm{Y}_4 = t\frac{\partial}{\partial x} + \frac{\partial}{\partial a},\\
&\mathrm{Y}_5 = x\frac{\partial}{\partial x} + t\frac{\partial}{\partial t} + 2q\frac{\partial}{\partial q}, \quad \mathrm{Y}_6 = x\frac{\partial}{\partial x} + 3t\frac{\partial}{\partial t} - 2u\frac{\partial}{\partial u} - 2a\frac{\partial}{\partial a},\\
&\mathrm{Y}_7 = x\frac{\partial}{\partial x} + 3t\frac{\partial}{\partial t} - 2a\frac{\partial}{\partial a} - 2b\frac{\partial}{\partial b}.
\end{aligned} \tag{2.6}$$

The corresponding point transformations can be written, for example, as

$$\begin{aligned}
&x^* = \frac{A_5}{A_6A_7}(x - A_4t) + A_1, \qquad t^* = \frac{A_5}{A_6^3A_7^3}t + A_2, \qquad u^*(x^*, t^*) = A_6^2(u(x, t) + A_3),\\
&a^* = A_6^2A_7^2(a - A_3b - A_4), \qquad b^* = A_7^2\,b, \qquad q^* = A_5^2\,q
\end{aligned} \tag{2.7}$$

with seven arbitrary constants $A_1, \dots, A_7 \in \mathbb{R}$.

The equivalence transformations (2.7) can be used to invertibly map the PDE (2.5) into one of the standard forms of the KdV equation, for example,

$$u_t + uu_x + u_{xxx} = 0, \tag{2.8}$$

thus reducing the number of constant parameters from three to zero. To do this, one removes the term au_x ($a \neq 0$), and re-scales the variables. In common KdV derivations, one of two common ways is used to remove the term au_x. This can be achieved by passing to a traveling wave coordinate $x^* = x - at$, $t^* = t$, $u^*(x^*, t^*) = u(x, t)$, that is, using the transformation (2.7) with

$$A_1 = A_2 = A_3 = 0, \quad A_4 = a, \quad A_5 = A_6 = A_7 = 1.$$

Another way consists in a pure shift of the dependent variable, i.e., the transformation (2.7) with

$$A_1 = A_2 = A_4 = 0, \quad A_3 = a/b, \quad A_5 = A_6 = A_7 = 1, \tag{2.9}$$

leading to the choice of variables $x^* = x$, $t^* = t$, $u^*(x^*, t^*) = u(x, t) + A_3$. Both of the above choices, as well as any other combined transformation leading to $a^* \propto a - A_3 b - A_4 = 0$ in (2.7), yield an equivalent PDE from the family (2.5), given by (after the omission of the asterisks)

$$u_t + buu_x + qu_{xxx} = 0. \tag{2.10}$$

Due to the freedom in changing the signs of x and u, i.e., in the choice $A_5, A_6, A_7 = \pm 1$ in (2.7), one can assume without loss of generality that in (2.10), $b, q > 0$. Finally, using again the transformations (2.7) with

$$A_1 = A_2 = A_3 = A_4 = 0, \quad A_5 = q^{-1/2}, \quad A_6 = 1, \quad A_7 = b^{-1/2},$$

one transforms the PDE (2.10) into the standard form (2.8).

2.2. Equivalence transformations and point symmetries

Lie groups of equivalence transformations (2.2) are closely related to Lie groups of point symmetries of differential equations (e.g., [1,2]). Let $\mathbf{R}\{x\,;\,u\} \in \mathcal{F}_K$ given by (2.1) be a specific DE system within the indicated family, for a fixed set of constitutive quantities K.

Consider a one-parameter Lie group of point transformations

$$\begin{aligned} (x^*)^i &= f^i(x, u; \varepsilon) = x^i + \varepsilon \xi^i(x, u) + O(\varepsilon^2), \quad i = 1, \ldots, n, \\ (u^*)^\mu &= g^\mu(x, u; \varepsilon) = u^\mu + \varepsilon \eta^\mu(x, u) + O(\varepsilon^2), \quad \mu = 1, \ldots, m, \end{aligned} \tag{2.11}$$

with the infinitesimal generator

$$\mathrm{X} = \xi^i(x, u)\frac{\partial}{\partial x^i} + \eta^\mu(x, u)\frac{\partial}{\partial u^\mu}, \tag{2.12}$$

and the kth prolongation of (2.12) is given by

$$\begin{aligned} \mathrm{X}^{(k)} &= \xi^i(x, u)\frac{\partial}{\partial x^i} + \eta^\mu(x, u)\frac{\partial}{\partial u^\mu} + \eta_i^{(1)\mu}(x, u, \partial u)\frac{\partial}{\partial u_i^\mu} \\ &\quad + \cdots + \eta_{i_1 \ldots i_k}^{(k)\mu}(x, u, \partial u, \ldots, \partial^k u)\frac{\partial}{\partial u_{i_1 \ldots i_k}^\mu}. \end{aligned} \tag{2.13}$$

Here the extended infinitesimals $\eta_i^{(1)\mu}, \ldots, \eta_{i_1 \ldots i_k}^{(k)\mu}$ are defined through the formulas

$$\eta_i^{(1)\mu} = \mathrm{D}_i \eta^\mu - (\mathrm{D}_i \xi^j) u_j^\mu, \qquad \eta_{i_1 \ldots i_k}^{(k)\mu} = \mathrm{D}_{i_k} \eta_{i_1 \ldots i_{k-1}}^{(k-1)\mu} - (\mathrm{D}_{i_k} \xi^j) u_{i_1 \ldots i_{k-1} j}^\mu, \tag{2.14}$$

for $\mu = 1, \ldots, m$, and $i, i_p = 1, \ldots, n$ for $p = 1, \ldots, k$.

A one-parameter Lie group of point transformations (2.11) acting on the independent and dependent variables of the DE system $\mathbf{R}\{x\,;\,u\}$ (2.1) defines a one-parameter Lie group of point symmetries if its kth extension (2.13) leaves invariant the solution manifold of $\mathbf{R}\{x\,;\,u\}$ [2,27,28]. The notion of point symmetries extends to higher-order local symmetries, nonlocal symmetries, and other related objects (see, e.g., [2,21,23,29–31]).

For families of DE systems, *classifications* of their analytical properties with respect to constitutive elements are of high theoretical and practical importance. Examples of such problems include local and nonlocal symmetry and conservation law classifications. Classification tables are usually presented modulo known equivalence transformations, i.e., only for the forms of constitutive functions and/or parameters that are not related by an equivalence transformation. Such classifications are available for multiple families of equations of mathematical physics; see, for instance, [2,30].

Equivalence transformations can be used directly to simplify symmetry determining equations during symbolic computations. A symbolic software package `SymmetryClassification` for `Maple` developed by Huang and Lisle includes a `detEqsForEquiv` routine that generates determining equations of point equivalence transformations. A reduced simplified version of the `rifsimp`-generated symmetry classification tree can consequently be obtained [9]. A related paper [32] contains detailed examples of point symmetry classifications for two nonlinear PDE families.

The following obvious relationships hold between the group of equivalence transformations of a DE family (2.1) and the groups of point symmetries of its members:

- A point symmetry (2.11) of a DE system $\mathbf{R}\{x\,;\,u\} \in \mathcal{F}_K$ is an equivalence transformation of the family $\mathcal{F}_K$ (2.1) if (2.11) is point symmetry for all systems in $\mathcal{F}_K$.
- An equivalence transformation (2.2) of the DE family $\mathcal{F}_K$ (2.1) is a point symmetry of its every member if and only if it does not involve components corresponding to the arbitrary elements of the family.

In practice, one-parameter Lie group of point symmetries are computed using the *invariance condition* (e.g., [1,2,27,28]): the transformation (2.11) is a point symmetry of a nondegenerate DE system $\mathbf{R}\{x\,;u\}$ (2.1) if and only if for each $\alpha = 1, \dots, N$,

$$\mathrm{X}^{(k)} R^\alpha(x, u, \partial u, \dots, \partial^k u) = 0, \tag{2.15}$$

whenever

$$R^\sigma(x, u, \partial u, \dots, \partial^k u) = 0, \quad \sigma = 1, \dots, N. \tag{2.16}$$

The computation of point symmetries from the determining equations (2.15) and (2.16) proceeds through the Lie's algorithm. The symmetry determining equations are obtained by evaluating (2.15) on the solution manifold of the given system (through the substitution of the DEs (2.16) and their differential consequences into the invariance condition (2.15), and split using the fact that the symmetry components ξ^i, η^μ do not depend on the derivatives of u. The application of the Lie's algorithm and its extensions to nontrivial DE systems often requires extensive algebraic manipulation. Indeed, split symmetry determining equations are linear overdetermined PDE systems for the unknown symmetry components ξ^i, η^μ; in practice, such systems may involve from a dozen to hundreds or thousands of equations. To treat such systems, symbolic computer software is commonly used [21,23]. In particular, methods based on differential Gröbner bases and the characteristic set method are successfully used in computer algebra packages [3,33–35]. A review of relevant work and symbolic software can be found in [2,36] and references therein.

Groups of equivalence transformations (2.2) of DE families can be systematically sought using the above Lie's algorithm, similarly to finding Lie groups of point symmetries, with arbitrary elements treated as dependent variables, and the dependence of the transformation components restricted accordingly. Details are given in Section 2.4.

2.3. Extensions of the notion of equivalence transformations

Definition 2.2 can be extended in several different ways. Details and further examples of such extensions can be found in [3,4,7,37,38] and references therein, and are out of scope of this paper. We briefly review several generalizations of equivalence transformations that are relevant to systematic symbolic computations.

A general approach to the extension of the notion of one-parameter Lie groups of point equivalence transformations (2.2) can be formulated in terms of seeking transformations of the form

$$\begin{aligned} (x^*)^i &= f^i[x, u, K], \quad i = 1, \dots, n, \\ (u^*)^\mu &= g^\mu[x, u, K], \quad \mu = 1, \dots, m, \\ (K^*)^\ell &= G^\ell[x, u, K], \quad \ell = 1, \dots, L, \end{aligned} \tag{2.17}$$

that map PDE systems in the family $\mathcal{F}_K$ into PDE systems in the same family. Generally, transformations (2.17) may be neither point, nor local transformations; the bracket notation in (2.17) denotes possible local or nonlocal dependencies of the transformation components on the independent and dependent variables and arbitrary elements, as well as, possibly, on derivatives of the dependent variables.

The following specific instance of general transformations (2.17) has been considered in the literature.

(A) Generalized equivalence transformations. An extension of one-parameter Lie groups of point equivalence transformations (2.2), in particular, Lie groups with the corresponding infinitesimal generators

$$\mathrm{X} = \xi^i(x, u, K)\frac{\partial}{\partial x^i} + \eta^\mu(x, u, K)\frac{\partial}{\partial u^\mu} + \theta^\nu(x, u, K)\frac{\partial}{\partial K^\nu}, \tag{2.18}$$

were defined in [4] (cf. [3]), and are referred to as groups of "generalized equivalence transformations". The essential difference here is that the transformation components for the independent and/or dependent variables $\xi^i(x, u, K)$, $\eta^\mu(x, u, K)$ may additionally depend on arbitrary elements K. Examples of such transformations are known in the literature. For example, they were computed in [39] for a class of nonlinear (1+1)-dimensional Schrödinger equations with a potential $V(x, t)$ and a power nonlinearity $\gamma = \text{const}$ given by

$$i\psi_t + \psi_{xx} + |\psi|^\gamma \psi + V(x, t)\psi = 0.$$

For an infinite-parameter group of generalized equivalence transformations derived in that work, transformations for the dependent variable $\psi(x, t)$ depend on the arbitrary constant parameter γ.

Remark 2.6. It is also worth mentioning that reliable examples of Lie groups of generalized equivalence transformations (2.18) where the components $\xi^i(x, u, K)$ and/or $\eta^\mu(x, u, K)$ of the independent and/or dependent variables would depend on *arbitrary functions* present in the given system, rather than *arbitrary constant parameters*, have not appeared in the literature that has been available up to this day. The example given in the original work [4] does not seem to be correct, as can be verified by a direct substitution. The first and only example of the indicated kind the author is aware of is due to Opanasenko, Bihlo and Popovych, appearing in their yet unpublished work [40]. Consider the KdV-Burgers PDE family for the unknown function $u(x, t)$ given by

$$u_t + Cuu_x = \sum_{k=0}^{r} A_k u_k + B, \tag{2.19}$$

where $u_k \equiv \partial^k u/\partial x^k$, $k = 1, \dots, r$, $r \geq 2$ is a fixed integer, and $A_k = A_k(x, t)$, $B = B(x, t)$, $C = C(x, t)$ are constitutive functions (arbitrary elements). The work [40] contains several statements concerning the structure of the generalized equivalence transformation *groupoid* of the PDE family (2.19). In particular, equivalence transformations may be used to invertibly map any PDE within (2.19) into one with $A_r = 1$ and $A_1 = 0$. Moreover, in [40], it is shown that when $A_r = 1$ and $A_1 = 0$, the family (2.19) admits the equivalence transformations with the transformed variables given by

$$x_* = F_1 x + F_0, \qquad t_* = G, \qquad u_*(x_*, t_*) = U_1 u + U_0, \tag{2.20}$$

where $F_0 = F_0(t)$, $G = G(t)$ and $U_1 = U_1(t)$ are arbitrary functions, $F_1 = F_1(t)$ is related to $G(t)$, U_0 is given by

$$U_0 = U_0(x, t) = \frac{F_1'x + F_0'}{F_1 C} U_1, \tag{2.21}$$

and the transformations for A_k, B, C are given by complex formulas found in [40].

Importantly, the transformation of the dependent variable $u(x, t)$ (2.21) involves the arbitrary element $C(x, t)$. These transformations are interpreted as the generalized equivalence group of the PDE family (2.19) if one formally extends the set of arbitrary elements to include the time derivative C_t as well as x-derivatives of C of orders $1, \dots, r$. The transformations (2.20) do not arise as point generalized equivalence transformations of the PDE family (2.19) without such prolongation.

The above example is readily verified using GeM symbolic software discussed in this contribution (see Sections 3 and 4.3).

Remark 2.7. One may also note that it is possible for the equivalence transformations to involve constant arbitrary elements, and therefore belong to the class (2.17), even when these transformations are point equivalence transformations (2.2). An instance of such a transformation is in fact contained in Example 2.5. Indeed, in the equivalence transformations (2.7), the choice $A_3 = a$ in (2.9) corresponds to a generalized equivalence transformation, since a is a constant constitutive parameter.

(B) Generalized extended equivalence transformations. One can consider further extensions of the notion of equivalence transformations, for example, the ones involving nonlocal transformations of variables and/or constitutive components. For example, *generalized extended equivalence groups* include transformations of the form (2.17), where the expressions for the transformed quantities are not necessarily point, but become a point transformation with respect to the independent and dependent variables after fixing the values of the arbitrary elements. Examples of transformations (2.17) involving arbitrary functions that arise as solutions of ordinary differential equations are contained, for instance, in [7]; we present an example from that paper below.

Example 2.8. Consider the PDE family

$$f(x)u_t = (g(x)u_x)_x + h(x)u^m, \tag{2.22}$$

where $f(x)$, $g(x)$, $h(x)$ are arbitrary smooth constitutive functions satisfying $f(x)g(x)h(x) \neq 0$, and m is a constant constitutive parameter. It is shown in [7], for example, that the transformations

$$\begin{gathered} x^* = \phi(x), \quad t^* = \delta_1 t + \delta_2, \quad u^* = \psi(x)\, u, \quad m^* = m, \\ f^* = \frac{\delta_0\, \delta_1}{\phi_x\, \psi^2} f, \quad g^* = \frac{\delta_0\, \phi_x}{\psi^2} g, \quad h^* = \frac{\delta_0}{\phi_x\, \psi^{m+1}} h \end{gathered} \tag{2.23}$$

map the PDE family (2.22) into itself, for any choice of smooth $\phi(x) \neq 0$, constants δ_0, δ_1, δ_2 satisfying $\delta_0\, \delta_1 \neq 0$, if $\psi(x)$ is an arbitrary solution of an ODE

$$\left(\frac{g\, \psi_x}{\psi^2}\right)_x = 0.$$

Generalized extended equivalence transformations similar to (2.23) have been extensively used in symmetry and conservation law classification problems in [7,37,41,42] and other works of the same group. In fact, the following early example of generalized extended equivalence transformations appears in [43], and is discussed in detail in [3].

Example 2.9. Consider a family of PDE systems of nonlinear telegraph equations given by

$$v_t = u_x, \qquad v_x = c^2(u)\, u_t + b(u), \tag{2.24}$$

with dependent variables $u = u(x, t)$, $v = v(x, t)$ and two arbitrary functions $b(u)$, $c(u)$. Then for any fixed value of the parameter ε, the change of variables

$$\begin{aligned} &x^* = x - \varepsilon v, \quad v_t = u_x, \qquad t^* = t - \varepsilon \int_{u_0}^{u} \frac{c^2(z)}{1 - \varepsilon b(z)}\, dz \\ &u^* = u, \qquad\quad v^* = v \end{aligned} \tag{2.25}$$

maps each nonlinear telegraph system to another system from the same family, with new arbitrary functions given by

$$b^*(u^*) = \frac{b(u)}{1 - \varepsilon b(u)}, \qquad c^*(u^*) = \frac{c(u)}{1 - \varepsilon b(u)}. \tag{2.26}$$

The transformations defined by (2.25), (2.26) are nonlocal transformations that do not form a Lie group of point transformations in the (x, t, u, v) space. Indeed, the transformed quantities x^*, t^*, u^*, v^* depend not only on the values of the independent and dependent variables x, t, u, v, but also on an *a priori* specification of the arbitrary elements $b(u)$, $c(u)$, and hence belong to the class of generalized extended equivalence transformations.

Remark 2.10. From the computational point of view, it is important to remark that in general, the generalized extended equivalence transformations do not form a Lie group of point transformations even in the space of variables extended by arbitrary elements. For example, the transformations defined by (2.25) and (2.26) do not form a Lie group in the space of (x, t, u, v, b, c), and hence do not belong to the set of generalized equivalence transformations (2.18).

(C) Potential equivalence transformations. Another example of transformations which are outside of the class of local Lie groups of equivalence transformations (2.2) is given by potential equivalence transformations. Such nonlocal transformations can arise as Lie groups of point equivalence transformations of a *potential system*. Consider a family of nonlinear wave equations in (1+1) dimensions [2,8] given by

$$u_{tt} = (c^2(u)u_x)_x, \tag{2.27}$$

for the unknown $u = u(x, t)$, with an arbitrary constitutive function $c(u)$. One can show that the infinitesimal generators of the group of point equivalence transformations of the family (2.27) are given by

$$\begin{aligned} &\mathrm{Z}_1 = \frac{\partial}{\partial t}, \quad \mathrm{Z}_2 = \frac{\partial}{\partial x}, \quad \mathrm{Z}_3 = \frac{\partial}{\partial u}, \quad \mathrm{Z}_4 = t\frac{\partial}{\partial u},\\ &\mathrm{Z}_5 = u\frac{\partial}{\partial u}, \quad \mathrm{Z}_6 = t\frac{\partial}{\partial t} + x\frac{\partial}{\partial x}, \quad \mathrm{Z}_7 = t\frac{\partial}{\partial t} - c\frac{\partial}{\partial c}, \end{aligned} \tag{2.28}$$

and the global group has the form

$$x^* = a_6x + a_2, \quad t^* = a_6a_7t + a_1, \quad u^* = a_5u + a_4t + a_3, \quad c^*(u^*) = a_7^{-1}c(u), \tag{2.29}$$

where $a_1, \dots, a_7$ are arbitrary constants with $a_5\, a_6\, a_7 \neq 0$. Additional equivalence transformations arise as local equivalence transformations of a PDE system nonlocally related to (2.27) [2,8,44,45]. In particular, for an arbitrary $c(u)$, PDEs of the family (2.27) admit local conservation laws corresponding to multipliers $1, t, x, xt$. For the multiplier t, the conservation law is

$$\frac{\partial}{\partial t}(tu_t - u) - \frac{\partial}{\partial x}(tc^2(u)u_x) = 0. \tag{2.30}$$

The corresponding family of potential systems with a nonlocal variable $w(x, t)$, is given by

$$w_x = tu_t - u, \qquad w_t = tc^2(u)u_x; \tag{2.31}$$

every PDE of this family is nonlocally related and equivalent (in a non-invertible manner) to a PDE of the family (2.27). A computation of Lie groups of point equivalence transformations of the family of potential systems (2.27) yields the eight-dimensional Lie algebra of generators

$$\begin{aligned} &\mathrm{W}_i = \mathrm{Z}_i, \quad i = 2, 4, 7,\\ &\mathrm{W}_1 = \frac{\partial}{\partial w}, \qquad \mathrm{W}_3 = \mathrm{Z}_3 - x\frac{\partial}{\partial w},\\ &\mathrm{W}_5 = \mathrm{Z}_5 - t\frac{\partial}{\partial t} - x\frac{\partial}{\partial x}, \qquad \mathrm{W}_6 = \mathrm{Z}_6 + w\frac{\partial}{\partial w},\\ &\mathrm{W}_8 = tu\frac{\partial}{\partial t} + w\frac{\partial}{\partial x} + u^2\frac{\partial}{\partial u} - 2uc\frac{\partial}{\partial x}. \end{aligned} \tag{2.32}$$

In particular, the time translation Z_1 disappears, and two new generators, the obvious W_1 and an additional generator W_8, appear as local equivalence transformation generators of the potential system (2.31).

The global group has a rather complicated form; it is simpler to write it in two parts. The one corresponding to the first seven generators is given by

$$\begin{aligned} x^* &= A_5^{-1}A_6x + A_2, \quad t^* = A_5^{-1}A_6A_7t, \quad u^* = A_5u + A_4t + A_3,\\ w^* &= A_6w - A_3A_5^{-1}A_6x + A_1, \quad c^*(u^*) = A_7^{-1}c(u) \end{aligned} \tag{2.33}$$

(cf. [2,8]), where $A_1, \dots, A_7$ are arbitrary constants with $A_1A_2A_3 \neq 0$, corresponding to the generators $\mathrm{W}_1, \dots, \mathrm{W}_7$. The global group corresponding to the infinitesimal generator W_8 yields a one-parameter group of transformations

$$\begin{aligned} &x^* = x - Bw, \qquad t^* = \frac{t}{1 + Bu}, \qquad u^* = \frac{u}{1 + Bu},\\ &w^* = w, \qquad c^*(u^*) = (1 + Bu)^2\, c(u) \end{aligned} \tag{2.34}$$

with a parameter B. The equivalence transformations (2.34) are nonlocal projective-type transformations of the nonlinear wave equation family (2.27); they have been employed in a symmetry classification problem [8]. Further examples of nonlocal equivalence transformation can be found in, e.g., [2,3].

In Section 4.2, the symbolic software package `GeM` is used to compute the eight generators (2.32) and the corresponding global groups (2.33), (2.34).

2.4. Systematic computation of Lie groups of generalized equivalence transformations

For a given family $\mathcal{F}_K$ of DE systems $\mathbf{R}\{x\,; u; K\}$ (2.1), the following algorithm can be used for the computation of (generalized) point equivalence transformations (2.2) and (2.18).

If the set K involves arbitrary functions, it is assumed that these are functions of only the components of x, u, and are not functions of, for example, derivatives of u.

1. Replace the constitutive functions and/or parameters $K = (K^1, \dots, K^L)$ by new dependent variables $(K^1(x), \dots, K^L(x))$. Thus consider a new DE system $\widetilde{\mathbf{R}}\{x\,; u, K\}$ with $m + L$ dependent variables and no arbitrary elements.

2. Seek point symmetries of $\widetilde{\mathbf{R}}\{x\,;\,u,K\}$, with infinitesimal generators

$$\mathrm{X} = \xi^i(x,u,K)\frac{\partial}{\partial x^i} + \eta^\mu(x,u,K)\frac{\partial}{\partial u^\mu} + \theta^\lambda(x,u,K)\frac{\partial}{\partial K^\lambda}. \tag{2.35}$$

3. Obtain the *split system of determining equations* for $\widetilde{\mathbf{R}}\{x\,;\,u,K\}$, as outlined in Section 2.2.
4. If the arbitrary elements K of the original family $\mathbf{R}\{x\,;\,u;\,K\}$ (2.1) of DE systems contained arbitrary functions, introduce restrictions of the form

$$\frac{\partial \xi^i(x,u,K)}{\partial K^\gamma} = 0, \quad \frac{\partial \eta^\mu(x,u,K)}{\partial K^\delta} = 0, \quad \frac{\partial \theta^\lambda(x,u,K)}{\partial x^j} = 0, \tag{2.36}$$

etc., as appropriate, to exclude the dependence of transformation components of the arbitrary elements on variables they do not depend on, as well as the dependence of transformation components of the variables of the system on inappropriate arbitrary elements. For example, for the DE family (1.1), the infinitesimal generator of the generalized equivalence transformations has the form

$$\mathrm{X} = \xi(x,t,u,c)\frac{\partial}{\partial x} + \tau(x,t,u,c)\frac{\partial}{\partial t} + \eta(x,t,u,c)\frac{\partial}{\partial u} + \theta(x,t,u,c)\frac{\partial}{\partial c}, \tag{2.37}$$

and the transformation for $c(u)$ must not explicitly depend on the variables x, t. Therefore the restrictions on the component θ are given by

$$\frac{\partial \theta(x,t,u,c)}{\partial x} = \frac{\partial \theta(x,t,u,c)}{\partial t} = 0. \tag{2.38}$$

5. In order to simplify computations, additional restrictions can be introduced at this stage, for example,

$$\frac{\partial \xi^i(x,u,K)}{\partial K^\gamma} = 0, \quad i=1,\dots,n, \quad \gamma = 1,\dots,L, \tag{2.39}$$

if the transformations for the independent variables are assumed to be independent of the arbitrary elements.
6. Append all restrictions, in the form of linear PDEs, to the split system of determining equations.
7. Simplify and solve the augmented split system of determining equations, to find the infinitesimal generators (4.7) of the equivalence transformations.
8. Integrate to obtain the global group. For each infinitesimal generator (4.7), the corresponding one-parameter Lie group of equivalence transformations (2.2) is found through the solution of the initial-value problem

$$\begin{aligned}
&\frac{d}{d\varepsilon}(x^*)^i = \xi^i(x^*,u^*,K^*), \quad i=1,\dots,n,\\
&\frac{d}{d\varepsilon}(u^*)^\mu = \eta^\mu(x^*,u^*,K^*), \quad \mu=1,\dots,m,\\
&\frac{d}{d\varepsilon}(K^*)^\lambda = \theta^\lambda(x^*,u^*,K^*), \quad \lambda=1,\dots,L,\\
&(x^*)^i|_{\varepsilon=0} = x^i, \quad (u^*)^\mu|_{\varepsilon=0} = u^\mu, \quad (K^*)^\lambda|_{\varepsilon=0} = K^\lambda,
\end{aligned} \tag{2.40}$$

where ε is the group parameter.

Remark 2.11. The above procedure is readily generalized to the case when the given DE system involves both an arbitrary function and its derivative(s) (for example, $c(u)$ and $c'(u)$). In such cases, in Step 1 of the above algorithm, both the arbitrary function at hand and its derivative(s) can be replaced by different additional dependent variables (for example, $C(x,t)$ and $D(x,t)$).

3. Symbolic computation of equivalence transformations

The symbolic software package `GeM` for `Maple` contains stable routines for point and local (higher-order) symmetry analysis of systems of ordinary and partial differential equations, routines for local conservation law analysis, and other related functions. In conjunction with the standard `Maple` routine `rifsimp` for the reduction of overdetermined systems and case splitting, the `GeM` package has been used to study multiple problems involving conservation law and symmetry analysis and classification. For details on the available routines, additional options, methods, background, and examples, see [21,23,46].

It is worth noting that the recent versions of `Maple` software itself, in particular, `Maple 2016`, include a number of built-in routines for symmetry computations, including `DeterminingPDE`, `Infinitesimals`, `InfinitesimalGenerator`, within the standard `PDEtools` package; it will take time to assess the stability, completeness, and performance of these routines with DE systems of practical interest.

The `GeM` package uses an efficient representation of the DEs and resulting symmetry/conservation law determining equations, through the treatment of dependent variables and their derivatives as `Maple` symbols instead of functions or expressions. For instance, the `Maple` representation of a function $G(x,y,z)$ and its partial derivative by G_x in `GeM` routines is respectively `G` and `Gx`. Consequently, for a composite function, differentiations significantly simplify; e.g., if $H = H\,(x,y,G(x,y,z),G_x(x,y,z))$, then in the `GeM` representation, one has `H` = `H(x, y, G, Gx)`; its partial derivative by y is represented as

$$\begin{aligned}\frac{\partial}{\partial y}H\,(x,y,G(x,y,z),G_x(x,y,z)) &= \texttt{diff(H(x,y,G,Gx),y)}\\ &\quad \texttt{+ diff(H(x,y,G,Gx),G)* Gy+diff(H(x,y,G,Gx),Gx)* Gxy},\end{aligned} \tag{3.1}$$

not requiring the use of composite functions or multi-level `Maple` index notation.

In the implementation of recent versions of the GeM package, its author has made a conscious choice to move away from the standard Maple package structure, keeping GeM a collection of routines and variable definitions. This structure provides direct access to all variables and internal routines for an advanced user, while the chosen naming convention prevents the clash with user variables and functions.

The GeM package can be used to automate computations of Lie groups of point and generalized equivalence transformations. The sequence of steps for such computations, following the algorithm of Section 2.4, is outlined below. We note that the algorithm described in the current work can be applied to a given family of DE systems (2.1) as it stands, or to a related system (such as a potential system or a nonlocally related subsystem [44]), in order to seek more general (for example, nonlocal) equivalence transformations.

In the subsequent section, examples of symbolic computations of equivalence transformations using Maple and GeM are presented.

Step A. Initialization. Clear the variables. Initialize the package. If required, also initialize a sub-package for global group computations.

```
restart :
read("d : /gem32_12.mpl") :                                        (3.2)
read("d : /gem_globgroup_and_equiv.txt") :
```

Step B. Declare variables. The user creates a collection of independent variables called, for example, ind, and a collection of dependent variables and constitutive elements (which for the purpose of equivalence transformation computations are treated as dependent variables) called, for example, dep, and declares them.

```
gem_decl_vars(indeps = [ind], deps = [dep]);
```

The readability of names of variables and their derivatives is enhanced when one employs small Latin letters to denote the independent variables, and capital Latin letters for the dependent variables (cf. (3.1)).

Step C. Declare the DEs. The user defines differential equations named, for example, Eq1 and Eq2, and declares them.

```
gem_decl_eqs([Eq1, Eq2], solve_for = [LeadDer]);
```

The equations need to be solvable algebraically with respect to the specified leading derivatives LeadDer specified in the list solve_for. Within the same routine, necessary differential consequences of the DEs are automatically computed and solved with respect to the differential consequences of the leading derivatives.

Step D. Generate the determining equations. Next, one generates the equivalence transformation determining equations (Section 2.4, Step 3), using the command

```
det_eqs := gem_symm_det_eqs([ind, dep]);
```

where the list [ind,dep] denotes the common dependence of all components of the generator (4.7).

The routine gem_symm_det_eqs outputs a set containing the overdetermined split linear system of determining equations for the equivalence transformation tangent vector field components $\xi^i(x, u, K)$, $\eta^\mu(x, u, K)$, $\theta^\lambda(x, u, K)$. In the GeM notation, these components are denoted by symbols like xi_x (for an independent variable x) and eta_U (for a independent variable u).

In order to list the tangent vector field components, one can use the routine

```
OFF :
sym_components := gem_symm_components();
ON :
```

Here the OFF: and ON: commands are used to switch off and on the output suppression of the function dependencies in Maple. It is done here to force Maple to print the equivalence transformation components explicitly, with their dependencies.

Step E. Restrict dependencies. As per Section 2.4, Step 4, one can restrict the sets of variables on which the equivalence transformation tangent vector field components depend, by appending linear differential equations like (2.36) to the determining equations. Restriction equations are generated using the command

```
restriction_eqs := gem_generate_EquivTr_dependence([
    [[<variables1>], [<dep1>]],
    [[<variables2>], [<dep2>]],
    ...,
]);
```

This will cause the equivalence transformation component(s) listed in a comma-separated sequence <variables1> to depend on variables listed in a comma-separated sequence <dep1>, etc. The output stored in the variable restriction_eqs contains a Maple set of linear differential restrictions in the form (2.36).

The augmented split set of equivalence transformation determining equations is then constructed using a set union

```
det_eqs_all := det_eqs union restriction_eqs :
```

Step F. Rif-simplify the augmented determining equations. At this optional step, one uses the rifsimp routine [33] for the Gröbner basis-based reduction of the overdetermined linear system of determining equations obtained at the previous step.

```
simplified_eqs := DEtools[rifsimp](det_eqs_all, sym_components,
    casesplit, mindim = 1);
```

The current step is particularly important for large systems of determining equations, since through the elimination of the redundancy, it commonly leads to a drastic simplification and reduction of the number of PDEs in the system. The output of rifsimp is a Maple table, with the simplified set of equations found in simplified_eqs[Solved].

The rifsimp routine is capable of the solution space dimension prediction (when used with mindim = 1 option), and case splitting/classification (when used with casesplit option); for details, see [21,23,46] or Maple help.

Step G. Solve the determining equations. The solution as a set is obtained, for example, by Maple pdsolve routine:

```
eq_tr_sol := pdsolve(simplified_eqs[Solved], sym_components);
```

Though for general nonlinear PDE systems, pdsolve is known to sometimes return incomplete/non-general solution, this routine is one of the best ones in its class, and is rather efficient for linear determining equations. In particular, the dimension of the solution set (the number of free constants _Ci in eq_tr_sol) can be compared with simplified_eqs[dimension] from Step F.

Step H. Output the equivalence transformation generators. For this purpose, one uses the following GeM routine, which outputs both the generators and the respective free constants and/or free functions in the general solution eq_tr_sol.

```
gem_output_symm(eq_tr_sol);
```

Step I. Generate global group(s) of equivalence transformations. In order to generate the global group of equivalence transformations corresponding to a given infinitesimal generator within the general solution eq_tr_sol, one "extracts" the corresponding generator. For example, in order to extract the transformation corresponding to _C6, one calls the routine

```
Particular_Transform := gem_extract_symm(sym_sol, spec = {_C6 = 1});
```

(This routine sets free constants _Ci within eq_tr_sol not mentioned in spec to zero.) Then the Lie theorem ODEs (2.40) are generated,

```
Particular_Transform_ODEs := gem_global_group(
    Particular_Transform, group_param_name ='a');
```

and solved:

```
Particular_Transform_Global := dsolve(Particular_Transform_ODEs);
```

The procedure of Step I can be repeated for the remaining linearly independent equivalence transformation generators. In some cases, in particular, when the local equivalence transformation group is not too large, it may be beneficial to skip the gem_extract_symm step, and generate the global group ODEs and integrate them for the general solution eq_tr_sol instead of the one given by Particular_Transform.

Remark 3.12. We note that the order and/or the form of solutions obtained in Steps G, H, I may differ in different versions of Maple, and also depending on the naming of variables a user employs. This has to do with sorting of variables and the introduction of arbitrary constants by pdsolve. The final solution (i.e., the infinitesimal generators) will, in any case, provide an equivalent basis of the Lie algebra of infinitesimal generators for the sought generalized equivalence transformations.

Remark 3.13. From the computational point of view, it is important to remark that in general, the generalized extended equivalence transformations do not form a Lie group of point transformations even in the space of variables extended by arbitrary elements. For example, the transformations defined by (2.25) and (2.26) do not form a Lie group in the space of (x, t, u, v, b, c), and hence do not belong to the set of generalized equivalence transformations (2.18). It follows that the systematic procedure described in a current section will generally miss generalized extended equivalence transformations.

4. Examples of symbolic computation of equivalence transformations

4.1. Equivalence transformations of the dimensional KdV family

We begin with a simple example of computing Lie groups of equivalence transformations of the dimensional KdV family (2.5), according to the sequence of steps presented in Section 3. The program code can have, for example, the following form.

Initialization: same as (3.2).

Declaration of variables:

```
ind := x, t;
dep := U(ind), A(ind), B(ind), Q(ind);
gem_decl_vars(indeps = [ind], deps = [dep]);
```

(capital letters are used for the parameters a, b, q in (2.5)).

Declaration of the PDE family, solved for the leading derivative u_t:

```
KdV family := diff(U(ind),t)+A(ind)* diff(U(ind),x)
    + B(ind)* U(x,t)* diff(U(ind),x)+Q(ind)* diff(U(ind),x,x,x)=0;
gem_decl_eqs([KdV family],solve_for = [diff(U(ind),t)]);
```

Computation of the split system of equivalence transformation determining equations, using ind,dep declared above, and output of infinitesimal components:

```
det_eqs := gem_symm_det_eqs([ind, dep]);
OFF : sym_components := gem_symm_components(); ON :
```

This step returns 182 determining equations stored in det_eqs.

Restriction equations are obtained by demanding that the infinitesimal equivalence transformation components for the constitutive parameters a, b, q may only depend on those parameters themselves.

```
restriction_eqs := gem_generate_EquivTr_dependence([
    [[x, t], [x, t, U, A, B, Q]],
    [[U], [x, t, U]],
    [[A, B, Q], [A, B, Q]]
]);
```

The full set of determining equations is created and simplified as follows:

```
det_eqs_all := det_eqs union restriction_eqs :
simplified_eqs := DEtools[rifsimp](det_eqs_all, sym_components,
    casesplit, mindim = 1);
```

This step results in 36 simplified linear PDEs (use `linalg[vectdim](simplified_eqs[Solved]);`) on the equivalence transformation components. The dimension of the solution set of this system, i.e., the dimension of the Lie algebra of infinitesimal generators of equivalence transformations in the chosen form, is contained in the variable `simplified_eqs[dimension]` and equals 7.

The equivalence transformation components are solved for and displayed using the commands

```
eq_tr_sol := pdsolve(simplified_eqs[Solved], sym_components);
gem_output_symm(eq_tr_sol);
```
(4.1)

which outputs seven generators corresponding, up to scaling constants, to (2.6) (sorted here according to the output expression length as a text string):

$$
\begin{aligned}
\_C6, \quad & X_1 = \frac{\partial}{\partial x},\\
\_C7, \quad & X_2 = \frac{\partial}{\partial t},\\
\_C2, \quad & X_3 = \frac{\partial}{\partial u} - b\frac{\partial}{\partial a},\\
\_C5, \quad & X_4 = t\frac{\partial}{\partial x} + \frac{\partial}{\partial a},\\
\_C3, \quad & X_5 = \frac{1}{2}t\frac{\partial}{\partial t} + \frac{1}{2}x\frac{\partial}{\partial x} + q\frac{\partial}{\partial q},\\
\_C1, \quad & X_6 = -\frac{3}{2}t\frac{\partial}{\partial t} - \frac{1}{2}x\frac{\partial}{\partial x} + u\frac{\partial}{\partial u} + a\frac{\partial}{\partial a},\\
\_C4, \quad & X_7 = -\frac{3}{2}t\frac{\partial}{\partial t} - \frac{1}{2}x\frac{\partial}{\partial x} + a\frac{\partial}{\partial a} + b\frac{\partial}{\partial b}.
\end{aligned}
$$

In order to generate a global group of equivalence transformations corresponding, for example, to the constant A_7 in (2.7) (integration constant _C4 above), one uses the commands

```
Particular_Transform := gem_extract_symm(sym_sol, spec = {_C4 = 1});
Particular_Transform_ODEs := gem_global_group(
    Particular_Transform, group_param_name ='s');
Particular_Transform_Global := dsolve(Particular_Transform_ODEs);
```

and obtains the global transformation

```
A(s) = A_0 * exp(s), B(s) = B_0 * exp(s), Q(s) = Q_0, U(s) = U_0,
t(s) = t_0 * exp(−(3/2) * s), x(s) = x_0 * exp(−(1/2) * s)
```
(4.2)

with the group parameter s, and the values t_0, x_0, etc., corresponding to the values of the variables before the transformation. The constant A_7 in (2.7) is identified with $e^{s/2}$ in (4.2), `B_0` of (4.2) with b in (2.7), `B(s)` with b^*, etc. (For details, see `E_g_01_equiv_tr_KdV_dim_ABQ.mw`.)

4.2. Nonlocal equivalence transformations of a family of nonlinear wave equations

Consider a family (2.27) of (1+1)-dimensional nonlinear wave equations for $u(x, t)$, involving an arbitrary constitutive function $c(u)$. Local equivalence transformations of (2.27) are given by (2.29) with infinitesimal generators (2.28).

The PDE (2.27) admits a local conservation law (2.30), and consequently a potential system (2.31), equivalent but nonlocally related to the wave equation (2.27) for every $c(u)$ [44,45].

We now use the algorithm presented in Section 3 to compute the point equivalence transformations (2.32) of the potential system (2.31).

After the initialization (3.2), the program code is as follows. (Note that the extended list of "dependent variables" for the computation includes the constitutive function $c(u)$ denoted by the capital C.)

```
ind := x, t;
dep := U(ind), W(ind), C(ind);
gem_decl_vars(indeps = [ind], deps = [dep]);
```

The equations are declared as follows.

```
UW1 := diff(W(ind), x) = t * diff(U(ind), t) − U(ind);
UW2 := diff(W(ind), t) = t * C(ind)^2*diff(U(ind),x);
gem_decl_eqs([UW1, UW2], solve_for = [lhs(UW1), lhs(UW2)]);
```

One may seek Lie groups of generalized equivalence transformations by requiring that the component for the constitutive function $c(u)$ is independent of x, t, w.

```
    det_eqs := gem_symm_det_eqs([ind, dep]);
    OFF : sym_components := gem_symm_components(); ON :
    restriction_eqs := gem_generate_EquivTr_dependence([
        [[x, t, U, W], [x, t, U, W, C]],
        [[C], [U, C]]]);
    det_eqs_all := det_eqs union restriction_eqs :
    simplified_eqs := DEtools[rifsimp](det_eqs_all, sym_components,
        casesplit, mindim = 1);
```

This leads to 25 simplified linear determining equations, which are subsequently solved to obtain an eight-dimensional Lie algebra of the equivalence transformation infinitesimal generators.

```
    eq_tr_sol := pdsolve(simplified_eqs[Solved], sym_components);
    gem_output_symm(eq_tr_sol);
```

The output (in Maple 18; may slightly vary across versions) is

$$\begin{array}{rr}
\_C3, & \mathrm{X}_1 = \dfrac{\partial}{\partial w}, \\
\_C8, & \mathrm{X}_2 = \dfrac{\partial}{\partial x}, \\
\_C4, & \mathrm{X}_3 = t\,\dfrac{\partial}{\partial u}, \\
\_C1, & \mathrm{X}_4 = x\dfrac{\partial}{\partial w} - \dfrac{\partial}{\partial u}, \\
\_C7, & \mathrm{X}_5 = t\,\dfrac{\partial}{\partial t} - c\,\dfrac{\partial}{\partial c}, \\
\_C2, & \mathrm{X}_6 = t\,\dfrac{\partial}{\partial t} + x\,\dfrac{\partial}{\partial x} + w\,\dfrac{\partial}{\partial w}, \\
\_C5, & \mathrm{X}_7 = u\,\dfrac{\partial}{\partial u} - t\,\dfrac{\partial}{\partial t} - x\,\dfrac{\partial}{\partial x}, \\
\_C6, & \mathrm{X}_8 = -2cu\,\dfrac{\partial}{\partial c} + u^2\,\dfrac{\partial}{\partial u} + tu\,\dfrac{\partial}{\partial t} + w\,\dfrac{\partial}{\partial x},
\end{array} \tag{4.3}$$

which (up to renumbering and rescaling) coincides with the generators listed in the formulas (2.32).

As an illustration, we now compute a global group of equivalence transformations corresponding to the projective transformation W_8 of (2.32), which corresponds to the integration constant _C6 in the output (5.5). One proceeds as follows.

```
Projective_Transf := gem_extract_symm(sym_sol, spec = {_C6 = −1});
Particular_Transform_ODEs := gem_global_group(
    Projective_Transf, group_param_name = 'B');
Particular_Transform_Global := dsolve(Particular_Transform_ODEs);
```

and obtains the output

```
x(B) = x_0-B*W_0, t(B)=t_0/(1+BU_0), U(B) = U_0/(1+BU_0),
w(B)=W_0, c(B)=c_0*(1+BU_0)^2
```

corresponding to the global transformation (2.34) with the group parameter B.

(See E_g_02_equiv_tr_waveUW.mw.)

4.3. Generalized equivalence transformations of the KdV–Burgers family

Consider a family (2.19) of (1+1)-dimensional KdV-Burgers equations involving $r + 3$ arbitrary elements that are functions of x, t. In [40], equivalence group computations were done through tedious algebraic manipulation. For the current computational example, following [40], we fix $r = 3$ and choose $A_1 = 0, A_3 = 1$. The explicit PDE family is then given by

$$u_t + Cuu_x = A_0u + A_2u_{xx} + u_{xxx} + B, \tag{4.4}$$

with four arbitrary elements $A_0(x, t), A_2(x, t), B(x, t), C(x, t)$. First we compute the Lie group of point generalized equivalence transformations of (4.4).

(a) Point equivalence transformations. After the initialization (3.2), the program code is as follows. The variables and the PDE family are declared:

```
ind := x, t;
dep := U(ind), A0(ind), A2(ind), B(ind), C(ind);                         (4.5)
gem_decl_vars(indeps = [ind], deps = [dep]);
```

```
BKdV := diff(U(ind),t)+C(ind)* U(ind)* diff(U(ind),x)=A0(ind)* U(ind)
   + A2(ind)* diff(U(ind),x,x)+diff(U(ind),x,x,x)+B(ind);               (4.6)
gem_decl_eqs([BKdV ],solve_for = [diff(U(ind),t)]);
```

We seek most general Lie groups of point generalized equivalence transformations with generators in the form

$$\mathrm{X} = \xi^x \frac{\partial}{\partial x} + \xi^t \frac{\partial}{\partial t} + \eta^u \frac{\partial}{\partial u} + \theta^{A_0} \frac{\partial}{\partial A_0} + \theta^{A_2} \frac{\partial}{\partial A_2} + \theta^{B} \frac{\partial}{\partial B} + \theta^{C} \frac{\partial}{\partial C}, \tag{4.7}$$

where the components θ^j corresponding to the arbitrary functions $A_0(x, t), A_2(x, t), B(x, t), C(x, t)$ may depend on x, t, A_0, A_2, B, C, and the components ξ^t, ξ^x and η^u corresponding to the variables may depend on x, t, u, A_0, A_2, B, C. The restriction equations become

```
restriction_eqs := gem_generate_EquivTr_dependence([
   [[C,A0,A2,B],[x,t,A0,A2,B,C]]]);                                     (4.8)
det_eqs_all := det_eqs union restriction_eqs :
```

```
simplified_eqs := DEtools[rifsimp](det_eqs_all,sym_components,
   casesplit,mindim = 1);                                               (4.9)
```

This leads to 36 simplified linear determining equations, which are subsequently solved and output using commands (4.1). The output (in `Maple` 18; may slightly vary across versions) is

$$\begin{aligned}
&\_\mathrm{C2}, \quad \mathrm{X}_1 = \frac{\partial}{\partial x},\\
&\_\mathrm{C3}, \quad \mathrm{X}_2 = \frac{\partial}{\partial t},\\
&\_\mathrm{F1}, \quad \mathrm{X}_3 = F_1'(t)\,\frac{\partial}{\partial A_0} + F_1(t)\left(u\frac{\partial}{\partial u} + B\frac{\partial}{\partial B} - C\frac{\partial}{\partial C}\right),\\
&\_\mathrm{C1}, \quad \mathrm{X}_4 = \frac{3}{2}\left(-t\frac{\partial}{\partial t} + A_0\frac{\partial}{\partial A_0} + B\frac{\partial}{\partial B}\right) + \frac{1}{2}\left(-x\frac{\partial}{\partial x} + A_2\frac{\partial}{\partial A_2}\right).
\end{aligned} \tag{4.10}$$

Here in addition to scalings (X_4) one has an equivalence transformation involving an arbitrary function of t, but no generalized equivalence transformations (2.20) and (2.21) found in [40]. For details, see an accompanying Maple file `E_g_03a_BKdV_point_eq_tr.mw`

(b) Generalized equivalence transformations. In order to compute the generalized transformations (2.20) and (2.21), one must allow the infinitesimal transformation components for the arbitrary element $B(x, t)$ to depend on C_t, C_x, C_{xx}, and C_{xxx} [40]. The above code in (4.5), (4.6) and (4.8) is modified as follows. First, additional dependent variables $C_t, \ldots, C_{xxx}$ are declared:

```
dep := U(ind), A0(ind), A2(ind), B(ind), C(ind),
CT(ind), CX(ind), CXX(ind), CXXX(ind);
```

Differential relationships between $C, C_t, \ldots, C_{xxx}$ are then added to the PDE system:

```
gem_decl_eqs([BKdV,diff(C(ind),t) = CT(ind),diff(C(ind),x) = CX(ind),
   diff(CX(ind),x) = CXX(ind),diff(CXX(ind),x) = CXXX(ind)],
solve_for = [diff(U(ind),t),diff(C(ind),t),diff(C(ind),x),
   diff(CX(ind),x),diff(CXX(ind),x)]);
```

Restriction equations are modified:

```
restriction_eqs := gem_generate_EquivTr_dependence([
   [[x,t],[x,t]],
   [[U],[x,t,U,C]],
   [[A0,A2,B,C,CT,CX,CXX,CXXX],[x,t,A0,A2,B,C,CT,CX,CXX,CXXX]]]);
```

Then the simplification and solution of the overdetermined system (4.9) takes place. The output includes and extends the generators given in (4.10). In particular, it includes an additional generalized equivalence transformation generator (4.7) with

$$\eta^u = \eta^u(x, t, u, C) = F_6(t)u + \frac{F_3'(t)x + F_7'(t)}{C},$$

featuring an explicit dependence on the arbitrary element C (cf. (2.20) and (2.21)). For details, see an accompanying Maple file `E_g_03b_BKdV_generalized_eq_tr.mw`.

5. Equivalence transformations of a family of nonlinear equations describing anti-plane shear motions of a fiber-reinforced material

The framework of incompressible hyper/viscoelasticity, in particular, its formulations that take into account possible anisotropy due to the presence of a fiber matrix, provide a practically important toolbox for the description of various classes of materials, including applications in medicine (biological membranes), cell biology (cell membranes), industry (textiles and ribbers), and more [25]. Partial differential equations describing finite deformations of such solids are force balance-type (3 + 1)-dimensional equations for the time evolution of Eulerian positions of Lagrangian points of the solid. The model is formulated through the definition of invariants and pseudo-invariants that may involve deformation tensors, deformation rates, and fiber directors. In the incompressible setting, in three dimensions, the model is given by four PDEs (three equations of motion and the incompressibility condition). The arbitrary elements present in such models are contained in the form of the strain energy density function W, and are given by contributions of one or more arbitrary functions describing various aspects of the material behavior. For example, the general form of the strain energy density in the incompressible hyperelastic setting is given by

$$W = W_{iso}(I_1, I_2) + W_{aniso}(I_4, \dots, I_9),$$

where W_{iso} and W_{aniso} are isotropic and anisotropic contributions, respectively; I_1, I_2 are invariants of the left and right Cauchy–Green deformation tensors (the third invariant $I_3 = J^2 = 1$, the unit Jacobian of the incompressible deformation tensor), and $I_4, \dots, I_9$ are pseudo-invariants involving fiber direction vectors (for details, see, e.g., [24,25], and references therein).

A basic yet important set of physical systems describing finite displacements of fiber-reinforced Mooney–Rivlin materials under the standard reinforcing model [24] is formulated in terms of the strain energy density

$$W = a(I_1 - 3) + b(I_2 - 3) + q(I_4 - 1)^2, \tag{5.1}$$

where $a,\ b,\ q > 0$ are constant material parameters.

In [24], a family of nonlinear wave equations has been derived, describing one-dimensional S-wave propagation dependent on the fiber direction, within the above framework:

$$G_{tt} = \left(\alpha + \beta \cos^2\gamma \left(3\cos^2\gamma\,(G_x)^2 + 6\ \sin\gamma\,\cos\gamma\ G_x + 2\sin^2\gamma\right)\right) G_{xx}. \tag{5.2}$$

Here $G(x, t)$ describes the finite displacement amplitude of anti-plane shear motions, in the material z-direction, of a nonlinear incompressible hyperelastic medium, reinforced by fibers located in the (x, z)-plane of the material frame. The fibers are assumed to make a constant angle γ with the material direction x. The model involves three arbitrary elements, the material parameters $\alpha = 2(a + b) > 0$, $\beta = 4q > 0$, and the material fiber angle $\gamma \in [0, \pi/2]$.

When $\gamma = \pm\pi/2$, i.e., the fibers are aligned along the material z-direction, and the PDEs (5.2) reduce to the classical linear wave equations $G_{tt} = \alpha G_{xx}$, in particular, fiber effects vanish. When the fiber angle is $\gamma = 0$, Eqs. (5.2) assume a simple form

$$G_t = (\alpha + 3\beta G_x^2)\,G_{xx}. \tag{5.3}$$

We now use the algorithm of Sections 3 and 2.4 to compute equivalence transformations of the PDE family (5.2). The PDEs involve three arbitrary elements; we assume $\gamma \neq \pm\pi/2$. For the sake of simplicity in `Maple` notation, we denote $\alpha = a$, $\beta = b$; moreover, in order to simplify computations, as we will see, it is beneficial to use the arbitrary element $k = \tan\gamma \in \mathbb{R}$ instead of γ itself. The corresponding infinitesimal generators of point equivalence transformations consequently assume the form

$$\mathrm{X} = \xi\frac{\partial}{\partial x} + \tau\frac{\partial}{\partial t} + \eta\frac{\partial}{\partial G} + \theta^1\frac{\partial}{\partial a} + \theta^2\frac{\partial}{\partial b} + \theta^3\frac{\partial}{\partial k}. \tag{5.4}$$

For the current application, it is sufficient to consider regular equivalence transformations (2.2), so the dependencies of the infinitesimal components are chosen as follows:

$$\xi = \xi(x, t, G), \qquad \tau = \tau(x, t, G), \qquad \eta = \eta(x, t, G),$$

$$\theta^\nu = \theta^\nu(a, b, k), \quad \nu = 1, 2, 3.$$

The program sequence for equivalence transformation computations proceeds as follows. The `Maple` modules are initialized according to (3.2). The PDE family (5.2), independent variables, the dependent variable, and arbitrary elements are defined as variables of the problem.

```
PDE1:= diff(G(x,t), t, t) = (a + b*cos(g)^2*
    (3*cos(g)^2*diff(G(x,t), x)^2 +6*sin(g)*cos(g)*diff(G(x,t), x)
     + 2*sin(g)+ ^2))* diff(G(x,t), x, x);


    ind := x, t;
    a := A(ind); b := B(ind); k := K(ind);
    dep := G(ind), a, b, k;
```

The `rifsimp` routine that will be used for the determining equation reduction, by design, can only process differential polynomials, yet the sine and cosine functions present in the PDEs (5.2) are not polynomial functions of the constitutive parameter $k = \tan\gamma$. This restriction can be overcome by temporarily treating the sine and cosine as "arbitrary functions":

```
subs_sin_cos := {cos(g) = CG(K(x,t)),sin(g) = SG(K(x,t))};
PDE2 := subs(subs_sin_cos,PDE1);
```

Finally, the variables of the problem and the PDE family are declared:

```
gem_decl_vars(indeps = [ind],deps = [dep]);
gem_decl_eqs([PDE2],solve_for = [diff(G(x,t),t,t)]);
```

As usual, we seek generalized equivalence transformations, initially allowing all infinitesimal components to depend on all variables of the problem, and display these infinitesimal component dependencies.

```
det_eqs := gem_symm_det_eqs([ind, dep]) :
OFF : sym_components := gem_symm_components(); ON :
```

This yields 187 determining equations. To seek usual equivalence transformations, we restrict the dependencies of local transformation components as follows:

```
restriction_eqs := gem_generate_EquivTr_dependence([
   [[x,t],[x,t]],
   [[G],[x,t,G,A,B,K]],
   [[A,B,K],[A,B,K]]]);
```

Moreover, a relationship ensuring the proper treatment of $\sin\gamma$, $\cos\gamma$ in relation to $k = \tan\gamma$ is imposed:

```
cos_sin_cond:= {CG(K)^2=1/(K^2+1),SG(K)^2=K^2/(K^2+1)};
```

Importantly, the relationships `cos_sin_cond` are equivalent to polynomial ones, and can be handled by `rifsimp`. Finally, the equivalence transformation determining equations are given by the set

```
det_eqs1 := det_eqs union restriction_eqs union cos_sin_cond :
```

This set of equations is simplified using the command

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

which outputs 39 simplified PDEs, and computes the dimension of the solution space, i.e., the number of equivalence transformations in the specified ansatz: `simplified_eqs[dimension]=8`.

Finally, the simplified system of determining equations remains to be solved. Before that, we substitute the actual values for the sine and cosine functions, keeping in mind that $\gamma \in [0, \pi/2]$.

```
solve_sin_cos := allvalues(solve(cos_sin_cond,{CG(K),SG(K)}));
simplified_eqs2 := eval(subs(solve_sin_cos[1],simplified_eqs[Solved])) :
```

The solution and output of the equivalence transformation generators proceed as follows:

```
eq_tr_sol := pdsolve(simplified_eqs2, sym_components); gem_output_symm(eq_tr_sol);
```

The output (in Maple 18) is

$$\begin{aligned}
\_C5, \quad \mathrm{X}_1 &= \frac{\partial}{\partial G},\\
\_C7, \quad \mathrm{X}_2 &= \frac{\partial}{\partial x},\\
\_C8, \quad \mathrm{X}_3 &= \frac{\partial}{\partial t},\\
\_C3, \quad \mathrm{X}_4 &= t\,\frac{\partial}{\partial G},\\
\_C4, \quad \mathrm{X}_5 &= t\,\frac{\partial}{\partial t} + x\,\frac{\partial}{\partial x} + G\,\frac{\partial}{\partial G},\\
\_C6, \quad \mathrm{X}_6 &= -\frac{1}{2}\,t\,\frac{\partial}{\partial t} + a\,\frac{\partial}{\partial a} + b\,\frac{\partial}{\partial b},\\
\_C1, \quad \mathrm{X}_7 &= -x\,\frac{\partial}{\partial x} - 2a\,\frac{\partial}{\partial a} - \frac{4b}{k^2+1}\,\frac{\partial}{\partial b} + k\,\frac{\partial}{\partial k},\\
\_C2, \quad \mathrm{X}_8 &= -x\,\frac{\partial}{\partial G} + \frac{2bk}{(k^2+1)^2}\,\frac{\partial}{\partial a} + \frac{4bk}{k^2+1}\,\frac{\partial}{\partial b} + \frac{\partial}{\partial k}
\end{aligned} \tag{5.5}$$

(for details, see `E_g_04_equiv_tr_elast_antiplane_tan.mw`). The transformation generator X_8 in (5.5) corresponds to shifts in $k \equiv \tan\gamma$. The global group is found in Maple using the commands

```
Particular_Transform:= gem_extract_symm(sym_sol,spec={_C8=1});
Particular_Transform_ODEs:= gem_global_group(
 Particular_Transform, group_param_name='S');
Particular_Transform_Global:= dsolve(Particular_Transform_ODEs);
```

and is given by

$$\begin{aligned}
&G^* = G - Sx, \qquad \tan\gamma^* = \tan\gamma + S,\\
&\alpha^* = \alpha + 2\beta\cos^4\gamma\left(\frac{S^2}{2} + S\,\tan\gamma\right), \qquad \beta^* = \beta\cos^4\gamma\left(\tan^2\gamma + 2S\tan\gamma + S^2 + 1\right)^2,
\end{aligned}$$

with a group parameter $S \in \mathbb{R}$. In particular, the choice of $S = -k = -\tan\gamma$ lets one invertibly map the PDE (5.2) into the form (5.3) with $\gamma^* = 0$. This equivalence transformation is given by

$$x = x^*, \qquad t = t^*, \qquad G = G^* - x\tan\gamma, \qquad \alpha = \alpha^* + \beta^* \tan^2\gamma, \qquad \beta = \beta^* \cos^{-4}\gamma. \tag{5.6}$$

The substitution of (5.6) into the full PDE (5.2) yields

$$G^*_{t^*t^*} = (\alpha^* + 3\beta^*(G^*_{x^*})^2)\,G^*_{x^*x^*} \tag{5.7}$$

with

$$\alpha^* = \alpha + \frac{\beta}{4}\sin^2 2\gamma, \qquad \beta^* = \beta\cos^4\gamma$$

which is Eq. (5.3) up to the omission of the tildes.

We note that Eq. (5.7) involves the parameter α^* that is not necessarily positive. In particular, loss of hyperbolicity may occur when the coefficient of G_{xx} becomes negative [25]. A sufficient condition for the loss of hyperbolicity not to happen under the indicated transformation is given by

$$\sin^2(2\gamma) < \frac{4\alpha}{\beta}. \tag{5.8}$$

The condition (5.8) is satisfied for all fiber orientations γ, for example, in the common situation of relatively weak fiber contributions, $\beta < 4\alpha$. In this case, obvious non-dimensionalizing scaling transformations

$$\widehat{x} = \frac{x^*}{L}, \qquad \widehat{t} = \frac{\sqrt{\alpha^*}\,t^*}{L}, \qquad \widehat{G} = \sqrt{\frac{3\beta^*}{\alpha^*}}\,\frac{G^*}{L}, \qquad L \in \mathbb{R}^+ \tag{5.9}$$

of the PDE family (5.7) can be used to further map these equations into a single nonlinear wave equation

$$\widehat{G}_{\widehat{t}\widehat{t}} = (1 + (\widehat{G}_{\widehat{x}})^2)\,\widehat{G}_{\widehat{x}\widehat{x}}. \tag{5.10}$$

Thus through systematically computed equivalence transformations (5.6) and (5.9), the PDE family (5.2) is invertibly reduced to a wave equation (5.10) that involves no arbitrary elements. Properties of the PDE (5.10) and the extended versions of the model are discussed in [25].

6. Discussion

The current contribution describes a practical algorithm of computation of Lie groups of point equivalence transformations and generalized equivalence transformations of families of differential equations, and a symbolic implementation of that algorithm in the GeM package for Maple. The method is directly applicable to DE families with arbitrary elements in the form of constants or functions of independent and dependent variables. As discussed in Section 2, the computation is based on a replacement of arbitrary elements by effective extra dependent variables, and a computation of Lie groups of point symmetries of such a modified DE system, with appropriate additional conditions imposed on the transformation components corresponding to the arbitrary elements. The steps for symbolic computations are outlined in Section 3. In particular, the simplification of an overdetermined system of equivalence transformation determining equations is done using the highly efficient and robust rifsimp routine built into Maple software.

Section 4 contains basic examples of equivalence transformation computations using Maple/ GeM. Equivalence transformations for the classical family of dimensional KdV equations, and nonlocal equivalence transformations of a general class of nonlinear wave equations were computed therein. Additionally, the first computational example of a genuine generalized equivalence transformation was presented (Section 4.3).

In Section 5, equivalence transformations of a family of nonlinear equations describing anti-plane shear motions of a fiber-reinforced material were considered. For the first and the third example, it was shown how the equivalence transformations can be used to invertibly map a given DE family into a single equation of a significantly simpler form, involving no arbitrary elements. In order to simplify the presentation and interpretation, the examples chosen for the current manuscript were limited to a situation with one time and one spatial variable. However, the computation algorithm can be applied to ODEs as well as PDEs and systems thereof in multi-dimensions; the presented software can be used without any changes.

It is worth mentioning the computational efficiency of the algorithm within the GeM package that generates the determining equations for local symmetries and equivalence transformations. For a usual PDE system arising in physics, involving one to four independent variables and a comparable number of dependent variables, a typical computation and splitting of symmetry/equivalence transformation determining equations, in a problem that does not involve classifications, is finished usually within seconds, or a fraction of a second, on a standard desktop computer. The numbers of split determining equations can substantially differ, depending on the DE system at hand. Examples in Sections 4 and 5 involve less than two hundred determining equations; similar numbers of determining equations arise, for example, in the study of point symmetries of magnetohydrodynamics equilibrium equations [47]. We note that both GeM software and the Maple rifsimp routine can efficiently processes substantially more complicated cases; for example, calculations involving 31,918 and 58,273 determining equations were successfully completed in a related application of local conservation law computations in fluid dynamics models [48].

It is straightforward to extend the algorithm and the program sequence described in Section 3 to address equivalence transformation classification problems. An example of that kind would be a problem of classification of generalized equivalence transformations of the PDE family (5.2), with arbitrary elements α and β, with respect to the classifying parameter γ; in particular, for special values of γ, additional equivalence transformations may arise. The casesplit option of the rifsimp routine enables one to perform such classifications. Specific examples can be found in [2] (Section 5.4.2), with Maple code examples available from [46].

In conclusion, we note that there exist a significant number of challenging questions both in the theory of equivalence transformations and in the realm of practical computation. We briefly outline some of them here.

1. Equivalence transformations of a family of DEs, understood in the general geometrical sense, are evidently not exhausted by Lie groups of point equivalence transformations or their generalizations discussed in Section 2.3. The definition of broader classes of equivalence transformations which can be systematically analyzed remains an open problem, as does the analysis of completeness of equivalence transformation classifications within those classes.
2. A basic extension of the notion of point equivalence transformations is the notion of generalized equivalence transformations (Section 2.3(A)). Examples where the local transformation components for dependent and independent variables involve constant arbitrary elements are well-known. To the best of authors' knowledge, the first example in the literature of generalized equivalence transformations where the transformations of dependent and/or independent variables involve arbitrary *function(s)* of the model appears in [40]. The corresponding symbolic computations were performed in the current contribution in Section 4.3.
3. The so-called generalized extended equivalence transformations (see Section 2.3B) provide significant, practically useful extensions of sets of equivalence transformations admitted by certain families of differential equations. It is necessary to come up with systematic approaches to the computation of generalized extended equivalence transformations, and symbolic software-assisted calculation algorithms.
4. Symbolic computations of symmetries and equivalence transformations of families of *linear* partial differential equations present a substantial challenge, due to the presence of infinite-parameter symmetry groups. The same is true for DE families that admit linearizations by point transformations. A review of theoretical results pertaining to the computation of symmetries of linear PDEs and their software implementation can be found in [23]. In future work, it is important to develop a better understanding of the structure of Lie groups of (generalized) point equivalence transformations of linear PDE families, and extend software implementations for point symmetry computation of linear DEs to include the computation of some classes of equivalence transformations of families of linear equations.
5. In some classes of PDE models that arise, for example, in models of nonlinear compressible elastodynamics, constitutive functions may depend on combinations of derivatives of dependent variables of the problem, or more generally, be differential functions of the dependent variables [49,50]. The algorithm described in the current contribution can be used to compute generalized point equivalence transformations for such models where the transformations for the arbitrary elements depend only on those arbitrary elements themselves. It remains an open problem to find examples, general formulas, and an overall algorithmic approach to the computation of some broader classes of equivalence transformations for such problems, in particular, equivalence transformations with a general action on such arbitrary elements with complicated dependencies. This and some of the above questions are connected to the problems studied within the Cartan's method of equivalence [10,12] and the related recent developments, e.g., [13,14,16,18–20].

Acknowledgment

The author is grateful to NSERC of Canada for research support through a Discovery grant RGPIN-2014-05733.

Appendix A. Supplementary data

Supplementary material related to this article can be found online at http://dx.doi.org/10.1016/j.cpc.2017.06.013.

References

[1] L.V. Ovsiannikov, Group Analysis of Differential Equations, Academic Press, 2014.
[2] G.W. Bluman, A.F. Cheviakov, S.C. Anco, Applications of Symmetry Methods to Partial Differential Equations, Springer, 2010.
[3] I.G. Lisle, Equivalence Transformations for Classes of Differential Equations (Ph.D. thesis), University of British Columbia, 1992.
[4] S.V. Meleshko, J. Nonlinear Math. Phys. 3 (1–2) (1996) 170–174.
[5] N. Ibragimov, M. Torrisi, A. Valenti, J. Math. Phys. 32 (11) (1991) 2988–2995.
[6] N. Bila, J. Symbolic Comput. 46 (6) (2011) 659–671.
[7] O. Vaneeva, R.O. Popovych, C. Sophocleous, Acta Appl. Math. 106 (1) (2009) 1–46.
[8] G.W. Bluman, A.F. Cheviakov, J. Math. Anal. Appl. 333 (1) (2007) 93–111.
[9] S.-L.T. Huang, Algorithms for Symmetry Analysis (Master's thesis), University of Canberra, 2009.
[10] É. Cartan, Les sous-groupes des groupes continus de transformations, in: Annales scientifiques de l'école normale supérieure, vol. 25, 1908, pp. 57–194.
[11] É. Cartan, Les systèmes de Pfaff, à cinq variables et les équations aux dérivées partielles du second ordre, in: Annales scientifiques de l'école normale supérieure, vol. 27, Elsevier, 1910, pp. 109–192.
[12] É. Cartan, Les problèmes d'équivalence, in: Séminaire de mathématiques, 1937.
[13] N. Kamran, Contributions to the Study of the Equivalence Problem of Elie Cartan and its Applications to Partial and Ordinary Differential Equations, Vol. 45, Acad. Royale de Belgique, 1989.
[14] R.B. Gardner, The method of equivalence and its applications, SIAM, 1989.
[15] I.M. Anderson, N. Kamran, et al., Duke Math. J. 87 (2) (1997) 265–319.
[16] I.G. Lisle, G.J. Reid, J. Symbolic Comput. 26 (3) (1998) 355–379.
[17] S. Neut, M. Petitot, R. Dridi, J. Symbolic Comput. 44 (3) (2009) 261–270.
[18] F. Valiquette, SIGMA Symmetry Integrability Geom. Methods Appl. 9 (0) (2013) 29–43.
[19] B. Kruglikov, D. The, J. Reine Angew. Math. (2017).
[20] I.M. Anderson, C.G. Torre, J. Math. Phys. 53 (1) (2012) 013511.
[21] A.F. Cheviakov, Comput. Phys. Comm. 176 (1) (2007) 48–61.
[22] A.F. Cheviakov, J. Eng. Math. 66 (1–3) (2010) 153–173.
[23] A.F. Cheviakov, Math. Comput. Sci. 4 (2–3) (2010) 203–222.
[24] A.F. Cheviakov, J.-F. Ganghoffer, S. St.Jean, Int. J. Non-Linear Mech. 71 (2015) 8–21.
[25] A.F. Cheviakov, J.-F. Ganghoffer, J. Mech. Behav. Biomed. Mater. (2015).
[26] C. Su, C. Gardner, J. Math. Phys. 10 (3) (1969) 536–539.
[27] P.E. Hydon, Symmetry Methods for Differential Equations: A Beginner's Guide, Vol. 22, Cambridge University Press, 2000.
[28] P.J. Olver, Applications of Lie Groups to Differential Equations, Vol. 107, Springer Verlag, 2000.
[29] G.W. Bluman, S. Kumei, Symmetries and Differential Equations, Vol. 154, Springer Science & Business Media, 2013.
[30] N.H. Ibragimov, CRC Handbook of Lie Group Analysis of Differential Equations, Vol. 1, CRC press, 1995.

[31] N.H. Ibragimov, CRC Handbook of Lie Group Analysis of Differential Equations, Vol. 3, CRC press, 1995.
[32] I. Lisle, S.-L.T. Huang, J. Eng. Math. 66 (1–3) (2010) 201–216.
[33] G.J. Reid, A.D. Wittkopf, A. Boulton, European J. Appl. Math. 7 (06) (1996) 635–666.
[34] T. Wolf, in: J. Grabmeier, E. Kaltofen, V. Weispfenning (Eds.), Computer Algebra Handbook, Springer, 2002, pp. 465–468.
[35] T. Chaolu, P. Jing, J. Eng. Math. 66 (1–3) (2010) 181–199.
[36] W. Hereman, Math. Comput. Modelling 25 (8–9) (1997) 115–132.
[37] N. Ivanova, R. Popovych, C. Sophocleous, 2005, arXiv preprint math-ph/0505015.
[38] R.O. Popovych, M. Kunzinger, H. Eshraghi, Acta Appl. Math. 109 (2) (2010) 315–359.
[39] R.O. Popovych, N.M. Ivanova, H. Eshraghi, J. Math. Phys. 45 (8) (2004) 3049–3057.
[40] S. Opanasenko, A. Bihlo, R. Popovych, Group analysis of general KdV-Burgers equations (in preparation–private comunication), 2016.
[41] N. Ivanova, R. Popovych, C. Sophocleous, Lobachevskii J. Math. 31 (2) (2010) 100–122.
[42] O. Vaneeva, R. Popovych, C. Sophocleous, J. Math. Anal. Appl. 396 (1) (2012) 225–242.
[43] E. Varley, B. Seymour, Stud. Appl. Math. 72 (3) (1985) 241–262.
[44] G.W. Bluman, A.F. Cheviakov, J. Math. Phys. 46 (12) (2005) 123506.
[45] G.W. Bluman, A.F. Cheviakov, N.M. Ivanova, J. Math. Phys. 47 (11) (2006) 113505.
[46] GeM Software Package, Examples, and Description. http://math.usask.ca/~shevyakov/gem/.
[47] A.F. Cheviakov, Phys. Lett. A 321 (1) (2004) 34–49.
[48] A.F. Cheviakov, M. Oberlack, J. Fluid Mech. 760 (2014) 368–386.
[49] J.E. Marsden, T.J. Hughes, Mathematical Foundations of Elasticity, Courier Corporation, 1994.
[50] A.F. Cheviakov, J.-F. Ganghoffer, J. Math. Anal. Appl. 396 (2) (2012) 625–639.